1 2 9 0

UNIVERSIDADE Đ
COIMBRA

Flávio Joaquim Fernandes Pereira

INTELLIGENT SYSTEM FOR PLANNING TRANSPORT NETWORK INFRASTRUCTURES

Internship Report in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Eng. Rafael Maia from Grama company, and Prof. António Dourado from Department of Informatics Engineering and presented to Faculty of Sciences and Technology / Department of Informatics Engineering.

October 2020

Flávio Joaquim Fernandes Pereira

INTELLIGENT SYSTEM FOR PLANNING TRANSPORT NETWORK INFRASTRUCTURES

1 2 9 0 UNIVERSIDADE Đ COIMBRA

Faculty of Sciences and Technology

Department of Informatics Engineering

Intelligent System for planning Transport Network Infrastructures

Flávio Joaquim Fernandes Pereira

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Eng. Rafael Maia from Grama company, and Prof. António Dourado from Department of Informatics Engineering, and presented to the Faculty of Sciences and Technology/Department of Informatics Engineering.

October 2020

UNIVERSIDADE Ð COIMBRA

Abstract

New modes of mobility and better access to points of interest are just some of the benefits that good transportation network infrastructure planning can bring to a city. This report describes the development of an Intelligent System whose ultimate goal is transport network infrastructure planning.

To answer this problem, Motum was developed as a data visualization web application that a) detects patterns in the movement of the population by using clustering algorithms, b) determines which transport modes the inhabitants choose for their trips by using supervised learning algorithms and c) suggests locations that may possibly require more transport infrastructure by using data mining simple approaches. These three features provide city transport planners with relevant information for their decision making. These decisions are also supported by trip maps, heat maps and mobility charts.

The methods that serve the functionalities of the web application were tested and clustering of trips results showed good state of the art validation metrics, along with a graphical analysis and interpretation of the patterns that the clusters may yield. Transport mode multiclass classification showed high performance of the machine learning models in classifying modes of transport for any trips data set. More specifically, macro F-measure of 83.03%, a Precision of 84.02%, a Recall of 82.32% and Accuracy of 85.75%.

Keywords

Public Transports, Machine Learning, Clustering, Planning of Transportation Networks Infrastructures, Artificial Intelligence, Amazon Web Services

Contents

Acronyms

AI Artificial Intelligence. 10, 16

ANN Artificial Neural Network. x, 16, 17, 60, 91, 95

AWS Amazon Web Services. xiv, 3, 27, 28, 33, 70

CDR Call Detail Records. 24, 44

CRUD Create, Read, Update and Delete. 35

DL Deep Learning. 16

DT Decision Tree. 11, 91, 93, 94

GBM Gradient Boosting Machines. 13, 60

GPS Global Positioning System. 44, 45

IBM International Business Machines Corporation. xiv, 27, 28

IQR Interquartile Range. 54, 56

KNN K-Nearest Neighbor. 11, 60, 62

LGBM Light Gradient Boosting Machines. xv, 13, 60, 61, 91, 93–95, 97

LSTM Long Short Term Memory. 17

ML Machine Learning. 6, 7, 10, 13, 16, 23, 60, 90, 91, 97, 98

RF Random Forest. 12, 60, 91, 93–95

RNN Recurrent Neural Network. 16, 17

S3 Simple Storage Service. 68

UML Unified Modeling Language. 34

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this document is to summarize the work done at Grama during the internship, which is a discipline of the Master's degree in Informatics Engineering course lectured in the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra (DEI-FCTUC). It usually has a duration of two semesters, and makes part of the final year of the degree's programme. Both the internal supervisor from DEI-FCTUC, Prof. Dr. António Dourado, and an external and main supervisor from Grama, Eng. Rafael Maia, guided the whole process.

This chapter is divided into six sections. We start by presenting in Section 1.1 the work environment in which the internship was placed. Then, in Section 1.2 and 1.3 the context and motivation, and the main goals of the project are presented. In Section 1.4 and 1.5 both planning of the first and second semester and the risk analysis are presented. Finally, in Section 1.6 the document structure is conveyed.

1.1 Work Environment

The internship took place at Grama, Lda., Grama, a software company founded in January of 2017 and based in Coimbra. It has highly trained software engineers that have designed, developed and delivered software solutions for some of the largest mobile operators in the world. It currently has ongoing projects in Europe and South America, mostly in the Telecoms and Financial markets. Among its areas of expertise are project management, solutions design, UI/UX design, full-stack software development and automated testing. Grama is part of Nest Collective, which is a community of small tech businesses and startups that share office space, business opportunities and knowledge.

1.2 Context and Motivation

Given that platforms for planning transport network infrastructures are increasingly in demand, and nowadays, people's location is being collected in an automated way, the trips they take throughout the day can be inferred in order to aid the development of these platforms. With these technological advances, insights into people's mobility in relation to existing transport infrastructure in a city can be obtained, with the main objective of improving accessibility and the provision of public transports. And this is where the need for Motum arises - a proof of concept web application that aims to answer four major questions:

- Can people mobility be represented in a city map along with the surrounding transport infrastructures?
- Can patterns from large population movements in a city be extracted?
- Can transport modes of trips be inferred from their characteristics?
- Are there large population movements that possibly require additional service from the transport network?

Open PFLOW [8] is a novel data set creation approach that continuously reports the spatiotemporal positions of all individuals in urban areas based on open data. This data is used in this project to create a platform to assist government entities in planning transport network infrastructure.

1.3 Objectives

The general objective of the internship was to develop the first version of a web application to assist government entities in the planning of transport network infrastructure.

Within the scope of the Motum web app, the objectives were to:

- Develop representations for the mobility of people together with transport infrastructures that visually allow to obtain conclusions for the planning of these infrastructures.
- Detect patterns in people's trips so that interpretable travel groups are obtained.
- Assign modes of transport to trips that are not labeled and that have accessible features.
- Suggest locations for new bus stops in places where large population movements possibly do not have the necessary service from the transport network.

Users should be able to take advantage of visual representations of mobility, transport infrastructure and also visualize results obtained through the application of Machine Learning methods. Administrators will be able to manage all application data and for this reason, the application must have authentication. The design and quality assurance of the application is not in the objectives for the student and will be carried out by members of the company responsible for those areas.

1.4 Planning

The planning of the project was divided in two phases: one for the first semester and another for the second one; all this in order to fulfill the must have requirements.

1.4.1 First Semester

In the first semester, the main concerns consisted in studying the AWS infrastructure for web platform hosting and to gather the State of the Art on Transportation Infrastructure Planning; more properly on human mobility patterns extraction and transport mode classification which were thought to be the key points of the implementation process.

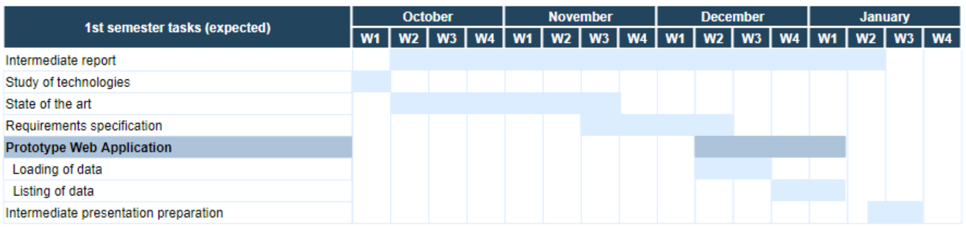


Figure 1.1: Gantt Diagram of the suggested plan for the first semester.

Figure 1.1 shows the Gantt diagram of what was expected to happen in the first semester. It is possible to view all the tasks planned for the first semester and the estimated time needed to perform each one.

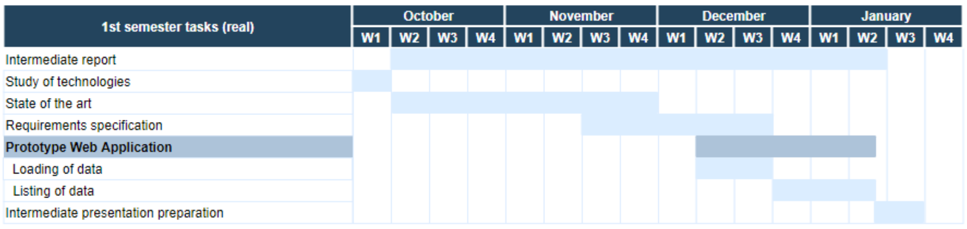


Figure 1.2: Gantt Diagram of the tasks performed in the first semester.

As figure 1.2 shows, some unexpected **deviations** occurred during the first semester's work:

- The prototype was only finalized in the middle of January, when it was almost time for the intermediate defense, due to some problems that arose regarding the intern's learning curve with AWS infrastructures and some Lambda functions timeout struggles.
- The study of technologies was not fully done in the beginning of the semester, but also during the prototype implementation phase. This way, hands-on experience was addressed in addition to the initial theoretical learning.
- Due to the workload that the company was having, requirements were being produced iteratively and they were only ready near January.
- The intern had no contact with the mobility data in the first semester.

1.4.2 Second Semester

A plan for the second semester was made as a suggestion of the intern.

Figure 1.3 shows the Gantt diagram of what was expected to happen in the first semester. It is possible to view all the tasks planned for the first semester and the estimated time needed to perform each one. The data expected to be obtained was Call Detail Records,

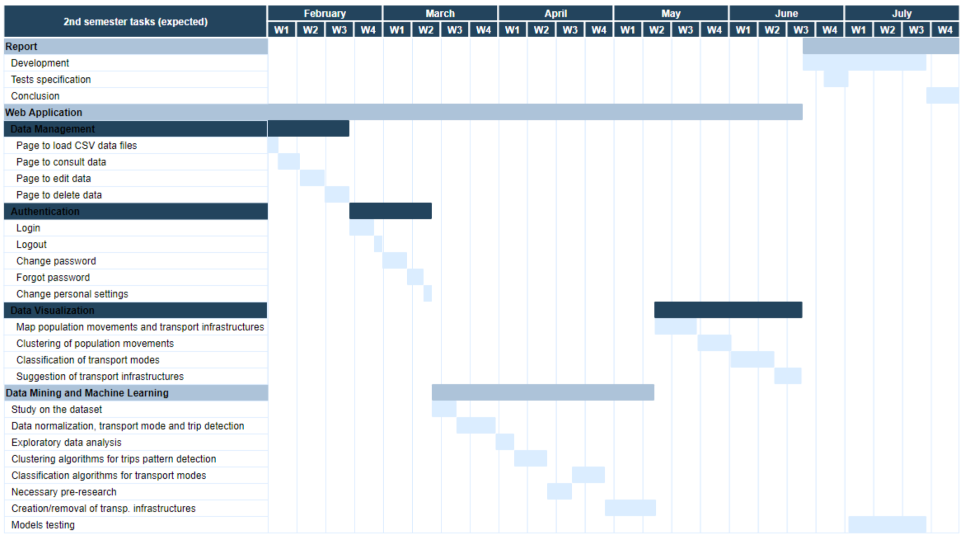


Figure 1.3: Gantt Diagram of the suggested plan for the second semester.

but were not obtained. Therefore, steps like extracting transport mode from CDRs were not necessary. Alternatively, similar data were obtained, that is, spatiotemporal location of individuals in Tokyo. The project was also expected to be completed by the end of July, which did not happen and will be explained.

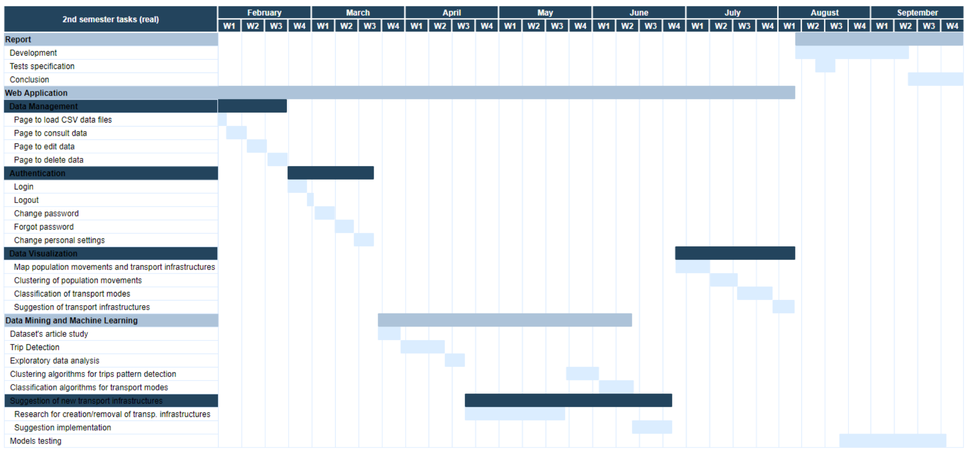


Figure 1.4: Gantt Diagram of the tasks performed in the second semester.

It was initially planned to develop a version of the Motum Web App with at least all the must have requirements presented in section 7.2 implemented. Figure 1.4 shows all the tasks necessary to fulfill the must have requirements, although one of the requirements had to be changed in order to finalize the project, as its reason will also be explained.

The second semester process included the following main tasks:

- Loading, Query, Update and Delete of Historical Data
- Session Management
- Trip detection by using the spatiotemporal positions of individuals
- Pattern detection in trips
- Transport mode assignment to trips

- Bus stop locations suggestion
- Historical Data Visualization (trips and transport infrastructures)
- Data Mining and Machine Learning back end implementation on AWS
- Data Mining and Machine Learning front end visualization of results

1.5 Risk Analysis

Risk analysis is one of the most important aspects of managing any software project. Risk, in terms of managing a software project, is any situation that prevents a project from reaching a successful conclusion and therefore delays its execution. That said, it is crucial to conduct a risk analysis in order to anticipate possible problems and to avoid or mitigate them. At the end of the first semester, the possible risks that could occur during the development of the project were analyzed and listed, characterizing these taking into account their impact and probability.

Risk	**Probability**	**Impact**
Non-fulfillment with all imposed requirements	High	Medium
Not obtaining appropriate mobility data	Medium	High
Bad performance of ML models	Medium	Medium
Unsuccessful deployment and no feasibility of ML models in the cloud	Medium	High
Unsuccessful implementation of complex map functionalities	Medium	Medium

Table 1.1: Risk Analysis.

With the analysis carried out, it was possible to identify the risks presented in 1.1 table, which was very useful since it identifies the probability of those risks showing up and the impact they might cause in the project. Following, the risks are described and the probabilities of occurring as well as the degree of impact they can cause are justified. Finally, the mitigation strategy that was designed for each risk is explained.

- **Non-fulfillment with all imposed requirements**
  - The workload of the internship was considered high for the amount of time available for development. For this reason, there was a high probability risk of not achievement of objectives by the end date of the project or a postponement so that these could be fulfilled. The impact of this was considered to be medium, given that does not prevent the project finalization, but it could affect the quality of the final product.

    To mitigate this risk, priorities were defined for each of the objectives so that the most important ones are met. Thus, in the case of non-compliance with any of the objectives, it was still possible to present a functional version of the product.

The order of requirements was defined by their priority, which was defined as nice to have or must have. It can be said that this risk partially occurred, since two must have requirements had to be changed, but the requirement that resulted from these two has been completed. The final product was impacted by this risk, since instead of having the feature of adding/removing transport infrastructure, it has a suggestion of bus stops.

- **Not obtaining appropriate mobility data**
  - Quality data is a very important factor for the good performance of Machine Learning models. That said, for the project problem it was necessary that data that represented mobility were obtained. This risk had a medium probability of happening and its impact was considered high, since not obtaining any data could lead to a change of theme or even termination of the project.

    Given that Grama tried several contacts with mobile operators to obtain this data but was unable to, it was necessary to mitigate this risk. The Open PFLOW data set was then discovered as a data set that portrays the mobility of the inhabitants of Tokyo.

- **Bad performance of Machine Learning (ML) models**
  - The risk of ML models' unpredictability is due to the fact that everything depends on the quality of the data and the intern's ability to create good models. In addition, Machine Learning relies on the creation of state-of-the-art hypotheses and assumptions that can be valid or invalid. Also, trial and error to identify the best settings for the best models is something that a data scientist has to face while trying to create good models. That said, in the beginning a project, it is difficult to deduce what is the performance the final models will achieve. The likelihood of this happening was medium, as there are other risks on which it depended. The impact of this is also medium, given that the poor performance of the models translates into the poor quality of the final product.

    In order to mitigate this risk, a state-of-the-art review was carried out to understand which were the best methodologies to implement for the research problem so that the process was well executed. Additionally, an acceptable amount of tests was performed in order to understand which settings result in the best possible performance using the Open PFLOW data set, although with some limitations: the tests were done at night, with a possible duration of 8 to 10 hours since they were made with the same computer on which, during the day, software was implemented and the report was written. This risk impacted the clustering algorithms more than the ML models since the latter obtained good visible results, whereas in clustering the results were not so conclusive.

- **Unsuccessful deployment and no feasibility of ML models in the cloud**
  - Due to the lack of experience in this necessary functionality, there was a medium probability that the intern would have some difficulties in deploying a trained ML model into the cloud. This risk has a high impact, given that the non-deployment of the trained model to the cloud results in flaws in the final product that prevent a major objective of the project from being achieved - the trips modes of transport classification.

    Prior research on approaches to the deployment of trained models and approaches to importing these models into a Lambda Function helped mitigate this risk. Thus, the models are used in real time, when the user requests them.

It can be said that this risk impacted the final product since the intern was able to successfully deploy the ML model but it turns out that requests for results to the model take a loading time that for the user may not be acceptable.

- **Unsuccessful implementation of complex map functionalities**
  - This risk has two aspects - both the front end and data analytics. In the data analytics part, initially it was thought to implement the feature of adding/removing infrastructures from the transport network and assess the consequences that these changes would cause in the transport network and inhabitants mobility. This is mostly seen as an optimization problem, rather than Machine Learning. Therefore, implementing such features had a risk. On the other hand, in the front end, although the intern already mastered the framework used in its implementation, he never developed maps that required such features that were required by this project, therefore implementing features for adding and removing infrastructure in the map was also considered risk.

    As can be read in 5.3.1, an investigation was done on Machine Learning and optimization techniques applied to transport networks. It was discovered that Graph Theory is a go-to for the problem that the intern faced. As such, a specific requirement had to be changed. Approaches were studied and implementations were tried using Graph Databases. This caused a deviation that may have happened due to the fact that the intern was in remote work and recurrent communication was not the same. The final work was impacted in the sense that the two infrastructure requirements were changed. Regarding the front end part, several libraries for the implementation of the map were investigated. The used library was adequate to meet the proposed objectives that at the time of developing the front end part no longer had to do with adding and removing infrastructure, but with the representation of the suggested bus stops.

The project had to be postponed due to some risks present in the table. In short, all requirements were met, with the exception of the transport infrastructures one, that had to be changed. The data set obtained was not ideal given its artificial nature, but it served the case given that it gives information that is similar to what GPS location data or even Call Detail Records both give. Regarding the suggestion of transport network infrastructures, it is concluded that there is room for improvements with the introduction of new data. The same can be said of the map, which is somewhat slow and there may be room for future optimization.

1.6 Document structure

In this section, the chapters of the document are spelled out, as well as the contents presented in them.

- **Chapter 1 - Introduction**

  - This is the first chapter of the report, in which the context and motivation that led to the need for this product are conveyed. It also explains the objectives of both the internship and its project. Finally, planning and Risk Analysis are presented.

- **Chapter 2 - Background Knowledge**

  - This chapter presents some concepts of Artificial Intelligence that the reader must acquire to fully understand the rest of the document.

- **Chapter 3 - State of the Art**

  - This chapter starts by presenting two platforms that also attempt to plan transport network infrastructures. After these platforms are explained and since mobility patterns extraction is a key point for the implementation process, a literature review on mobility inference was carried out. Then, a literature review was also carried out this time for people's trips transport mode classification. Lastly, a study was made about the serverless technologies to be used in the desired system.

- **Chapter 4 - Project Specifications**

  - This chapter describes all the project specification processes: the adopted development methodology, project requirements, use case diagrams of the application, system architecture and technologies.

- **Chapter 5 - Development**

  - In this chapter, the main aspects of the development process are explained. First, the data set is introduced and validated. Trips are extracted from positional data over the course of a day and an exploratory analysis is made on them. Next, the implementation of the features that required machine learning is described. The suggestion of new locations for bus stops is further described and, finally, the web application functionalities are presented.

- **Chapter 6 - Tests and Results**

  - In this chapter, the clustering tests performed are presented in an attempt to interpret results and evaluate the quality of these methods. Next, the tests for the classification of transport modes are presented. Finally, it explains how the web application tests will be carried out. All testing approaches are explained in this chapter.

- **Chapter 7 - Conclusion**

  - This chapter presents the conclusions to all the work done, the overcome difficulties and the improvements for future work.

Chapter 2

Background Knowledge

In this chapter, some background concepts necessary to understand the following chapters and document are introduced. In Section 5.2 some concepts related with Machine Learning are reviewed. Then, in Section 2.2 some definitions and concepts related with Deep Learning are also reviewed.

2.1 Machine Learning

Machine Learning (ML) is a sub-area of Artificial Intelligence (AI) where computer algorithms are studied in order to allow computer programs to automatically improve through experience [11], as these have the ability to learn from experience without direct human intervention.

Machine Learning can be classified in several types. The following are the three best known:

- **Supervised Machine Learning:** the algorithm uses features and corresponding labels. This process is also called training [12]. During this process, the algorithm gradually determines the relationship between features and their corresponding labels. After this relationship is settled, a model is obtained. Supervised learning problems can be grouped into two types of problems:
  - **Classification** is mainly used for problems where the output variable is qualitative, such as the problem of classifying the fruit that is in images [13].
  - **Regression** is mainly used for problems where the output variable is quantitative, such as predicting house prices [13].
- **Unsupervised Machine Learning:** the main objective is to find patterns in unlabeled data using a model that has no hints on how to categorize each object [12]. Therefore, the model needs to infer itself its own rules to do it.
- **Reinforcement Learning:** consists in learning to map actions to situations in order to maximize a numerical reward. Therefore, a Reinforcement Learning agent selects the actions that lead him to maximize the accumulation of rewards over time [14].

As previously mentioned, Machine Learning can be classified into different types. However, there is one particular concept common to all of them that can have distinct interpretations, *parameter*. This word can lead to confusion since it can be used in two different terms: *parameter* and *hyperparameter*. **Parameters** are internal variables to the model that are configured and estimated from data [15], such as weights in an artificial neural network or the coefficients in linear regression. Moreover, it can be also used in programming to refer to an argument of a function. On the other hand, **hyperparameters** are external variables to the model whose values cannot be estimated from data [15]. Examples are number of layers in a neural network and k in k-nearest neighbors.

Supervised learning algorithms that are mentioned throughout the document are now described in order to facilitate the understanding to the reader.

2.1.1 Supervised learning algorithms

K-Nearest Neighbor (KNN)

The **KNN** predicts a new sample using the K-closest samples from the training set [15]. The developer defines the distance between samples and the most commonly used metric is the Euclidean distance (i.e., the straight-line distance between two samples).

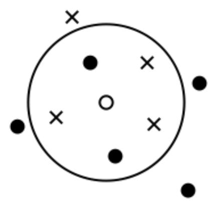


Figure 2.1: Example of KNN [1].

In figure 2.1 we can see a small white circle - which is the object to be classified - five black circles and four crosses. These object shapes represent the category that is assigned to each object in the training set. This is a 5-Nearest Neighbor, given that the 5 closest neighbors to the object are being analyzed in order to perform the majority voting of which class the object will belong to. In this case, it will be a cross (3 votes against 2). As it is noticeable, the major drawback of this algorithm is that it becomes significantly slow as the size of the data grows, given that it scans the training set in memory each time a prediction is needed

Decision Tree (DT)

Decision Trees are flowchart-like structures that predict target values by learning simple decision rules inferred from the data provided. Decision rules are usually in the form of nested if-then statements [15]. Both regression and classification problems can be solved by using Decision trees.

Figure 2.2 shows a decision tree that classifies Saturday mornings according to whether or not they are suitable for playing tennis. Each internal node represents a test on a feature, while each leaf node represents the decision taken after computing all features - a class label [2]. Branches represent sets of features that lead to these class labels.

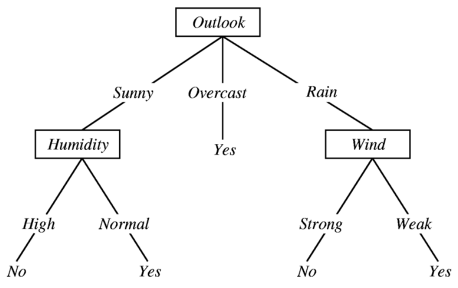


Figure 2.2: Example of a Decision Tree [2].

Random Forest (RF)

As the next algorithm belongs to the class of ensembles, it is necessary to define what ensembles are. **Ensemble Learning** methods combine many models' predictions into one model. These models are called Ensembles and tend to have much better predictive performance than single models [15].

RF consists of a large number of decision trees that work as an ensemble. Each model in the ensemble generates a prediction for a new sample and these predictions are averaged to give the forest's prediction [15]. In another Ensemble Learning technique, **Bagging**, all of the trees will look quite similar to each other. Thus, the predictions from these trees will be highly correlated, and averaging many highly correlated quantities does not reduce the variance of the results as much as averaging many uncorrelated quantities. Random forests solve this problem by forcing each split to consider only a subset of the features - randomness of the process. This can be called *decorrelating the trees.* These uncorrelated models will produce final predictions that are more accurate than any of the individual models' predictions [15] [13].

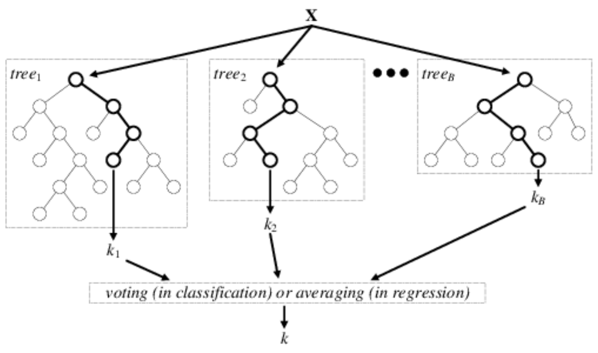


Figure 2.3: Example of a Random Forest [3].

Figure 2.3 represents a Random Forest where we can see that the β trees are relatively uncorrelated. Each decision tree in the ensemble generates a prediction for a new sample X and these predictions go to a voting process, for classification problems, or are averaged, for regression problems; so that we can have forest's final prediction.

Boosting algorithms

For the definition of the last two supervised learning algorithms, it is necessary to define what Boosting consists of. **Boosting** is one of the categories of Ensemble Learning. These are based on training several simpler models in order to produce a more robust final model [13]. However, in the Boosting algorithms, the models are not trained independently from each other, but in a sequential manner, based on an adjustment of the previously trained models. In addition, to maximize the performance of the final predictor, Boosting iteratively trains new models with a focus on observations in which the previous models made the most mistakes, thus decreasing the bias of weak learners. Then, the model is updated to prioritize the predictions with the greatest error in the observations of the test set. The way in which this training and update takes place is where the various Boosting algorithms differ.

Gradient Boosting Machines (GBM) are Boosting algorithms that, instead of establishing weights for weak learners, train new models directly on the error of the previous models, that is, the new models try to predict the error of the previous models instead of independently predicting the target [15] [16]. Considering decision tree as the base model and a regression problem, in the Gradient Boosting process, the first model firstly makes a very simple approximation of the prediction and we obtain the residual errors - predicted value subtracted from the observed value. Then, more models are trained to predict these residual errors, to try to predict the error of the first model. Thus, when we add the predictions of each model to obtain the final prediction, we obtain a more corrected version of the first prediction. Once a stage where residuals do not have any pattern that could be modeled, the algorithm may stop, otherwise it might lead to overfitting.

Light Gradient Boosting Machines (LGBM) is a Gradient Boosting framework that was designed to improve Gradient Boosting machines that use Decision Trees [17]. GBM was unsatisfactory in terms of efficiency and scalability when the feature dimension is high and data size is large. Light GBM brings two novel techniques to solve these problems, and differs from XGboost in the sense that while this grows trees depth-wise, LightGBM uses a leaf-wise algorithm [18]. The advantages it brings over traditional Gradient Boosting and even XGBoost are:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel and GPU learning.
- Capable of handling large-scale data.

2.1.2 Hyperparameter tuning

As previously mentioned, ML algorithms' hyperparameters are external variables to the model whose values cannot be estimated from data [15]. However, they can cause several performance problems and, therefore, to achieve optimal performances, they need to be tuned ([19], [20], [21]). There are three well known hyperparameter optimization approaches: Grid Search, Random Search and Automatic Hyperparameter Tuning.

Grid Search, tries a set of configurations of hyperparameters and trains the algorithm accordingly, choosing the hyperparameter configuration that gives the best performance. A grid of configurations is formed as the limits and steps between values of the hyperparameters were specified.

Random Search, navigates through the grid of hyperparameters randomly, normally obtaining similar performance to a full grid search with an appropriate number of evaluations. Authors show that if there is a region with hyperparameters close to optimal that occupies at least 5% of the search space, then a random search with an appropriate number of trials will be able to find that region with high probability.

In Grid Search and Random Search, the next trial is always independent to all the trials done before. For that, **Automatic Hyperparameter Tuning** uses knowledge about the relation between the hyperparameter settings and model's performance in order to make a better choice for the next parameter settings. This way, it typically requires less iterations to get to the optimal set of hyperparameter values. According to two articles [22] [23], three main components compose the algorithm:

- the **search space** of hyperparameters, which are distributions of hyperparameters that the developer sets.
- an **objective function**, which serves as the method to evaluate hyperparameter combinations.
- a **surrogate** and a **selection** function that act together to evaluate only hyperparameters that most likely will increase the validation score. The **surrogate** function can be thought of as an approximation of the objective function, which is used to propose hyperparameter sets to the objective function that likely improvement the validation score. There are different algorithms for this surrogate function but in this work the one called Tree Parzen Estimator (TPE) will be underlined. With the TPE algorithm, the surrogate function becomes a probabilistic model that maps hyperparameters to a probability of a validation score on the objective function. Shortly, it consists of an application of Bayes' Theorem. Hyperparameter sets that are put forward for evaluation by the objective function are selected by applying a criterion to the surrogate function that is defined by a **selection** function. Expected Improvement is commonly used as metric for that.

Briefly, in each iteration, everything is articulated as follows. A set of hyperparameter values that maximise the Expected Improvement is initially found. This combination of hyperparameters are fed to the objective function for evaluation and a score is retrieved. The surrogate function is updated with the feedback of the objective function by applying Bayes' theorem.

2.1.3 Clustering

As a case of Unsupervised Learning and a common way to detect patterns in data we have Clustering. Throughout this report, much emphasis will be placed on this type of methods in order to recognize patterns in trips data. **Clustering** relies on unsupervised machine learning to group unlabeled objects with similar properties into a set usually known as **Cluster**.

The literature presents many ways to categorize clustering methods since categories may overlap each other so that a method may have features from other categories. In this report, the division of clustering into their types is done following a chapter in a 2005 book [24], which presented most of the definitions that were needed for this section. Other authors, like in [25] and [26], also aided in parts where doubts arose. All 3 references contributed to the definitions that follow.

First, one way to divide clustering methods consists of**Exclusive**, that means that each object belongs to a single cluster, and **Overlapping**, or **Non-exclusive** Clustering, that means the opposite: an object can simultaneously belong to more than one cluster. Another division of Clustering types is **Partitional** vs. **Hierarchical**. Partitional Clustering can be thought of as a division of data objects into non-overlapping clusters and each of these data objects belong to exactly one cluster. Hierarchical Clustering, on the other hand, allows clusters to have subclusters. This way, a hierarchical tree is constructed with a set of nested clusters and the clusters formed in the end are also non-overlapping. Moreover, hierarchical methods can be categorized as **agglomerative** or **divisive**. The agglomerative one starts with each object as a separate group. Then, it iteratively merges the groups close to each other, until all the groups are merged into one - the highest level of the hierarchy - or a termination condition holds. The divisive approach begins with one cluster containing all the objects to classify. Then, iteratively splits each cluster into smaller ones until each object is in a single cluster, or some predefined termination condition holds. Finally, a clustering method can be categorized as **Complete** if all objects are assigned to a cluster, or **Partial**, otherwise. Partial Clustering is usually applied to objects in a data set that may not belong to well-defined groups.

There are two widely used forms of Clustering, which are important in the entirety of this work, are Fuzzy C-Means Clustering and K-Means Clustering. In detecting patterns in people mobility, these are the most used, however, as far as this work is concerned, K-Means was widely used.

Fuzzy C-Means Clustering is a Partitional method and consists in a form of clustering in which each object can belong to more than a single cluster, that is, every object gets assigned a membership degree to each cluster. This membership degree sets the degree to which objects belong to each cluster, so that objects in the center of a cluster have a higher degree of belonging to this cluster. Simply put, to each object is assigned a value between 0 and 1 which describes its degree of membership in a certain cluster and this degree ranges from 0 to 1.

K-Means Clustering is a Partitional and Exclusive method that makes the grouping of similar objects and discovers its hidden patterns. Briefly, K-means defines an arbitrary K number of centroids and then assigns to every object its nearest cluster. The process starts with randomly selected centroids as the first clusters and then performs calculations for each iteration, optimizing the positions of the centroids. **Centroid** is the average position of all the points belonging to a cluster.

In Fuzzy C-Means Clustering, each point has a membership value related to each cluster as opposed to K-Means, in which points can only belong to one cluster. In the Fuzzy variant, if it is necessary to define the cluster of each point it is necessary to defuse the result. A simple technique is to assign it to the cluster with the highest degree of belonging.

Another clustering algorithm that is important to emphasize in this work is the **DBSCAN** that is considered a density-based clustering algorithm that produces a partitional clustering. The number of clusters produced is entirely decided by the algorithm. Points located in low-density regions are classified as noise points, thus DBSCAN clustering result is not complete.

2.2 Deep Learning

Evolving every day as a subset of ML and generally as a case of Supervised Learning, Deep Learning (DL) is now one of the most typical AI techniques being used for the processing of large amounts of data, and for the scope of this work, it is not different.

DL programs benefit from multilayer Artificial Neural Networks to learn data representations from large amounts of it [27]. It works similarly to how we think the human brain works, such that a Deep Learning program would perform a task repeatedly, each time changing its parameters to improve the outcome. The term deep usually refers to the high number of hidden layers these networks have.

As Artificial Neural Networks are greatly used in DL, a further explanation may be needed. **Artificial Neural Network (ANN)** is a computation model inspired on the structure and functions of biological neural networks [27]. It consists of sets of units known as artificial neurons, parallelly organized in layers, and these serially connected in networks. These connections are outputs of neurons, which become an input to another neuron in the network. A weight that represents its relative importance on the neural network is assigned to each connection. Every neuron can possess many to many relationships with multiple inputs and output connections. The input data feeds the entire neural network by suffering various operations and producing output values. Figure 2.4 shows it.

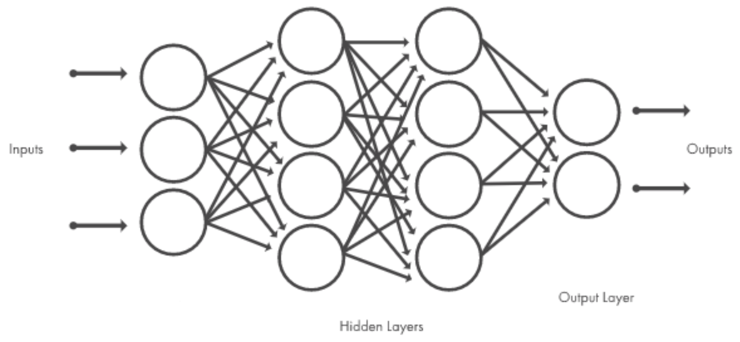


Figure 2.4: This figure represents an ANN which has its inputs, two hidden layers and one output layer. Adapted from [4].

Recurrent Neural Network (RNN) differ from feedforward neural networks in that the first can take sequential type data with arbitrary sequence size [28]. Moreover, an element of this sequence can influence its neighbors. These networks address the issue of reasoning about past events to inform the present ones, that is, RNN's target output sequence is the input sequence with an advance of 1 step. An RNN can be thought of as a ANN with loops. It is a succession of simple ANNs.

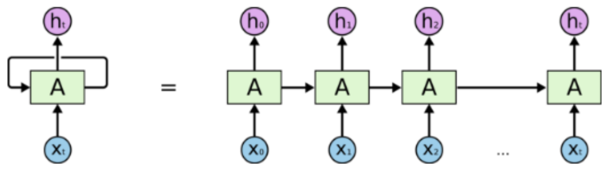


Figure 2.5: Unrolled recurrent neural network [5].

As can be seen in figure 2.5, its structure resembles a chain and each layer passes a message to a successor.

When RNNs were invented, researchers rapidly noticed that it would be difficult for this type of ANN to handle long-term dependencies. It is possible to think of an example of language modeling where this problem could be raised. Given any sentence, the needed past information to predict the next word could be too further back for what the RNN can handle.

As a reaction to this RNN's issue, Hochreiter and Schmidhuber (1997) [29] introduced a special kind of RNN, **Long Short Term Memory (LSTM)**. These were created with the main purpose of solving the long-term dependencies issue. LSTM's structure is quite more complex than RNN's.

Chapter 3

State of the Art

In this chapter, a comprehensive literature review is presented. Then, the key findings on transport networks infrastructure planning are described.
In section 3.1, the main platforms that attempt to solve similar problems are discussed. Then, in section 3.2 and 3.3, a plethora of available studies on urban mobility patterns inference and transport mode classification is analyzed.
Ultimately in section 3.4, the various cloud providers are studied in order to thoroughly support the architectural choices.

3.1 Similar Platforms

In this section, platforms with the similar goals are analyzed. For this, two existing platforms will be presented because they have aspects common to those of the internship product. Both provide relevant features for the planning of transport network infrastructure.

3.1.1 REMIX

At first glance, this platform presents its slogan as "Transportation is changing fast. We help you stay ahead.". By the slogan it is interpretable that as transport is changing, their platform makes the user follow that change.

Cities are being forced to adapt their transport infrastructures, in such an emerging way that a San Francisco's startup, called Remix, developed a cloud-based software-as-a-service platform to help inform their decisions. This platform provides transportation agencies with data - including statistics like collisions, curb and street density - and produces holistic approaches to improve policies, routes and infrastructures. Remix divide their solutions into four aspects:

- **Transit** Public transit planning platform (Design, Evaluation and Collaboration).

  Figure 3.1 was extracted from a *demo* on how to use REMIX. In this video, the transportation planner can (a) add layers from any custom data (ridership, land use, etc), (b) view networks more holistically by crossing proposed lines with all the existing ones, (c) place a rider on the map and watch how far he can go and how many jobs he can reach in a given number of minutes and (d) measure impact with or without the proposed project. Lastly, it is possible to export it.

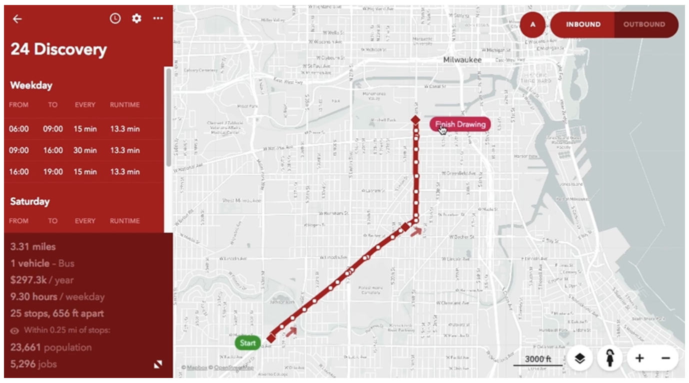


Figure 3.1: Image of Remix's Interface for the proposal of a new service, and a list of its related statistics on the left [6].

- **Streets** Data-driven streets redesign in real-time.

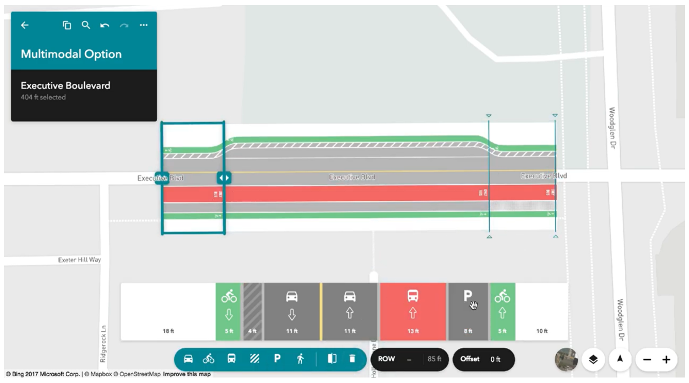


Figure 3.2: Image of Remix's Interface for planning a bicycle facility [6].

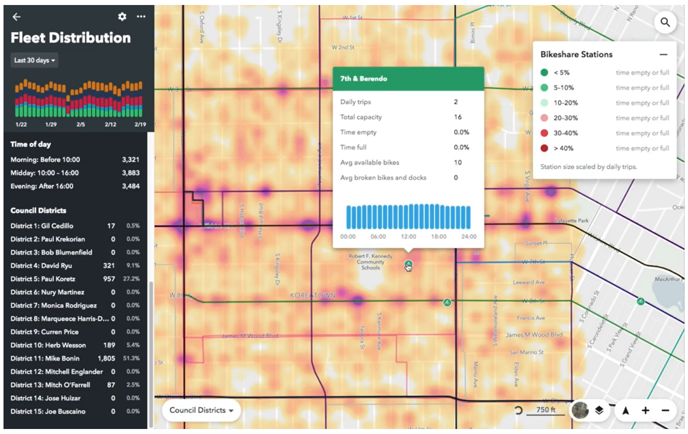
Figure 3.2 shows that REMIX has many functionalities, such as visualization of data and design of lanes all in one place. It is also possible to compare the scenario against the existing and actual conditions, and assess the impact on the transportation network

- **Shared Mobility** Easy-to-use dashboards with micro mobility performance measurement and planning of new and safer infrastructure for riders.

Figure 3.3 represents REMIX's ability to generate maps of points or even heat maps. It is also possible to query a variety of insights about trips. Insights like daily trips in a specific location of the map.

Dashboards in figure 3.4 allow the consultation of metrics so everyone can measure how their micromobility programs are performing day-to-day. For instance, to measure mobility, average vehicles on a certain day and location can be consulted.

- **Data Platform** Visualization of all relevant transportation data in an easy to use way.



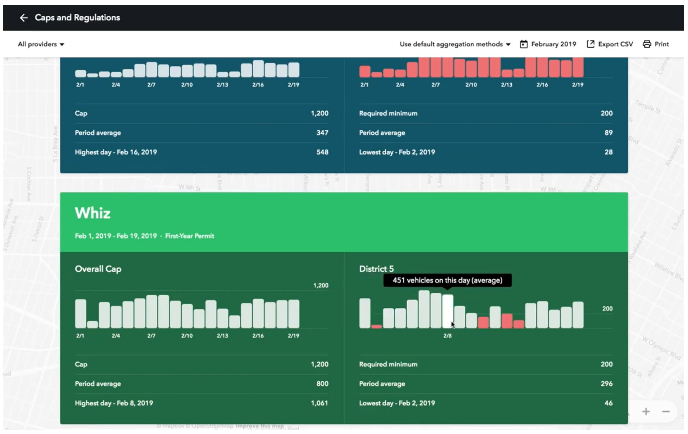
Time of day	
Morning: Before 10:00	3,321
Midday: 10:00 - 16:00	3,883
Evening: After 16:00	3,484

Council Districts		
District 1: Gil Cedillo	17	0.5%
District 2: Paul Krekorian	0	0.0%
District 3: Bob Blumenfield	0	0.0%
District 4: David Ryu	321	9.1%
District 5: Paul Koretz	957	27.2%
District 6: Nury Martinez	0	0.0%
District 7: Monica Rodriguez	0	0.0%
District 8: Marqueece Harris-D...	0	0.0%
District 9: Curren Price	0	0.0%
District 10: Herb Wesson	189	5.4%
District 11: Mike Bonin	1,805	51.3%
District 12: Mitchell Englander	0	0.0%
District 13: Mitch O'Farrell	87	2.5%
District 14: Jose Huizar	0	0.0%
District 15: Joe Buscaino	0	0.0%

7th & Berendo	
Daily trips	2
Total capacity	16
Time empty	0.0%
Time full	0.0%
Avg available bikes	10
Avg broken bikes and docks	0

Bikeshare Stations		
🟢	< 5%	time empty or full
🟢	5-10%	time empty or full
🟢	10-20%	time empty or full
🔴	20-30%	time empty or full
🔴	30-40%	time empty or full
🔴	> 40%	time empty or full

Figure 3.3: Image of Remix's Interface with a full picture of multiple modes mobility [6].



Cap	1,200	Required minimum	200
Period average	347	Period average	89
Highest day - Feb 16, 2019	548	Lowest day - Feb 2, 2019	28

Overall Cap		District 5	
Cap	1,200	Required minimum	200
Period average	800	Period average	296
Highest day - Feb 8, 2019	1,061	Lowest day - Feb 2, 2019	46

Figure 3.4: Image of Remix's dashboards [6].

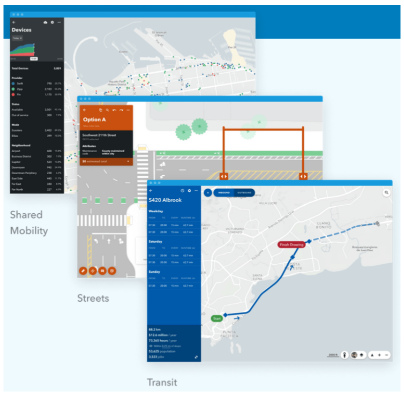


Figure 3.5: Remix's main aspects [6].

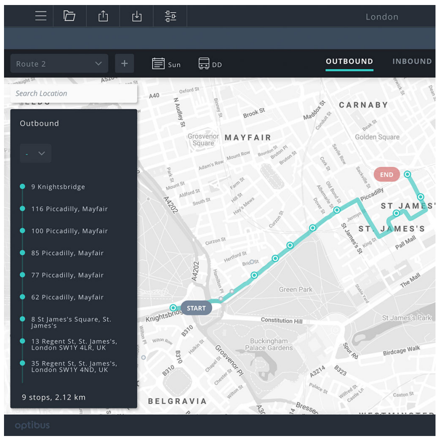
Figure 3.5 represents their main advantage - the fact that data is all gathered in one place. It shows that both shared mobility, streets and transit can all be be consulted in one platform.

3.1.2 Optibus

Optibus is a five-year-old startup from Tel Aviv (Israel) which came in as a web-based transit operations solution that creates technology to ensure better performance from the passenger's point of view.

Optibus has several modules: Route Planning and Timetables, Vehicle and Crew Scheduling, Rostering and On-Time.

Route Planning and Timetables



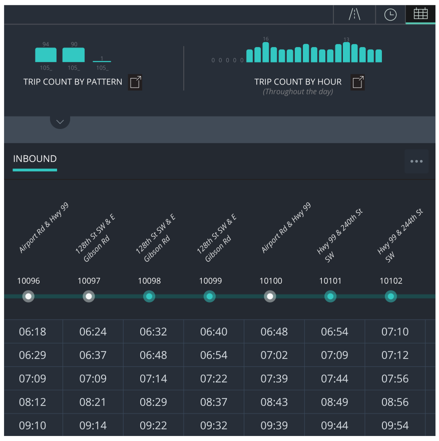
Outbound

- 9 Knightsbridge
- 116 Piccadilly, Mayfair
- 100 Piccadilly, Mayfair
- 85 Piccadilly, Mayfair
- 77 Piccadilly, Mayfair
- 62 Piccadilly, Mayfair
- 8 St James's Square, St. James's
- 13 Regent St, St. James's, London SW1Y 4LR, UK
- 35 Regent St, St. James's, London SW1Y 4ND, UK

9 stops, 2.12 km

Figure 3.6: Image taken from Optibus' website illustrating its route planning module [7].

The map represented in figure 3.6 can have satellite and street level pictures, so that their platform lets planners effectively visualize routes. Furthermore, it is even possible to verify where bus stops are located, check their picture, adjust their location or select a checkpoint so that the system will automatically create an optimal route passing through it. Also, when the map does not include planned roads, you can use the free draw feature to plan future routes. To easily understand and communicate business impact of any route change, and better optimize costs in the scheduling and rostering phases, Optibus provides many business metrics (figure 3.7).



Airport Rd & Hwy 99	128th St SW & E Gibson Rd	128th St SW & E Gibson Rd	128th St SW & E Gibson Rd	Airport Rd & Hwy 99	Hwy 99 & 240th St SW	Hwy 99 & 244th St SW
10096	10097	10098	10099	10100	10101	10102
06:18	06:24	06:32	06:40	06:48	06:54	07:10
06:29	06:37	06:48	06:54	07:02	07:09	07:12
07:09	07:09	07:14	07:22	07:39	07:44	07:56
08:12	08:21	08:29	08:37	08:43	08:49	08:56
09:10	09:14	09:22	09:32	09:39	09:44	09:54

Figure 3.7: Image taken from Optibus' website. It shows a timetable and, above, some of the provided business metrics [7].

With an interface that provides a draggable tab where users can find many metrics (figure 3.8), it becomes easier to understand the impact of timetables on service quality and their operating costs.

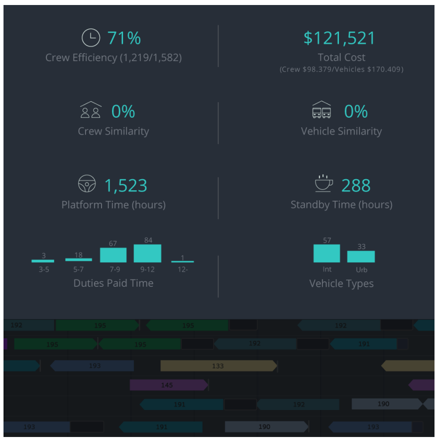


Figure 3.8: Opened business metrics tab from Optibus' website [7].

Vehicle and Crew Scheduling

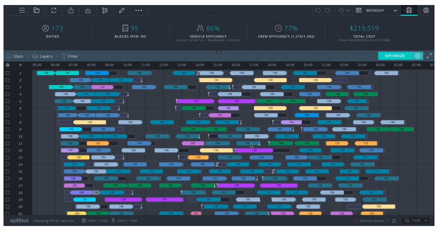


Figure 3.9: Image taken from Optibus scheduling module [7].

In figure 3.9, it is seen the Optibus Scheduling module, which is able to plan and schedule the movements of every driver and vehicle, discovering better options and therefore reducing costs.
This optimization runs in seconds to minutes and can be done under customized constraints and rules. For this purpose, their system takes advantage of machine learning algorithms.

Rostering

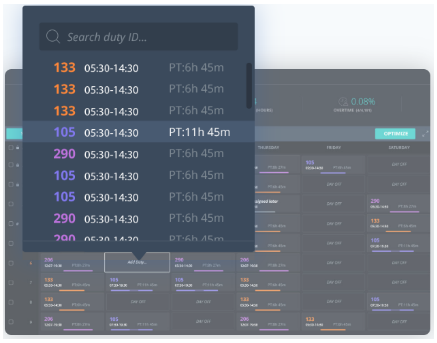

Search duty ID...
133 05:30-14:30 PT:6h 45m
133 05:30-14:30 PT:6h 45m
133 05:30-14:30 PT:6h 45m
105 05:30-14:30 PT:11h 45m
290 05:30-14:30 PT:6h 45m
105 05:30-14:30 PT:6h 45m
105 05:30-14:30 PT:6h 45m
290 05:30-14:30 PT:6h 45m
OPTIMIZE

Figure 3.10: Image taken from Optibus rostering module [7].

As seen in the previous modules, Optibus' has several optimization algorithms. These algorithms are also used in rostering (figure 3.10), saving manual work and helping them to achieve business goals.

On-Time Performance

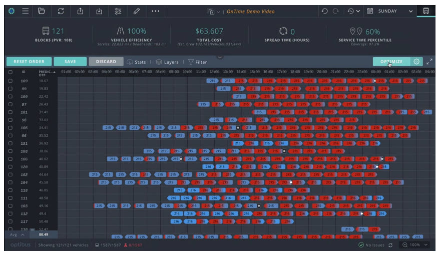


Figure 3.11: Image taken from Optibus on-time module [7].

The module represented by figure 3.11 provides to users a Gantt Diagram with trips. Each of them has a certain estimated chance of starting and ending on time. These chances are calculated by an ML algorithm which learns patterns from historical data. On-time performance is one of their metrics and it predicts the likelihood that each vehicle trip will be performed on-time. The user can then click on the optimize button to reduce the on-time performance of every trip. The resulting metrics are then shown above.

3.1.3 Remix vs Optibus Comparison

The purpose of the REMIX platform is mainly to inform user's decisions about transportation and mobility in general. It features various metrics and charts that allow the user to interpret data more easily. This data either can be statistics about mobility and holistic services or graphical representations of transport infrastructures in a map.

When we speak of Optibus, it is possible to denote a more limited map interaction in terms of moving and handling transport infrastructure. Conversely, the Optibus platform seems to excel at allowing public transit agencies to schedule their vehicles and trips using timetables and Gantt diagrams.

It follows that the Remix platform's goal is more in line with that of the internship project since both take advantage of a map to represent trips and infrastructures. While Remix allows you to interact with the infrastructures in more detail, Optibus does not, which makes Optibus distance itself from Motum. Motum focuses on the mobility of people, the application of techniques to understand this and, as an initial objective, had the removal/addition of infrastructure and consequent impact assessment. Remix does it. Optibus is more focused on planning and scheduling modes of transport, and creating hypothetical routes for them. Motum has some of the features of these two platforms, although more similar to Remix. What Motum also brings is the ability to group trips to determine the most similar population movements. In addition, it allows you to classify travel transportation and load a variety of data.

3.2 Literature Review on mobility patterns extraction

For the goal of the Internship, it is essential to infer human mobility patterns, therefore some studies on this aspect will be investigated in this chapter.

3.2.1 AllAboard - IBM's Intelligent Tool

Di Lorenzo et al. (2014) [30] presented a conference paper for IEEE with an objective resembling this study's. The researchers from IBM present AllAboard, which is an intelligent tool that analyzes cellphone data to aid cities' authorities in exploring urban mobility and optimizing public transport. The system specifically uses a mobility dataset extracted from Call Detail Records that describe call activity of a randomly chosen sample of 50,000 users every two weeks, for a total period of 20 weeks. The component of most interest in the work of AllAboard is the Mobility Miner component, which processes CDRs, and extracts information on users' stops, origin/destination flows, and shared route patterns. Their methodology consists of three steps: i) extraction of stops locations performed by each user; ii) estimation of aggregated Origin/Destination (O/D) flows between those stops; iii) extraction of shared route patterns from the sequences of stops visited by each user, that are used as candidate new routes for public transport optimization. For transit operators intended to use the system and that may not have a background in computer science, an interactive user interface was developed to allow transit operators the exploration of travel demand in both space and time, evaluate the quality of service that a transit network provides to the citizens, and test scenarios for transit network improvements. This way of presenting the tool to users is through a Web-based front-end and all the information is obtained via REST services asynchronous invocations. The estimation of O/D flows was validated through a comparison with the gravity model, which is the basic model used in transportation [31]. Optimization module was evaluated in an end-user-based evaluation at the European congress on Intelligent Transportation Systems. There, they prove that the system suggests routes that are representing areas of the city where current public transport was not well developed and also present, line by line, the ridership and travel times of the current and the proposed new network.

3.2.2 Human Activity Patterns

In a 2014 study, Demissie et al. [32] takes advantage of passive mobile positioning data like a) Calls Volume, b) Erlang and c) Handover, to infer intensity and human activity patterns throughout the day. The forms of data used were based on information like Cellphone usage, Residential buildings, Points of Interest, Bus movement, Taxi Movement and Presence of People. The two techniques applied on the information obtained off of cellular networks were (a) reconciliation of the spatial dimensions of the data and (b) normalization over time and space, via Fuzzy clustering. To validate those clusters as actual predictors of human activity they compare them with ground truth clusters of variables such as presence of people, buildings, POIs, bus and taxi movement. Therefore, their results demonstrated the potential of aggregate cellphone data in detecting density of activities that are superimposed on the different activity patterns, which they considered to be fundamental information for transportation and urban studies.

Following, Demissie et al. (2016) [33] proposed a study on this topic which is, in part, similar to this document's study. These researchers utilize data derived from cell phones to dynamically infer mobility patterns and, with that, improve transport networks planning. They take advantage of Call Detail Records (CDR) and SMS data. Furthermore, links between home and the five top destinations of each subject were defined by looking onto cell phone data, hence they can be called cell-based flows and seen as people's travel demand. For each one, the researchers defined a flow density corresponding to the level of travel demand in each link. The final aim of this study was to suggest new transit routes. Due to lack of data, they couldn't analyze the capacity of transit routes and affirm that their route recommendation is mainly based on flows between origin and destination. In the end, although they were not able to obtain all the information regarding the urban transit network of Dakar, they validate their work's results based on transit network data such as bus routes, network type, and origin and destination bus stop locations. Still in the same year, in [34], these authors produced a comparable but smaller study estimating origin-destination flows between districts of Senegal in which they use 2 different approaches: one for commuting trips (home to work) and another for irregular trips.

3.2.3 Human Mobility for Bus Stops Relocation

Also recently, Fabbiani et al. (2018) [35] published, within a book, a study with the two main goals of analyzing the mobility patterns of bus passengers and then relocating bus stops in an urban area, which are both similar to two objectives of this internship. As for analyzing mobilty patterns, these researchers obtain user demand and origin-destination matrices, although the focus of the work is the relocation of bus stops. For that purpose, they use a multi-objective evolutionary algorithm, as they were trying to find a configuration of bus stops that provides good trade-off between the quality of service and the operational costs. For this problem, they utilize various sources of data, such as bus capacity, number of bus stops, fuel cost, bus driver salaries costs, among others. Finally, bus stops relocation experimental results indicated that the proposed algorithm was able to compute improved solutions regarding both problem objectives and those improvements over the real situation of the used data were up to 16.7 % in travel time and 33.9 % in cost.

3.2.4 Mobility Patterns extraction through Clustering - Public Transportation Data

More specifically on Clustering, Mohamed Mahrsi et al. (2017) [36] proposed two approaches to cluster smart card data with the purpose of extracting mobility patterns in a public transportation system. The two approaches consisted in (a) partitioning of stations into different clusters having a similar usage (model-based count series clustering) and highlighting relationships between time of the day, location and usage, and (b) determining groups of passengers who behave similarly with respect to their boarding times (generative model-based clustering). Yet on Clustering, Thuillier et al. (2018) [37] proposed a much recent approach to address human mobility perception through mobile phone data. As data, they utilize call detail records as a real marker for human presence. Their first approach was to divide people's daily activities into six different profiles with a new profiling algorithm. Then, K-Means Clustering is applied to group individuals according to their profiles. With that, 12 different weekly patterns corresponding to the 12 main clusters were created, holding the main mobility behaviors of individuals over a territory. As result and due to validation on real data, they could demonstrate relevance in their study.

3.2.5 Mobility Studies Reviews

In an endeavor to generalize studies proposed for human mobility patterns, Rojas et al. [38] produced a review of studies that have utilized passively collected data, such as data from mobile phone networks and smartphone GPS data. As the authors convey, this review can provide an overview of the usefulness of these data in data-processing algorithms from four perspectives: preprocessing, trip end identification, trip mode detection and trip purpose identification. For these perspectives, it is stated that clustering, and in general machine learning, is an emerging approach and definitely a go-to area. With a similar purpose, Zhao et al. [39] produced an article summarizing recent studies on urban human mobility from a data mining point of view: from data collection and cleaning, to mobility models and its applications. They utilize areas from Physics to Computer Science (e.g., Machine/Deep Learning techniques such as LSTM).

3.3 Classification of trips transport mode

As in the internship it was necessary to apply Machine Learning to trips to classify them according to the modes of transport they used, some articles were studied in an attempt to obtain knowledge and know what currently exists in the literature on the subject.

In 2016, a study was proposed with the theme of trip reconstruction and transportation mode extractiom from mobile phone GPS data. Firstly, they extract trips from this data. For this, Apichon Witayangkurn et al. (2013) [40] defined four main steps: stay point extraction with outlier detection, change point detection, determining transportation mode and merging segments. The final output of these phases are stay points, walk segments and other transportation mode segments: bike, car and train. These modes of transport are labeled based on ground truth data. Transportation mode labels included stay, walk, bike, car and train. As features, these authors not only use traditional features, but also use information from GIS software, such as spatial train and road networks, which improve the classification by providing information of which space each travel segment is in. More precisely, these features are a percentage of points in train line and a percentage of points in road. In addition, to label the modes of transport for trips and to check the classification results, they developed a web-based trip visualization with Google map. For supervised learning, Decision Tree based classifier was used since outperformed other models for inferring mode such as Naive Bayes (NB), Bayesian Network (BN) and Support Vector Machines (SVM). In addition, Random Forest still outperformed Decision Trees. The classification features were Total Distance, Time Duration and percentage of points in train line and road network. Features associated with speed were also used: minimum speed, maximum speed, average speed, overall average speed, maximum acceleration and speed change rate. As technologies, Java Topology Suite (JTS) was used for spatial calculation such as finding points in geometry and spatial index for fast searching geometry and Java Machine Learning Library (Java-ML) for clustering, feature selection and classification. PostgreSQL along with PostGIS was used as the database to store GPS data and label data. Java Servlet and Google Map were finally used to develop the web-based visualization tool. The performance measures for the transportation mode classification, Precision and Recall were the chosen ones. 10-fold cross-validation was the choice for training and validation phases both with/without spatial features and with/without Bagging. The best results were obtained while using Bagging and spatial features - 87.80% and 84% for precision and recall, respectively.

Qiuhui Zhu et al. (2016) [41] produced a robust framework which can effectively and automatically identify transportation modes including car, bus, bike and walk. For that, a GPS trajectory segmentation algorithm is designed to divide them into single mode segments. That process was divided into two portions, label specification and segmentation procedure. The segmentation is designed based on logical assumptions and the output of the procedure is single transportation mode segments. Many features were used. Timeslice type was one of them, which determines whether the time the travel segment took place in a rush hour or not (they call it busy and idle). Many prominent features about velocity and acceleration were also extracted. For other features, the authors were inspired by the work of other researchers they refer to. With that, they extract some features like Heading Change Rate (HCR), Stop Rate (SR) and Velocity Change Rate (VCR). The algorithm used in the classification phase was Random Forest, as they prove to be better than Decision Tree through what has already been studied in the literature. A postprocessing method was also applied which considered the wholeness of trajectory to improve the precision of mode identification - this by replacing some misidentified modes. An overall accuracy of 82.85 % for identifying transportation modes is reached. Especially for car mode, 91.44 % precision and 92.91 % recall were reached.

3.4 Cloud Services

Serverless, or Function-as-a-Service is a concept used to describe cloud-hosted applications that utilize services to autonomously manage server-side logic and state. With this architecture we end up with a better division of concerns, cost benefits and more flexible and amenable to change system.

Nowadays, the number of applications relying on serverless architectures is greater than we all think, in such way that it is also becoming quite popular in the world of Big Data, and as researchers along with Carreira. et al. [42] state, "The serverless computing model is emerging as a compelling approach to address the data center resource management problem" and therefore "the need for developers to explicitly configure, deploy, and manage long-term compute units (e.g., VMs)" is being avoided.

Serverless computing was chosen for the architecture of the project due to a variety of factors, such as:

- No need for servers and its software provisioning, maintenance and installation
- More focus on the business logic of the application
- Pay for the time of execution instead of server units

Serverless penetration rate increased a lot in such way that recently many big companies are getting into this technology.

Google Cloud Functions was soon discarded. It was reported to have lag in its services and besides Javascript, its languages were still in beta phase.

Amazon Web Services (AWS), International Business Machines Corporation (IBM) and Azure Functions remained. All of them offer support in a wide variety of programming languages, therefore these providers will be compared henceforth.

Regarding the free-tier, all the compared cloud providers offer a million requests and 400,000 GB-seconds of computing time per month. If requests surpass this free-tier limit, they start getting billed approximately 0,000016 dollars for each used GB-s. Azure Functions' billing model was found to be the cheapest. Due to the free tiers, both AWS and Azure are acceptable for the project. The choice between them has to do only with the preferred environment and previous experience using it.

One known downside of Serverless happens when a new Lambda instance handles its first request. When a request arrives, a Lambda function checks if a container is already running

	Free-tier	**Pricing**	**Single function execution**
AWS Lambda	1M/month 400000 GB/s	0.00001667$ per GB/s	1000 executions at a time, max of 15 min
Microsoft Azure	1M/month 400000 GB/s	0.000016$ per GB/s	unlimited concurrency, max of 15 min
IBM	1M/month 400000 GB/s	0.000017$ per GB/s	unclear concurrency rate, unlimited execution time

Table 3.1: Free-tier, Pricing and Execution Time comparison between the three main providers: AWS, Azure and IBM

.

to serve the invocation. It is called "warm container" if there is an idle container available. However, if it cannot find an idle container, the Lambda will create a new one and that is what is called **"cold start"**. In this situation, the response time is increased due to the latency of starting a new container. Cold starts can then come back after some minutes have passed from the previous request.

As cold start latencies are provider and memory dependent, a comparison is presented below in Table 3.2.

Provider-Memory	**Median**	**Min**	**Max**	**STD**
AWS-128	265.21	189.87	7048.42	354.43
AWS-1536	250.07	187.97	5368.31	273.63
Google-128	493.04	268.5	2803.8	345.8
Google-2048	110.77	52.66	1407.76	124.3
Azure	3640.02	431.58	45772.06	5110.12

Table 3.2: Cold Start latencies in miliseconds. Comparison between cloud providers using NodeJS 6 [10].

AWS Lambda functions are known to achieve the best scalability and, as its proven in the table above, have also the lowest cold start latency. According to the previous comparison, the chosen cloud provider for the serverless architecture was AWS.

Chapter 4

Project Specifications

Motum's system aims to improve the planning of transport network infrastructures. For that, a front end must handle the representation of people's mobility, infrastructures of Tokyo, and back end's Data Analytics results.

This chapter aims to present the project management methodology of the developed system throughout the internship, as well as specifying its requirements. In addition, the reader will be elucidated about the system's architecture and the use cases that represent the interactions in the final solution.

4.1 Methodology

The project's methodology is based on Scrum, having been adopted according to the reality of the project, the company and the team. In this section, the workflow required by such an agile methodology will be detailed.

Based on continuous learning through the development, Scrum acknowledges that the development team, in this case singular, doesn't know everything about the desired final product at the start of the project [43]. This way, requirements need not be clearly defined at the beginning.

There are some aspects of Scrum that, given their importance, deserve due attention. Thus, adapted from Atlassian's article [43], these aspects are about to be clarified in the next sections.

4.1.1 Scrum Artifacts

Artifacts are things like documents, tools or programs produced to solve a problem in the development process. These artifacts are a product backlog, a sprint backlog and an increment, and throughout this process, the team keeps revisiting and investing in this scrum artifacts overtime.

- **Product Backlog** "To do" list with the stuff that needs to be done. Constantly revisited, re-prioritized and maintained by the Product Owner, which in this case, is the Internship Advisor from Grama. This list may contain features, requirements, enhancements, and fixes that act as input to the sprint backlog.

- **Sprint Backlog** List of items, user stories, or fixes, selected for implementation in the current sprint cycle. The list is defined in the planning meeting of the sprint and its final goal cannot be compromised.
- **Increment** Usable end-product from a sprint. However this may not be realistic to some teams, since there are teams that decide to release something to their customer at the end of each sprint.

4.1.2 Scrum Roles

There are the following three Scrum Roles: product owner, scrum master and the development team. Every Scrum role has key responsibilities that they must fulfill.

- **Scrum Team** The members of a team are responsible for organizing themselves in order to get the job done and complete their goals. Within the team can be several people from different positions within the company and they must meet everyday at the daily scrum for the sake of transparency during sprints.
- **Product Owner** Responsible for identifying the requirements, converting them into a prioritized list and deciding which one should be at the top of the list for the next Sprint. This list is called Product Backlog and must be continuously updated.
- **Scrum Master** Acts between the product owner and the team, helping both parties. Helps the product owner to define value and the development team deliver the value. Some of their activities include helping the owner planning the work, managing the backlog, and helping the team managing blockers, getting to the increment and self-organizing.

Since this project is part of an individual internship, the team has only one member, which is the intern. The product owner is the company's internship advisor, Eng. Rafael Maia and the scrum master is Eng. Miguel Oliveira.

4.1.3 Scrum Events

Scrum Teams must perform sequential meetings on a regular basis. These meetings are called Scrum Events and the following were considered important for the project development.

- **Sprint Planning** Planning of the work to be done throughout the current sprint. Here, the development team establishes the sprint goal.
- **Sprint** The time period in which the team works in an increment.
- **Daily Scrum** A small, daily meeting (usually about fifteen minutes) that always happens at same time of the day. This meeting has two main goals: (a) to put everyone involved in the project on the same page, and (b) to decide a plan for the next 24 hours.
- **Sprint Review** When the sprint has finished, the intern groups up with his advisor for an informal meeting to see demonstration of the the increment. Some items of the backlog list finally get a status of "Done" and stakeholders can now give feedback.

This increment is optionally released. The Internship advisor updates the backlog to the next sprint.

- **Sprint Retrospective** The intern gets together with his advisor to document and argue about what went right and wrong. The outcome of this retrospective should be to define what needs to be refined for the next time.

Succinctly, the Intern adopted a project methodology with two-week sprints and Daily scrums. Moreover, on the first day of each Sprint, the Intern and his Advisor cooperatively planned the Sprint. Also that day, the Review and Retrospective about the last sprint took action.

4.2 Requirements

In this section, a specification of the Software Requirements is presented.

4.2.1 Introduction

According to IEEE standard 729-1983 [44], a requirement is defined as:

- A condition or capability needed by a user to solve a problem or achieve an objective
- A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents
- A documented representation of a condition or capability as in 1 and 2.

4.2.2 Functional Requirements

These requirements were defined iteratively by the company's Internship Advisor and were adapted over the duration of the project as needed or difficulties arose. Some requirements have undergone necessary changes to be completed. The final requirements are found in the apendices 7.2.

4.2.3 Non functional Requirements

This type of requirements defines web application's attributes, which can be constraints that amplify its functionalities [45]. This system should fulfill some non functional requirements.

Usability

Usability, by Jakob Nielsen [46], is the quality attribute that evaluates how easily a user interface can be used. He divides Usability into five elements that he calls **attributes**, which can be defined by questions:

- **Learnability:** How easy is it for users to perform basic tasks in their first contact with the design?
- **Efficiency:** After users have learned how the design is, how quickly can they get things done?
- **Memorability:** When users return to design after a period without using it, how easily can they restore proficiency in that design?
- **Errors:** How many errors do users make? How serious are the errors? How easily can they stop making those errors?
- **Satisfaction:** How enjoyable is it to use the design?

To study the usability of the system produced in this work, it is necessary to define who are the users of the web application and then perform user testing [46]. Therefore, three steps are required:

- Select some users who represent people who will use the web application, such as planners for a city's transport network infrastructure.
- Formulate tasks that are appropriate for usability testing of the design. These tasks must be realistic, actionable and the user cannot have clues of how the task is performed.
- Silently observe what the user does. Assess the success of the task and the difficulties the user had with the web application interface.

Security

As the entire system contains the personal data of registered users, it is extremely important to keep that same data safe, by blocking unauthorized access. Furthermore, it is necessary to ensure that only duly authenticated users are allowed to change the data stored in the database [45]. To ensure security, an Amazon Web Services (AWS) service called Cognito will be used, which allows users to store users' data securely and assure their authentication. The HTTPS protocol is used for the communication between the client and server in the API Gateway and CloudFront, in order to protect the communication between them. Also, in order to verify the security of the system, OWASP Top Ten [47] shall be used. It consists of a standard web application security awareness document, which represents a broad consensus on the most critical security risks for web applications.

Performance

Performance consists of the speed with which the web application returns the answers to the user during its use and under certain workload [45]. There are several ways to measure performance. For example, Jakob Nielsen [48] defined 3 main metrics for response time in 1993. Nowadays, the metrics are still meaningful since they are based on how the human attention works:

- 0.1 second: if the system exceeds this time limit, its response does not appear to be instantaneous.

- 1 second: limit when the user already notices the delay, but that does not interrupt his line of thought.
- 10 seconds: user completely loses attention.

There are other performance metrics for web applications that can be measured, such as Server Time and Render Time [49]. In order to measure these values, tools will be used. For instance, HTML events such as *window.onload* and *DOMContentLoaded* can be used to measure web loading times [49].

4.3 Actors

This chapter presents the actors for the Transportation Network Infrastructure Planning web application.

The following types of actors can access the application:

- Platform User
- Data Provider
- Data Consumer

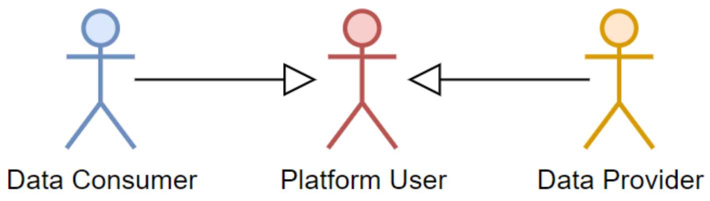


Figure 4.1: Actors that will use the Transportation Network Infrastructure Planning web application.

Actor	**Description**
Platform User	The User Platform represents an actor only containing the basic privileges. These are inherited by both the Data Provider and the Consumer
Data Provider	The Data Provider is a person or business who provides mobility or transportation infrastructure data
Data Consumer	The Data Consumer is a person or business who is looking for solutions on transportation network infrastructures planning

Table 4.1: Description of each Actor present in the Web Application.

4.4 Use Cases

In this project, there are use cases for each of the actors already presented.

In the further subsections, each one of these use cases is detailed and exemplified with Unified Modeling Language (UML) diagrams - more specifically Use Case Diagrams.

4.4.1 Platform User

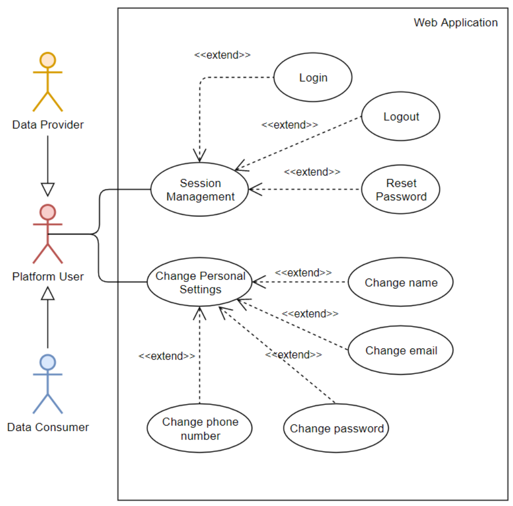


Figure 4.2: Platform User actor: Session Management and Change Personal Settings use cases.

Platform Users can manage the lifecycle of their user session within the web application along with changing their personal settings. As it is noticeable in figure 4.2, Data Provider and Data Consumer actors inherit all these use cases.

4.4.2 Data Provider

At high-level, Data Provider actor manages all the historical data.

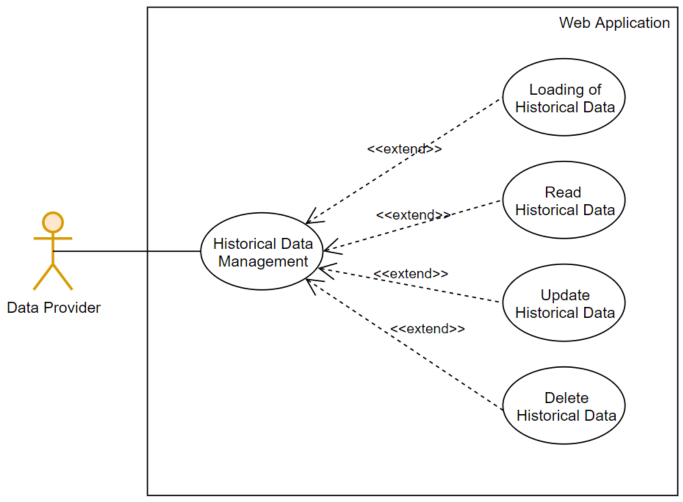


Figure 4.3: Data Provider actor: Historical Data Management use cases.

Data Provider provides the web application with Create, Read, Update and Delete (CRUD) operations on Historical Data. As seen previously, this actor also needs user session management.

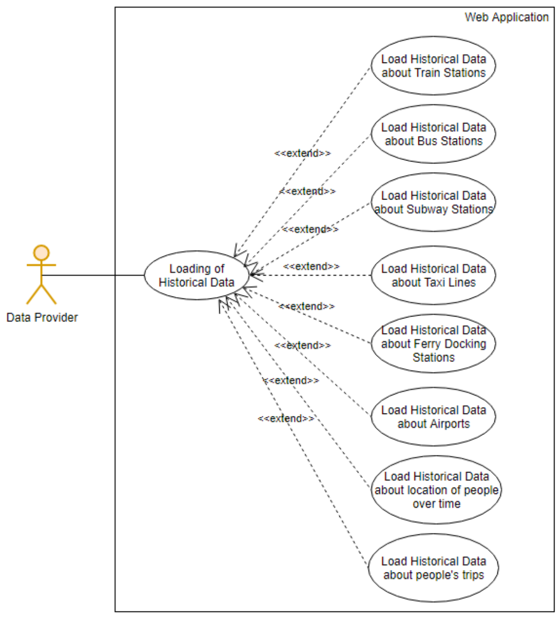

Web Application
Data Provider
Loading of Historical Data
<<extend>>
Load Historical Data about Train Stations
Load Historical Data about Bus Stations
Load Historical Data about Subway Stations
Load Historical Data about Taxi Lines
Load Historical Data about Ferry Docking Stations
Load Historical Data about Airports
Load Historical Data about location of people over time
Load Historical Data about people's trips

Figure 4.4: Data Provider actor: Loading of Historical Data use cases.

Loading of Historical Data

Figure 4.4 represents the use cases related to the loading of all types of Historical Data. These are use cases that belong to the Data Provider actor.

This Historical Data can be Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations, Airports and Mobile Device Events.

Read Historical Data

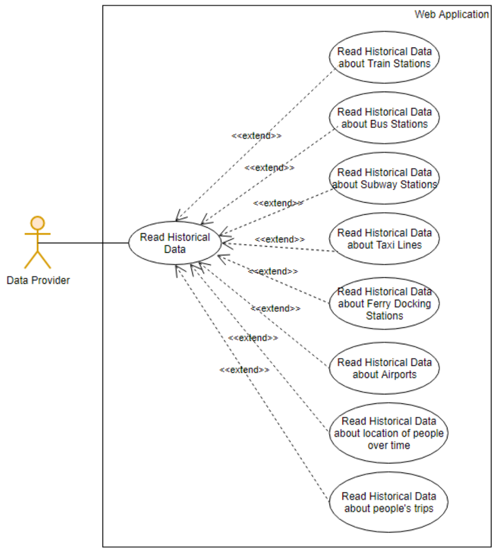

Web Application
Data Provider
Read Historical Data
<<extend>>
Read Historical Data about Train Stations
Read Historical Data about Bus Stations
Read Historical Data about Subway Stations
Read Historical Data about Taxi Lines
Read Historical Data about Ferry Docking Stations
Read Historical Data about Airports
Read Historical Data about location of people over time
Read Historical Data about people's trips

Figure 4.5: Data Provider actor: Read Historical Data use cases.

Figure 4.5 represents the use cases related to the query of all types of Historical Data.These are use cases that belong to the Data Provider actor.

This Historical Data can be Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations, Airports and Mobile Device Events

Update Historical Data

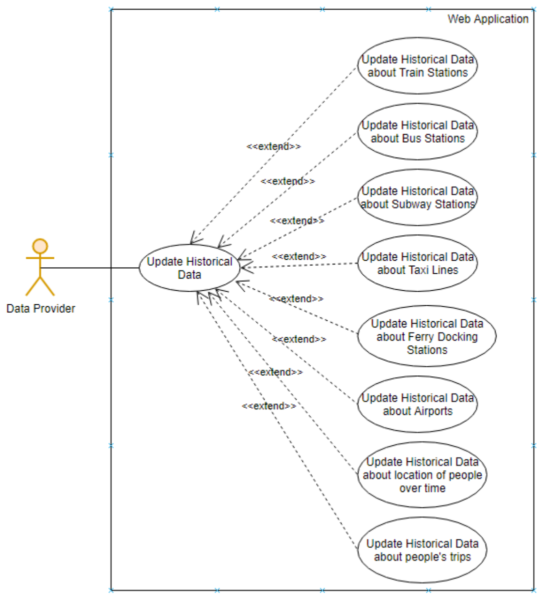

Web Application
Data Provider
Update Historical Data
<<extend>>
Update Historical Data about Train Stations
Update Historical Data about Bus Stations
Update Historical Data about Subway Stations
Update Historical Data about Taxi Lines
Update Historical Data about Ferry Docking Stations
Update Historical Data about Airports
Update Historical Data about location of people over time
Update Historical Data about people's trips

Figure 4.6: Data Provider actor: Update Historical Data use cases.

Figure 4.6 represents the use cases related to updating of all types of Historical Data. These are use cases that belong to the Data Provider actor.

This Historical Data can be Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations, Airports and Mobile Device Events.

Delete Historical Data

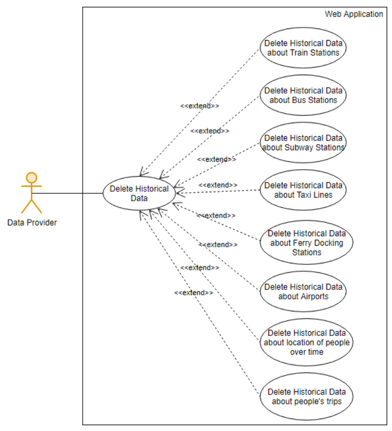

Web Application
Data Provider
Delete Historical Data
<<extend>>
Delete Historical Data about Train Stations
Delete Historical Data about Bus Stations
Delete Historical Data about Subway Stations
Delete Historical Data about Taxi Lines
Delete Historical Data about Ferry Docking Stations
Delete Historical Data about Airports
Delete Historical Data about location of people over time
Delete Historical Data about people's trips

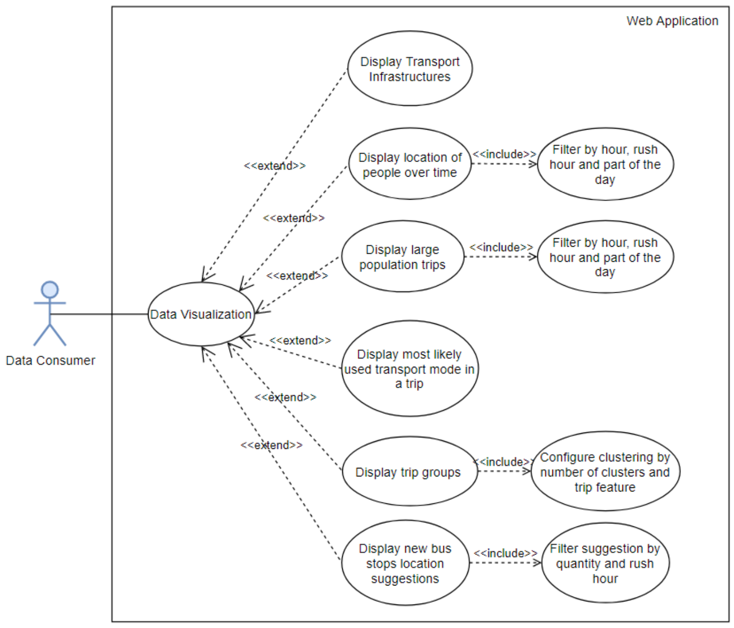
Figure 4.7: Data Provider actor: Delete Historical Data use cases.

Figure 4.7 represents the use cases related to the deletion of all types of Historical Data. These are use cases that belong to the Data Provider actor.

This Historical Data can be Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations, Airports and Mobile Device Events.

4.4.3 Data Consumer

At high-level, Data Consumer actor can visualize both Tranportation Infrastructures and Populations Movement related data.



Web Application
Data Consumer
Data Visualization
Display Transport Infrastructures
Display location of people over time
Display large population trips
Display most likely used transport mode in a trip
Display trip groups
Display new bus stops location suggestions
Filter by hour, rush hour and part of the day
Filter by hour, rush hour and part of the day
Configure clustering by number of clusters and trip feature
Filter suggestion by quantity and rush hour
<<extend>>
<<include>>

Figure 4.8: Data Consumer actor: Data Visualization use cases.

Figure 4.8 represents Data Consumer actor's activities. Based on historical data, Data Consumers can visualize transportation infrastructures and large population movements in a map. Moreover, information about population movement trend and most likely used infrastructure in a movement can be granted to a Data Consumer actor. Ultimately, based on historical data about population movement, these actors can get access to suggestions of new bus stops locations.

4.5 Architecture

To describe all communications between the various parts of the application, the model chosen to represent the architecture of the back end was the C4 model [50].

Firstly, in figure 4.9 we have the System Context Diagram that allows us to look at the big picture. The main system is represented as a box in the center of the diagram surrounded by its users and systems that interact with it. Here the focus is on systems and actors rather than technologies and protocols utilized. In this diagram, you can see the users who can interact and how they can interact with the application. The two users' connection to the Motum system lists the interactions that each one of them can have.

Then, the Container Diagram in figure 4.10 illustrates the high-level technology choices. Containers can host code or data, and can be web or desktop applications, mobile applications, databases or even file systems. This diagram shows how containers communicate

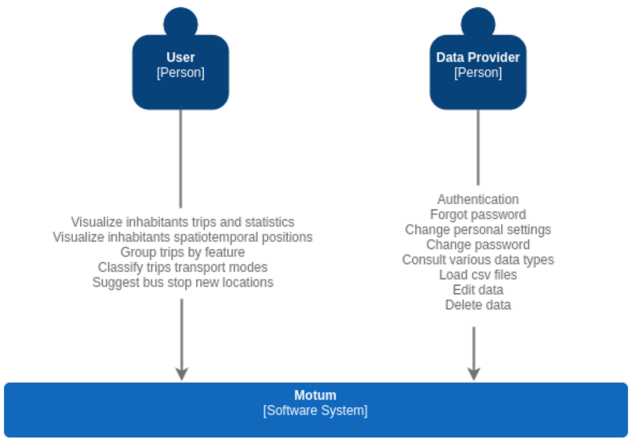


Figure 4.9: System Context Diagram.

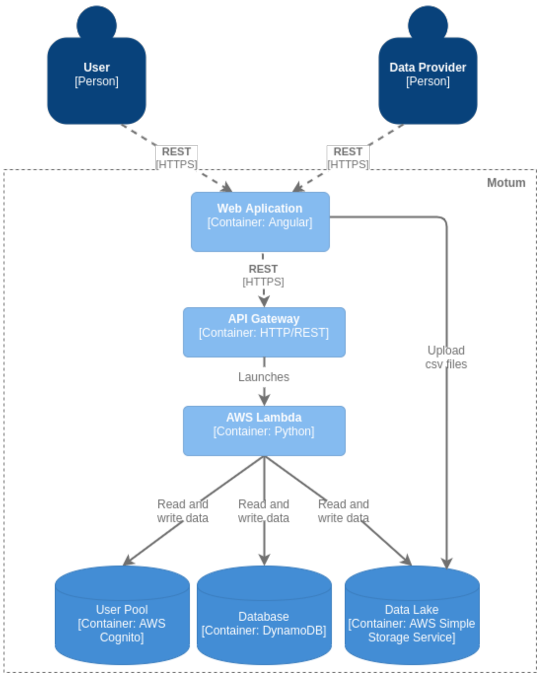


Figure 4.10: Container Diagram.

with each other. In this case, it is possible to see users who use web applications to establish contact with the API gateway, that launches lambdas in containers to deal with what the user desires. Solving user requests may require the use of DynamoDB, S3 and AWS Cognito services.

Finally, we have Component Diagram in figure 4.11 that decomposes the AWS Lambda container to a smaller level. The implemented functionalities are identified in components along with the technologies they utilize and responsibilities they have. As the name microservices says, each of these lambda functions deals with an aspect of the application that users may need. In figure 4.11, each component has a name that portrays its responsibility, together with the technology it uses for it. For example, a user that is using a web app to upload a file of train stations, indirectly launches a lambda function that grabs the file from the S3 and puts it into the train stations table at DynamoDB; and that is explained through arrows.

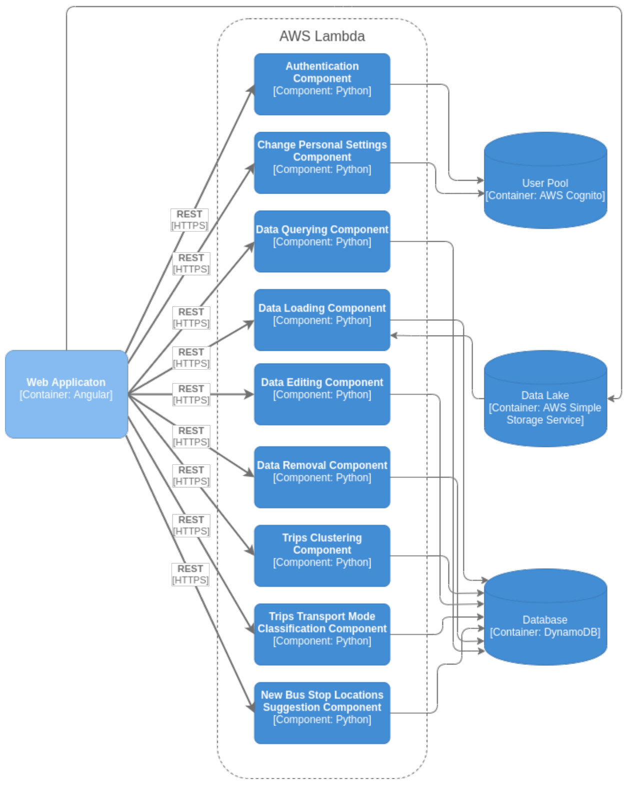

AWS Lambda
Authentication Component [Component: Python]
Change Personal Settings Component [Component: Python]
Data Querying Component [Component: Python]
Data Loading Component [Component: Python]
Data Editing Component [Component: Python]
Data Removal Component [Component: Python]
Trips Clustering Component [Component: Python]
Trips Transport Mode Classification Component [Component: Python]
New Bus Stop Locations Suggestion Component [Component: Python]
Web Applicaton [Container: Angular]
REST [HTTPS]
User Pool [Container: AWS Cognito]
Data Lake [Container: AWS Simple Storage Service]
Database [Container: DynamoDB]

Figure 4.11: Component Diagram.

4.6 Technologies

Technological choices were required during the application development; therefore, in this section, these choices are enumerated. Later, these will also be clarified.

4.6.1 Project Management

Given the complexity of the project, table 4.2 lists all the tools whose objective is to help in project management.

Technology	Function	Description
Slack	Communication between people involved.	Instant messaging cloud-based platform developed by Slack Technologies.
Trello	Tasks organization.	Web Application for project management and distribution of tasks.
Git	Version control system.	Version control system used mainly in software development, which can be used to record the history of any type of file.

Table 4.2: Project management tools.

4.6.2 Back-End

In table 4.3 are listed all the technologies used in the back end together with their function and description.

Technology	Function	Description
Python	AWS Lambda Functions programming language to respond to users requests.	Interpreted, object-oriented, high-level and easy to learn programming language with dynamic semantics.
AWS DynamoDB	Data sets storage in tables.	Key-value, fully managed and document database with built-in security, backup and restore.
AWS Simple Storage Service (S3)	Data Lake to store any data amount, at any time and from anywhere in the web. Specific use: data set storage before DynamoDB insertion.	Object storage service that offers industry-leading scalability,data availability, security, and performance.
AWS Lambda	Services Invocation through code (such as DynamoDB, S3,...)	Code execution without provisioning or managing servers.You pay only for the compute time you consume - there is no charge when your code is not running.

Table 4.3: Back-end Technologies.

4.6.3 Front End

In table 4.4 are listed all the technologies used in the front end together with their function and description.

Technology	Function	Description
Angular 8	Aiding on the creation of a dynamic web application.	Client-side TypeScript framework for building mobile and desktop applications.
HTML	Structure of the Web pages.	Standard Hypertext MarkupLanguage for browser documents design.
TypeScript	Logic of the Web pages.	Scripting superset of JavaScript which is compiled to JavaScript.
CSS	Styling of the Web pages.	Cascading Style Sheets describe how elements should be rendered on screen.

Table 4.4: Front-end Technologies.

4.6.4 Machine Learning

In table 4.5 are listed all the Machine Learning technologies used in this work together with their function and description.

Technology	**Functionality**	**Description**
Python	Programming language.	High level and interpreted programming language for a large number of uses.
scikit-learn	Machine Learning models development.	Python library for Machine Learning. It features tasks such as Classification, Regression, Clustering, Preprocessing and Dimensionality Reduction.
Pandas	Data processing and analysis.	Python library suited for analysis and manipulation of data. It provides high-performance and easy-to-use data structures and operations.

Table 4.5: Machine Learning Technologies.

Chapter 5

Development

In this chapter, the development of two components of the work is presented: Data Analytics and Data Visualization. For that, it was also necessary an introduction to the data set used, Open PFLOW [8]. In Section 5.1 everything related to the data set used, such as its validation, the creation of a new trips data set and its exploratory data analysis are presented. Then, in Section 5.2 the development process of the machine learning algorithms responsible for both the grouping of trips and the classification of trips transport modes is described and, in addition, the results obtained in the tests carried out throughout the process are shown. In Section 5.3 the process that the intern went through that led to a suggestion of bus stops is explained, from the first research on the subject that caused a change in requirements to the implementation of the suggestion method itself. Finally, in Section 5.4 the development of the web platform, how it can be used and the integration of machine learning models in the web platform are described.

5.1 Data set

In this section the validation of the mobility data set used in the project is presented. Then, the way the trips data set was extracted is explained and finally, an exploratory analysis of data is performed.

As it was predicted to be a possibility, in the Risk Analysis 1.5, the Call Detail Records (CDR) data set was not obtained. CDR gives information about the position of mobile phones in the carrier network which is applied to transport models, and to describe daily trip chains and intermodal connectivity [51]. That said, it was necessary to search for data sets, either CDR or Global Positioning System (GPS) locations that would give information on the location of people throughout the day and that had already been validated and anonymized. These searches were done mainly by contacting with people of interest and research on the internet. An artificial data set that depicts the mobility of people in Tokyo was obtained. It can in fact replace the Call Detail Records that were originally intended, as both provide similar information - the location of people over time. Call Detail Records provide information on irregular calls made over a phone service as well as the location from which they are made [51], while the Open PFLOW data set has the location of people over time with a constant period of 1 minute. Another difference is that the locations present on a CDR do not normally belong to the exact location from which calls are made, but to the nearest cell towers. In the case of Open PFLOW, the locations are exact. An advantage that Open PFLOW has is that the data has already been labeled with regard to the modes of transport that people are using at each timestamp.

Open PFLOW [52] is an artificial data set that represents the typical people mass movement of around 30 million people in Tokyo. More specifically, the data set contains the location of people every 1 minute over the course of a day. Real person's travel record can't be inferred from the data set since all the individuals in the data set were generated randomly in specific areas and their travel routes were generated using a simulator. Thus, there's no private policy concerns. At the moment, the data set is only available for Tokyo, although it is also being prepared for Chukyo and Hanshin.

The Open PFLOW data set contains the following features:

- id: unique agent id (int)
- time: YYYY/MM/DD HH:mm:ss(char[])
- longitude: SRID-4326 [1] (double)
- latitude: SRID-4326 (double)
- transport: STAY-99, WALK-1, VEHICLE-2, TRAIN-3, BICYCLE-4 (int)
- magnification factor: (double)

The magnification factor has to do with an average expansion factor strategy the authors adopted for population projection since the Person Trip Survey Data that they use developed a set of expansion factor on each surveyed person based on demographic attributes to expand samples to match the 2010 Population Census of Japan results, however, due to private policy, the version of expansion factor set was not publicly provided. Comparing the total population in Tokyo metropolitan area (nearly 30,000,000) and 500,000 behavior pattern amounts from Person Trip Survey Data, they have calculated an average expansion factor. Due to those values, in the study they conclude the expansion factor should have a value rounding 60 [8].

That said, their correlation with data from reality had to be analyzed in order to decide whether they could be used or not. The validation of the data is present in the paper that the authors of the data set wrote. That paper compares the data set with commercial GPS data and census data to evaluate its accuracy.

5.1.1 Evaluation

The authors had no ground truth data about people located in urban areas, so they compared the Open PFLOW data set with several commercial and census data to evaluate the accuracy of the data set in various aspects.

Mesh-based accuracy

For this aspect, congestion Degree Map, also called Traffic Congestion Statistics from ZENRIN DataCom Co. Ltd. was used as the hourly 500 m mesh population distribution. Error with RMSE and correlation per average mesh population were evaluated. They also utilized an original person trip survey result as the baseline to compare the performance.

Figure 5.1 shows the hourly RMSE and correlation coefficient of mesh population in Tokyo of the proposed method and a comparative result based on closed PT survey. The authors

[1]SRID-4325 declares a geographic spatial reference system [53]

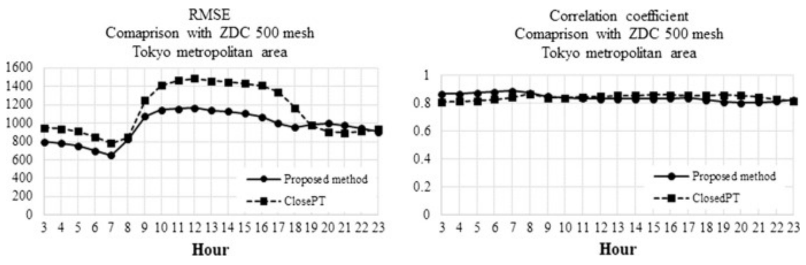


Figure 5.1: Comparison of accuracy of mesh population in Tokyo metropolitan area and Person Trip data, by using RMSE and correlation coefficient [8].

show that the results show a clear and strong relationship between the Open PFLOW data set and the congestion degree data. Moreover, the performance of the Open PFLOW data set is "similar to that of the original closed PT survey result, which indicates that the proposed method reconstructs realistic people mass movement from aggregated open data". However, it was found that the correlation of mesh population between Open PFLOW data set and congestion degree data decrease during working time. Then it was concluded that the Open PFLOW data set "can effectively reflect hourly population distribution and its changes in urban areas".

Link-based accuracy

Link traffic volume was also compared with the results of traffic censuses conducted in 2005 and 2010. This was done in order to examine how many people are moving in a road network. The authors define traffic volume as the number of people who traverse a specific road in one hour. Figure 5.2 presents the hourly traffic volume for the Open PFLOW Tokyo data set. Traffic volume in Tokyo's Open PFLOW data set was found to be congruent with the 2010 census' traffic volume per hour of the day.

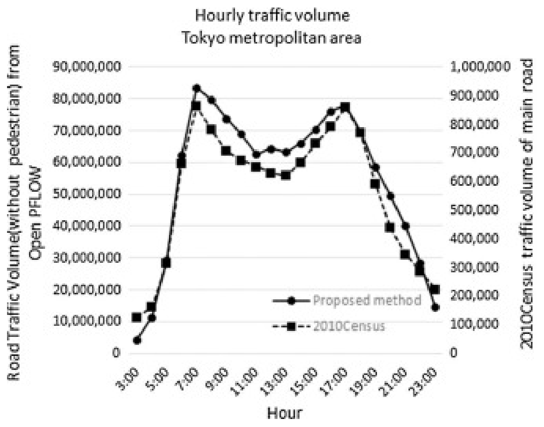


Figure 5.2: Comparison of hourly traffic volume in the main road of Tokyo and 2010 traffic census [8].

Accuracy of number of users for each transportation mode

Number of users per each transportation mode was also an aspect to evaluate. As comparative data, Person Trips' survey was chosen since it has a record of every trip's transportation mode. Figure 5.3 shows that, in the Open PFLOW data set of Tokyo area, the transportation modes "Train" and "Bicycle" are highly correlated in relation to the PT survey results, while "Vehicle" mode users in Open PFLOW is higher than that of the PT survey. They concluded that that happened due to PT survey reporting on private vehicles and some public transportation vehicles such as buses and taxis.

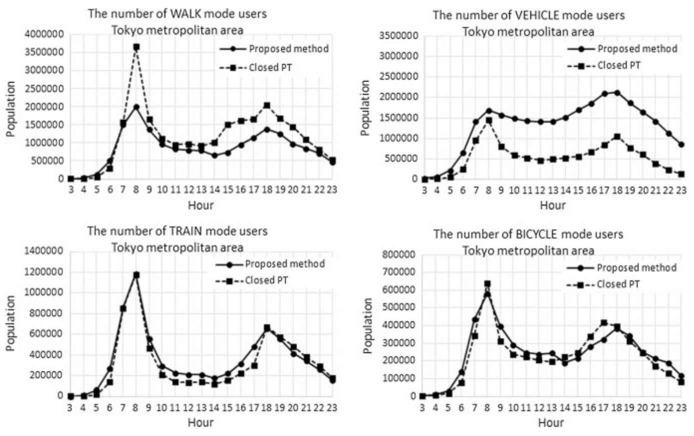


Figure 5.3: Number of users accuracy for each transportation mode in Tokyo [8].

Conclusion

Several aspects were compared with commercial data and traffic censuses in order to assess the accuracy of the Open PFLOW data set. The correlation coefficient between the Open PFLOW data set and commercial data was quite high. At rush hour periods - 6-10 a.m. and 4-8 p.m - the traffic volume of the Open PFLOW data set was closer to the traffic census than at other times of the day. Therefore, they concluded that the Open PFLOW data set can be considered reliable to depict people movement in urban area.

Some factors were nonetheless found to affect Open PFLOW's data set accuracy:

- The accuracy of the commercial data sets is not known, so it is difficult to assess how suitable our data set is for depicting actual people movement.
- The open data sets used to generate Open PFLOW data set are generated from census or survey results, which are normally done every 5 or 10 years.
- The multiple open data sources that generated the Open PFLOW data set were not collected in the same periods, which could cause extra error and affect the final accuracy.

Even so, the Open PFLOW data set brings great value by representing people mass movement, that is, each individual's location within a fixed time interval. The following excerpt from the data set paper refers to studies such as this one that require the locations of individuals to simulate people movement and advises the use of Open PFLOW for that very reason: "Users can modify or aggregate data in any format for their purpose. This provides high flexibility for researchers from different backgrounds to apply the data to various fields. Further, for specific issues, studies require the locations of individuals to simulate people movement. This data set is suited for such conditions." This is due to the fact that the accuracy level of the Open PFLOW data set is highly congruent with the real movement of people and that no real person's travel records can be inferred from the data set.

5.1.2 Trip Detection

As one of the objectives was to assess the large movements of the population, trips had to be detected from spatiotemporal positions of individuals. These trips were isolated through the sequence of transport modes people use. **Trip** will be from now on used as the concept of a movement from point of origin to the point of destination where an activity is being performed [54]. This trips are composed of various trip segments of 1 minute due to the structure of the Open PFLOW data set.

This trip extraction was facilitated by the fact that this is a specific case in which the transport modes are already assigned to each trip, although there are known cases in which the transport modes are first classified and then the trips are extracted, or the other way around - as can be seen in chapter 3.

The Open PFLOW data set comes with the already explained per minute features - from 3 am to 11 pm. People's locations were at the level of latitude and longitude, which were considered to be at a low granularity for two reasons: a) the objective of the study was in terms of large population movements and b) they were locations that were too specific for the ML algorithms to detect patterns. That said, the latitude and longitude of each geographical position were converted to a cell in a grid over Tokyo. These square-shaped cells were created with the intention of representing zones in Tokyo. The cell size used was 500x500 meters.

From a pre-chosen origin point with geographic coordinates (lat, long) where lat = 34.906781 and long = 138.981390, two imaginary and perpendicular axis were created - the abscissa and the ordinate. This point was defined as the origin point, as it was a lower left point in the Tokyo metropolitan area, which meant that a plane with that point as its origin could cover the entire area. For the comparison of geographical and cartesian coordinates, longitude equals x and latitude equals y, so that, for a known geographic point α, the distance from α to the point with equal latitude in the imaginary axis of the ordinates is calculated. Similar is done to the longitude: the distance from this α to the point on the axis of the imaginary abscissa where the longitude is equal is calculated.

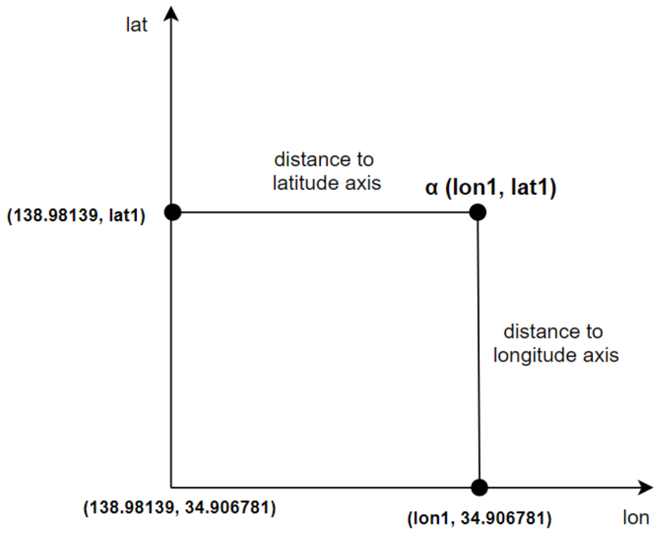


Figure 5.4: Imaginary plane that aids in the cell origin and destination IDs calculation.

Figure 5.4 visually represents the variables used in the whole process. In this way, two distances are obtained (rounded to the bottom kilometer), which represent the new coordinates of that point in an imaginary frame. Thus, these distances, in kilometers, assume that each cell will be 1 kilometer long on its side. Finally, it is necessary to divide these distances by the desired cell size - in this case, 0.5 kilometers. As the curvature of the earth must be taken into consideration, the distances were calculated based on the distance between circles of latitude and between circles of longitude [55]. Latitude circles are 111.3 km apart, which differs from the distance between longitude circles, which are never at the same distance from each other and 71 km is the average distance between two longitudes [56]. Consequently, the distance between two points can be calculated as follows:

$$avgLat = \frac{\alpha.latitude + originPoint.latitude}{2} * \frac{\pi}{180} \tag{5.1}$$

$$distanceToY = \frac{111.3 * \cos(avgLat) * (\alpha.longitude - originPoint.longitude)}{cellSize} \tag{5.2}$$

$$distanceToX = \frac{111.3 * (\alpha.latitude - originPoint.latitude)}{cellSize} \tag{5.3}$$

This distance calculation does not take into account Earth's spherical surface, but for short distances, it is equivalent to use this or another one with better accuracy, such as Haversine distance. An experimentation that compared different distance formulas for short and long distances was found [57]. Haversine was first being used - as it has more accuracy - but later in the project, it was not possible to revert the x and y of the cells back to the initial geographic coordinates.

At the end of the conversion, each location would then be represented by a cell with x and y coordinates - x corresponding to the distance to the imaginary horizontal axis and y to the distance to the imaginary vertical axis.

These records, which contain one minute of periodicity, were then iterated to create the trips data set. A STAY followed by the use of a mode of transport marks the start of a trip and , two consecutive STAYs are considered to be the end of a trip. The reason why two minutes of stopover was considered to be the end of a trip was primarily due to common sense. Cich et al. (2015) [58] used a cluster of potential stops between trips. These are evaluated by the time the person was standing there (stopDurationThreshold). If this time is less than this threshold, these locations belong to trips, otherwise they are considered stops. The threshold in this project was defined as two minutes, as two consecutive STAYs are considered to be the end of a trip (line 19 of pseudocode 2).

Most of the times, at 3 am, people are stopped, at home, although cases have also been discovered in which a person at 3 am was already using a transport mode (line 39 of pseudocode 2). In this edge case, the 3 am location was considered to be the origin of the movement. There are trips that can never happen that the person is stopped for 1 minute but still represent several different trips. In these cases, the means of transport changes without a STAY appearing among them. For this, the developed algorithm takes these cases into account and accounts them as several trips, instead of, for instance, accounting each one of these as a single trip of a multimodal type.

From the timestamp that each record had, it was possible to derive some features. Each time a person starts using a transport mode, a start time feature for the trip is saved (line 31 in 2). In the same way, the end time was also saved, but it was noticed that it did not bring much information. Furthermore, we found that in Tokyo a certain time can be considered a rush hour or not. According to the Open PFLOW data set's article used [8], these rush hours were between 7-9am and 6-7pm. Also from the timestamp, the part

Algorithm 2: Trip Detection algorithm

```
1  $dataSize \longleftarrow count(mobilityData)$;
2  $n \longleftarrow 0$;
3  while $n < dataSize$ do
4      if $userChanged$ then
5          if $transportHasBeenUsed$ then
6              $updateVariablesAndWriteTripToFile()$;
7          end
8      end
9      $cell \longleftarrow convertLatLongToCell()$;
10     if $transport = STAY$ and $not\ transportHasBeenUsed$ then
11         if $not\ originFound$ then
12             $originCell \longleftarrow cell$;
13             $tripDistance \longleftarrow 0$;
14             $tripStartDate \longleftarrow timestamp$;
15             $originFound \longleftarrow true$;
16         end
17     else if $transport = STAY$ and $transportHasBeenUsed$ then
18         $stayCount = stayCount + 1$;
19         if $stayCount = 2$ then
20             $partOfTheDay, rushHour \longleftarrow$
                 $calculatePartOfTheDayAndRushHour(timestamp.hour)$;
21             $destinationCell \longleftarrow cell$;
22             $tripDistance \longleftarrow$
                 $tripDistance + haversine(lastLocation, actualLocation)$;
23             $minutesElapsed \longleftarrow tripEndDate - tripStartDate$;
24             $averageVelocity \longleftarrow tripDistance/minutesElapsed$;
25             $writeTripToFile()$;
26             $originCell \longleftarrow cell$;
27             $tripDistance \longleftarrow 0$;
28         end
29         $resetAllVariables()$;
30     else if $transport \in [VEHICLE, TRAIN, BICYCLE, WALK]$ then
31         $partOfTheDay, rushHour \longleftarrow$
             $calculatePartOfTheDayAndRushHour(timestamp.hour);$
32         $tripDistance \longleftarrow$
             $tripDistance + distanceBetween(lastLocation, actualLocation);$
33         if $normalTripStarting$ then
34             $tripStartDate \longleftarrow timestamp;$
35         else if $multimodalTripEnding$ then
36             $updateVariablesAndWriteTripToFile();$
37             $originCell \longleftarrow cell;$
38             $tripDistance \longleftarrow 0;$
39         else if $userStartsDayTraveling$ then
40             $originCell \longleftarrow cell;$
41     $n{+}{+}$
42 end
```

of the day on which the trip was started was determined, thus having three parts of the day: morning, afternoon and night. While the duration of a trip was easily extracted by subtracting the final and initial timestamp of the trip, the distance of a trip was extracted in a more complex way. At first, this distance was calculated using the Haversine distance between the origin and destination of each trip, which did not reveal much about the actual distance traveled during the trip. The distance traveled during the whole trip was then calculated, by adding the distances minute by minute. Then, tests were performed using Decision Tree since its training is fast. It was immediately noticed that results improved with the new distance feature. In addition, based on the distance and duration of each trip, an average velocity feature was created. There were then three features that would help a lot in the transport mode classification phase. Finally, the features of the trips data set obtained were as table 5.1 indicates.

id	**origin_cell_x**
40	163
origin_cell_y	**destination_cell_x**
113	160
destination_cell_y	**part_of_the_day**
115	0
transport	**starting_hour**
1	8
rush_hour	**trip_distance**
1	2689.22
trip_duration	**average_velocity**
65	41.37

Table 5.1: Trip sample.

5.1.3 Exploratory Data Analysis

Exploratory Data Analysis is known to be really useful on answering questions and generating assumptions for further analysis, with its main objective being to obtain knowledge about the data.

In this part, the data will be described, visualized and patterns will be detected so that interesting features and relationships between these will become more clear. At this phase, there are no a priori expectations about the relations between the features.

To obtain various summary statistics, data was firstly described.

	trip distance	**trip duration**	**average velocity**
mean	7566.8	42.1	406.3
std	11586.7	60.4	552.3
min	67.4	1	2.1
25%	1485.3	10	40.6
50%	3249.5	27	138.7
75%	9307.9	58	944.1
max	408591	1024	69810

Table 5.2: Numerical features description.

In table 5.2 we can visualize mean, standard deviation, minimum and maximum values and the quantiles of all the numerical features, since categorical ones cannot be summarized using these statistics. Note that the distance is in meters, the duration in minutes and the velocity in meters per minute. This step usually allows quality assessment, data understanding and detection of anomalies.

Features Distribution

When dealing with the data set, the first thing to do was to examine the univariate distributions of features.

Categorical

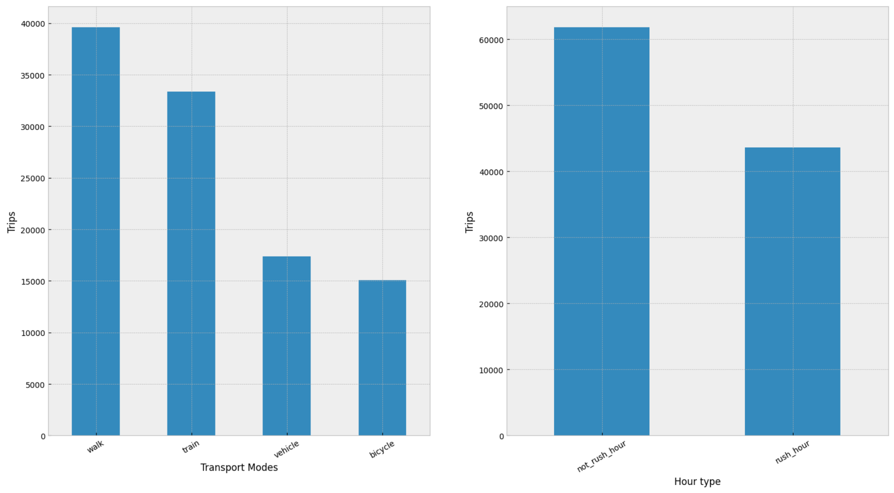


Figure 5.5: Bar plot for Rush Hour and Transport values distribution.

Bar plots can be used to show the number of observations for categorical features unique values.

As it is possible to see in figure 5.5, at rush hours there are fewer trips than at "normal" time - about 2/3.

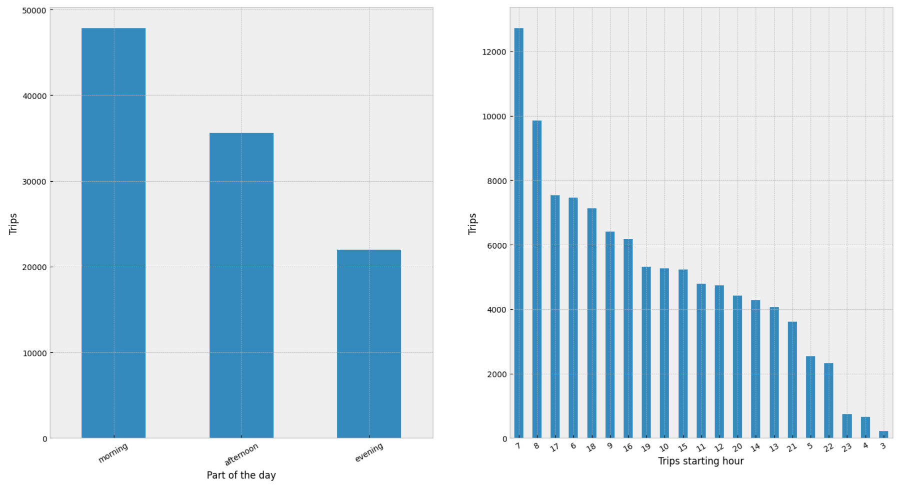


Figure 5.6: Bar plot for Starting Hour and Part of the day values distribution.

In terms of the use of transport, travel on foot along with train reign with similar percentage values. The use of bicycles comes next with vehicle, which also have similar values. In terms of percentages, we have 36 % for travel on foot, 29 % for travel by train, 16 % for travel by bicycle and 19 % for travel by vehicle.

When analyzing trips' starting hour in figure 5.6, it is possible to notice that it is in the morning rush hour that there are more trips. More precisely, the peak is located at 7 am, although there are also plenty of trips at 6, 8, 17 and 18 hours. The part of the day in which most trips are made is therefore the morning. Then it is in the afternoon and finally at night. In the morning, twice the amount of evening trips is made.

Numerical

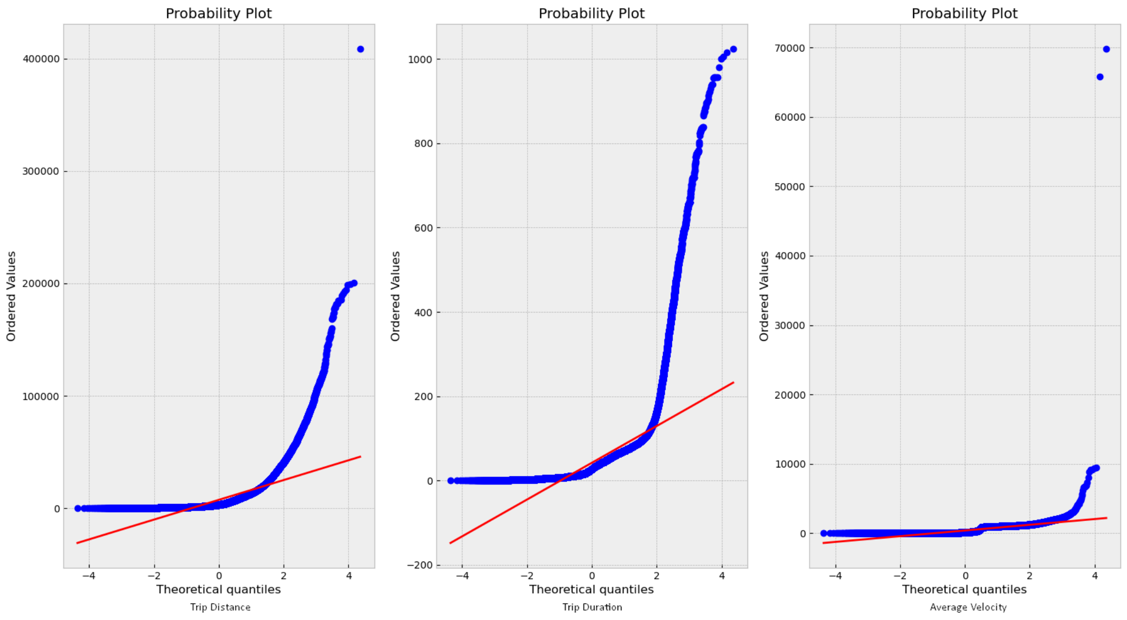


Figure 5.7: Probability plots.

By analyzing the distributions of numerical features with Normal Q-Q Plots, it was noticed that these numerical features do not follow a Gaussian distribution. A Q-Q plot is a scatterplot created by plotting two sets of quantiles against each other. If both sets of quantiles came from the same distribution, points forming a line that is roughly straight should be seen [59]. It was not necessary to resort to the Kolmogorov-Smirnov test and the use of hypotheses since the charts in 5.7 cleared all doubts.

Regarding the analysis of the distribution of numerical features, two different charts were created. A classic Histogram, which shows the number of observations for each value features can take, and a Histogram with a Kernel Density Estimation, which allows estimating the probability density function of a certain variable. Thus, a value of any given point in the feature space can be interpreted as a relative likelihood that the value of the random variable would equal that sample.

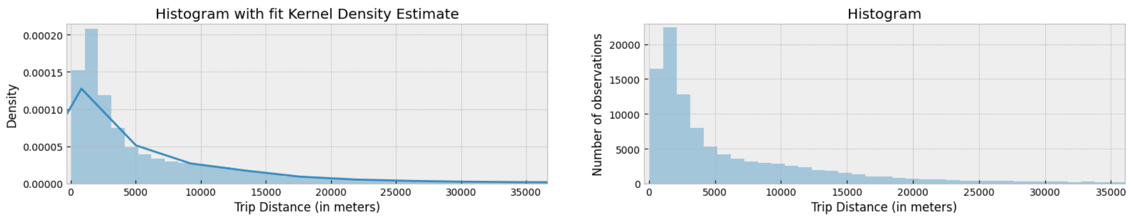


Figure 5.8: Histograms for Trip Distance with higher detail.

The histogram in figure 5.8 shows that most of the values of trip distance are located at the beginning of the distribution. In general, this data is spread and its distribution is skewed, instead of symmetric.

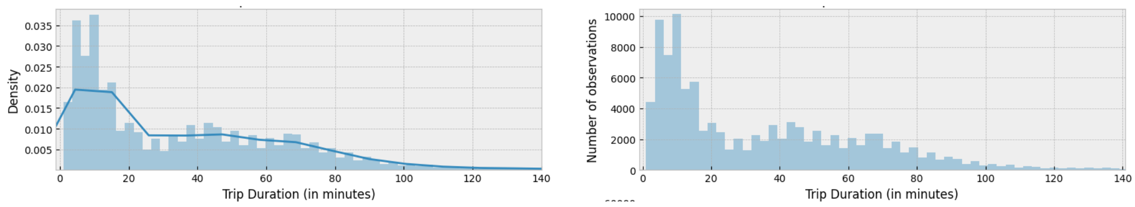


Figure 5.9: Histograms for Trip Duration with higher detail.

Similarly to trip distance, trip duration values in figure 5.9 are also located at the beginning of the distribution. It is not as accentuated as for trip distance, but in general, data here is also spread and its distribution asymmetric.

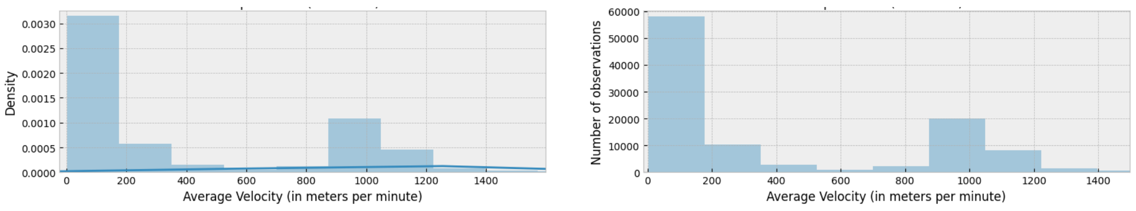


Figure 5.10: Histograms for Average Velocity with higher detail.

In histograms with a small amount of bins, it was only possible to notice that the average velocity values were between 0 and 1400 meters per minute (84 km/h). However, with a greater number of bins, in the 5.10 histogram, it is notorious that two groups represent the majority of average velocity values. One is between 0 and 300 meters per minute (18 km/h) and the other between 900 (54 km/h) and 1200 (72 km/h). This phenomenon may depend on the type of transport that is used, which will be analyzed later.

There is still a lot of predominance of outliers, so it is impossible to interpret the chart without using the zoom tool. It was then necessary to assess the need to remove outliers.

Outliers

The Data Science project begins with data collection and it is when outliers are first introduced to the population. However, it is not known about outliers in the collection phase. The outliers may result from an error during data collection or they may just be an indication of variance in the data.

Contrarily to categorical features, that do not possess outliers due to their nature, numerical had a bunch of outliers. Some outliers removal techniques were applied.

Box plots are a way of displaying the distribution of data based on a five number summary (minimum, first quartile (Q1), median, third quartile (Q3), and maximum). It is used to identify data distribution and detect outliers. The minimum line represents Q1 - 1.5 * Interquartile Range (IQR), while the maximum line represents Q1 + 1.5 * IQR.

In the first boxplot, there are quite extreme points that cause the values close to the median to be contracted. Therefore, the figure 5.11 is an already zoomed-in version of the boxplot for trip distance. Here we can see that there are outliers because there are points that fall up the maximum line.

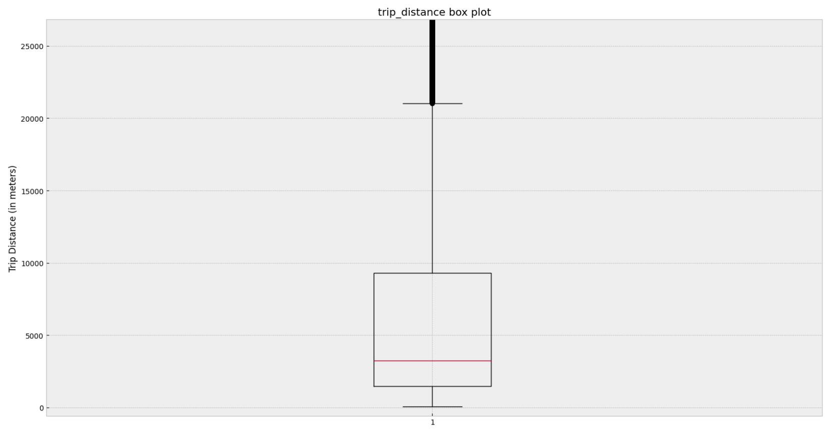


Figure 5.11: Zoomed-in Box Plot for Trip Distance (in meters).

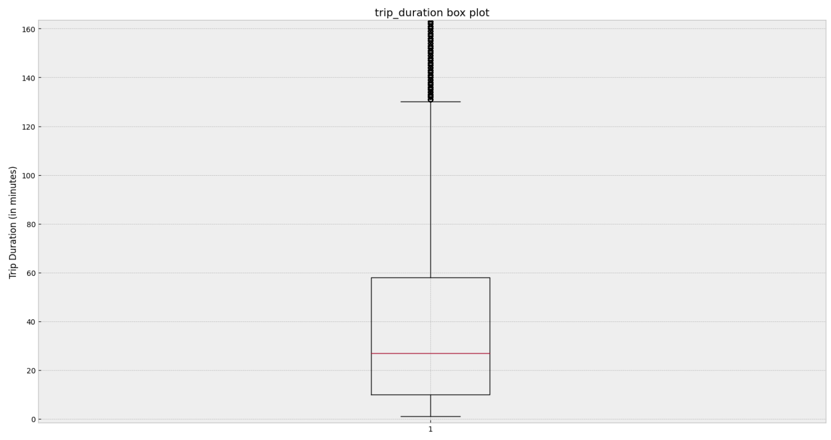


Figure 5.12: Zoomed-in Box Plot for Trip Duration (in minutes).

The same can be seen in figure 5.12 which is an already zoomed-in version of the boxplot for trip duration. Here we can see again that there are outliers because there are points that fall up the maximum line.

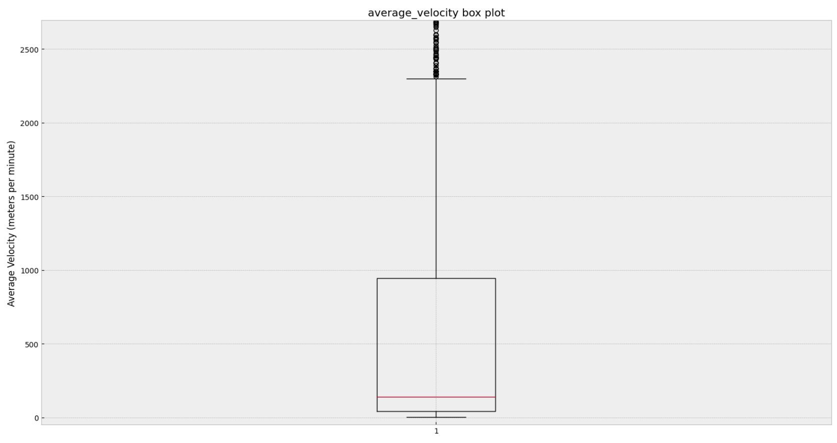


Figure 5.13: Zoomed-in Box Plot for Average Velocity (in meters per minute).

Finally, for average velocity, the same happens in the figure 5.13, which is an already zoomed-in version of the boxplot for trip duration. There are outliers because there are points that fall up the maximum line.

As Tukey's method [60] states, the IQR (Inter Quartile Range) is the distance between the lower (Q1) and upper (Q3) quartiles. Tukey [60] (1997) defined Q1 - (1.5 * IQR) and Q3 + (1.5 * IQR) as *inner fences*, Q1 - (3 * IQR) and Q3 + (3 * IQR) as *outer fences*, the observations between an inner fence and its nearby outer fence as "outside", and anything beyond outer fences as "far out".

As the data set of this work is skewed and does not follow a normal distribution, this method was chosen. It is applicable to skewed data since it makes no distributional assumptions and it does not depend on a mean or standard deviation [61].

So that, "far out" values, as Tukey defined, were removed from the data set.

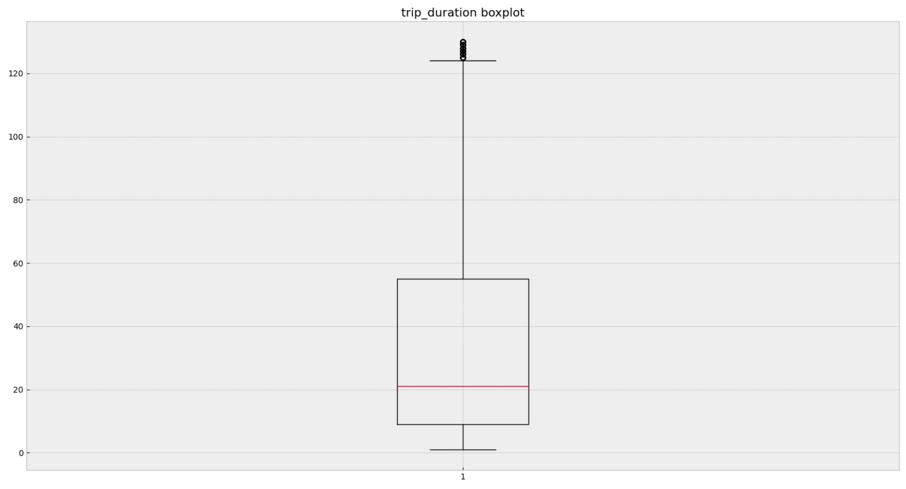


Figure 5.14: Box Plot for Trip Duration (in minutes) after Tukey's method.

The far out outliers have been removed, that is, those that pass Q3 + (3 * gls iqr). Therefore, there is a clear decrease in outliers, as seen in figure 5.14.

Features Correlation

The performance of Machine Learning algorithms is highly dependent on the data with which they are fed and, therefore, assessing the importance of features is of high priority. One way to check if there are relationships between the variables of the problem is through the correlation analysis in the data. Through correlation, it is possible to check if features depend on each other, if they are a cause for another feature and if they are associated.

Although there are only three numeric variables in the data set, for the supervised problem 5.2.2, these are possibly the most important.

Firstly, Pearson Correlation was evaluated among all features to measure the association of each pair of features in the data.

For that, it was necessary that the categorical variables of starting hour and part of the day were also dichotomous. As this test measures the association between continuous features [62], categorical ones were transformed into Dummy Variables, so that n binary features are created for the n values of each categorical feature. Moreover, every categorical feature in figure 5.15 is dichotomous, as they have 0/1 coding for the categories. When Pearson Correlation is used between a dichotomous feature and a continuous one, the correlation is also known as a Point-biserial Correlation Coefficient [63].

As Pearson correlation assumes that compared data is normally distributed and association

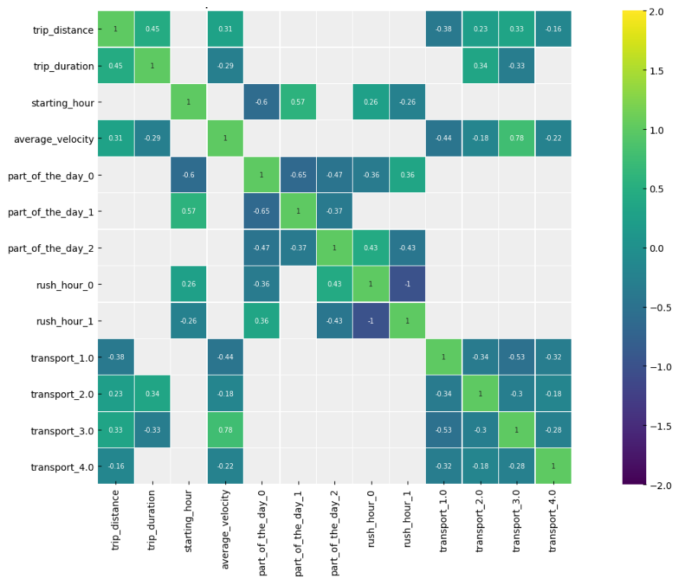


Figure 5.15: Pearson Correlation Coefficient.

levels between features were not so evident, Spearman correlation shall be used. It does not assume that data is from a specific distribution, as it is considered a non-parametric correlation measure. The Spearman rank-order correlation is a statistical procedure that is designed to measure the relationship between two variables on an ordinal scale of measurement [64].

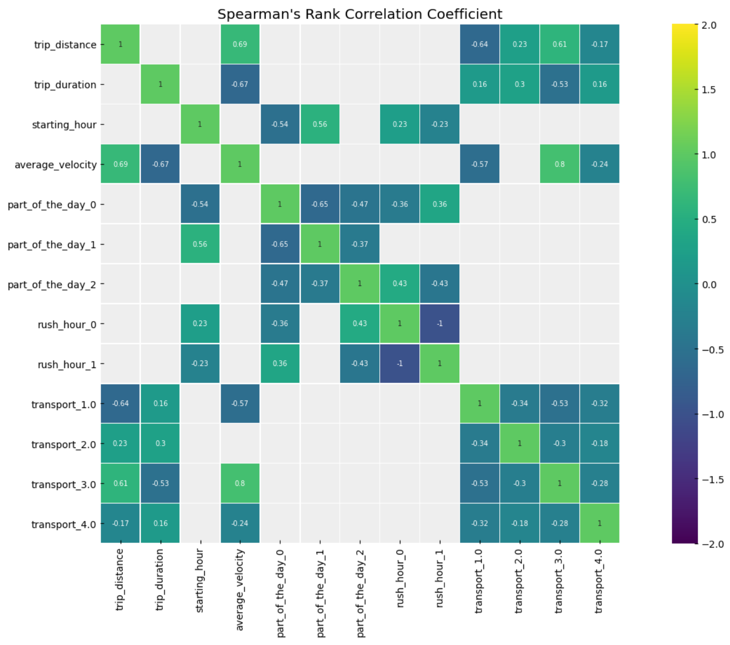


Figure 5.16: Spearman's Rank Correlation Coefficient.

First, as velocity comes from distance and duration, it is expected that there will be an increasing relationship with both in figure 5.16, that is, they are associated by a monotonic function.

As the Transport feature contains the labels for the transport mode classification problem, we also looked at the correlations with it, since they can help in that later stage. Therefore, figure 5.16 showed that the distance, duration and average speed of the trips are the most correlated with the transports modes. Analyzing each transport mode:

- Train (3.0) is particularly highly correlated both with distance, duration and average velocity.
- Walk (1.0) is not so correlated with the duration of the trip, but a lot with distance and average velocity.
- Bicycle (4.0) has tiny correlations with distance, duration and average velocity.
- Vehicle (2.0) only has correlations with distance and duration.

As has already been conveyed, these insights can be useful for the later phase of the project - that of classifying trips transport modes.

5.2 Machine Learning

This section is responsible for describing the development process of all used Machine Learning algorithms. Subsection 5.2.1 presents the approach for the grouping of trips, while subsection 5.2.2 presents the one for classifying trips transport modes.

5.2.1 Clustering of trips

The objectives of the internship required a set of large population movements to be found in order to present to user some metrics that make up the population movement. For this, clustering algorithms were used for the purpose. For the reader to understand, a simple example is a clustering of trips that takes into account the distance traveled on them. As the objective is to identify large population movements, clustering methods result in the division of trips into several clusters that have different traveled distances between them, but similar traveled distances within them. Thus, it is possible to isolate one of the clusters in which the distances covered are interesting for analysis so that we obtain more insights about it.

When clustering is performed on spatial objects, such as the location of people, we can say that Spatial Clustering is being applied. This was widely used to represent trips on the map. As there is a large amount of travel between the same origins and destinations, these were grouped by these two characteristics when represented on the map, which will be discussed later.

Algorithms

Since the data set had both categorical and numerical features, there were a variety of clustering algorithms that could be used here.

K-Means is considered the go-to algorithm when we are talking about a quick analysis on a data set that contains a very large number of samples [65]. Besides that, k-means generalizes to clusters of different shapes and sizes. In addition, since categorical features have been transformed into dichotomous, it allows the clustering of both numerical and categorical features, given that features with only 2 categories are not ordinal. This is because KMeans uses the Euclidean distance to measure the difference between the clusters' centroids. These centroids are averages of the objects that belong to a cluster.

As **Hierachical** Clustering algorithms are computationally expensive due to their space complexity, it was discouraged to use them for the large data set used in the project. Still on large data sets, the time complexity of a Hierarchical clustering gets really high [66]. Even so, it was used in small tests during development. A custom metric was even created to cluster mixed features, which produced interpretable results. This, for example, to cluster mixed features - such as origin and distance of trips.

DBScan allows the test of several distance metrics for large data sets but at the cost of having a high time complexity [67] [68]; although it does not assign a cluster to all objects, they are defined as noise and this can be good for outlier detection [67]. Even so, it was used since it is widely used in the state of the art of clustering. The two parameters - minimum points and epsilon - therefore had to be automatically calculated in order to find clusters that produce interpretable results.

K-Prototypes - basically a novel combination of k-means and k-modes - was used as a clustering of categorical and numerical features. K-Prototypes measures Euclidean distance between numerical features but measurse the distance between categorical features using the number of matching categories [69]. From the same authors, **K-Modes** was also used, which allows grouping data that contains only categorical features [70]. This one served the later problem of suggesting infrastructure locations.

Development

Clustering is often used to extract insights from data that are not clear to the naked eye. Imagine a store that wants to detect aspects that customers have in common. For this, the clustering can produce groups in which the customers have some characteristic in common that the store never imagined them to have. In the same way, a clustering method could group the trips in such a way that trips with similar patterns are in the same cluster.

In the case of trips, when passing all the features of the trips to the clustering algorithms, they will not produce interpretable results. There are too many dimensions for the algorithm to deal with. For this, combinations of features have been created that can possibly create insights logically expected.

Irrespectively of the clustering methods, these algorithms define clusters that are not known a priori, thus the final partition of data requires some kind of evaluation, in most applications (Rezaee et al., [71]). In this case, Internal Validation metrics were used. They are one of the ways to validate results produced by the grouping in which there is no target of expected labels to compare with the ones obtained. Charts were used to assess the interpretability of the clusters.

The hyperparameters of DBScan, epsilon and minimum points, are somewhat difficult to optimize. This process can be done by trial or through other techniques. By trial it could be quite long-lasting, so a technique that uses Nearest Neighbors was used. Since the minPts hyperparameter of DBScan is the minimum number of points that a cluster must have to be considered a cluster and epsilon is the distance by which a point must be from the center of a cluster to belong to it, Nearest Neighbors is used as a means to select the optimal eps for the selected minimum points [72].

In the case of K-Means and K-Prototypes, the number of desired clusters must be specified. Therefore, trial and error has been made until clustering results are found to be good or be meaningful.

5.2.2 Transport Mode Classification

Furthermore, a multiclass problem in which we want to determine which transportation modes were most likely used (train, vehicle, walk and bicycle) in trips was carried out. For that, a supervised learning problem had to be solved. These results are then fed to the web application so that it can label population movements as to their estimated transport percentages.

Algorithms

The algorithms chosen for Supervised Learning are widely used in the general State of the Art of Machine Learning:

- Decision Tree
- Random Forest
- Light GBM

In general, **Decision Tree** is known for presenting average results in a short time, which proved to be useful for specific tests during development. On the other hand, **Gradient Boosting** and **Random Forest** are more promising algorithms in the area and are known to obtain better results than Decision Trees.

In an article of models performance comparison that also explored spatial data, the Random Forest (RF) model outperformed all other methods in predictive accuracy; methods like Boosted Regression Trees, k-Nearest Neighbor and Support Vector Machine [73]. More specific to the problem of modeling travel mode choice, the RF model also outperformed every other model in a comparative study of Machine Learning (ML) classifiers [74]. RF against algorithms like Artificial Neural Network (ANN), Naive Bayes and Support Vector Machines obtained median accuracy results of 91.4%.

In another paper, a comparison of algorithms for multiclass classification is made, in which a Bayesian Regularization Artificial Neural Network obtained a prediction accuracy that was better than K-Nearest Neighbor (KNN) and SVM. It was from there that using a ANN as well was also thought of. In fact, it was used for systematic tests throughout the process and presented results similar to Random Forest, but with much more training time. In the final tests, this was not used. The number of layers and neurons needed was too large for the number of samples the trips data set contained.

Gradient Boosting Machines (GBM) were being used as what was expected to be the best ML algorithm, given its known good results. Until Light Gradient Boosting Machines (LGBM) was found. Guolin Ke et al. 2017 [17] claim that LGBM is 20 times faster than conventional Gradient Boosting Decision Tree at training, while achieving almost the same accuracy. In the paper, they propose the LGBM, which implements two novel techniques: Gradient-based One-Side Sampling and Exclusive Feature Bundling to deal with large number of data instances and large number of features, respectively. With the help of GOSS and EFB, LGBM can significantly outperform XGBoost and SGB both in terms of computational speed and memory consumption Since the hardware used for testing was not top notch, the advantages that LGBM promised to bring were adequate and therefore it was used.

Development

Given that the goal of this project phase was to predict transport modes, it was important to visualize numerical features related to transport mode. For that, boxplots were used.

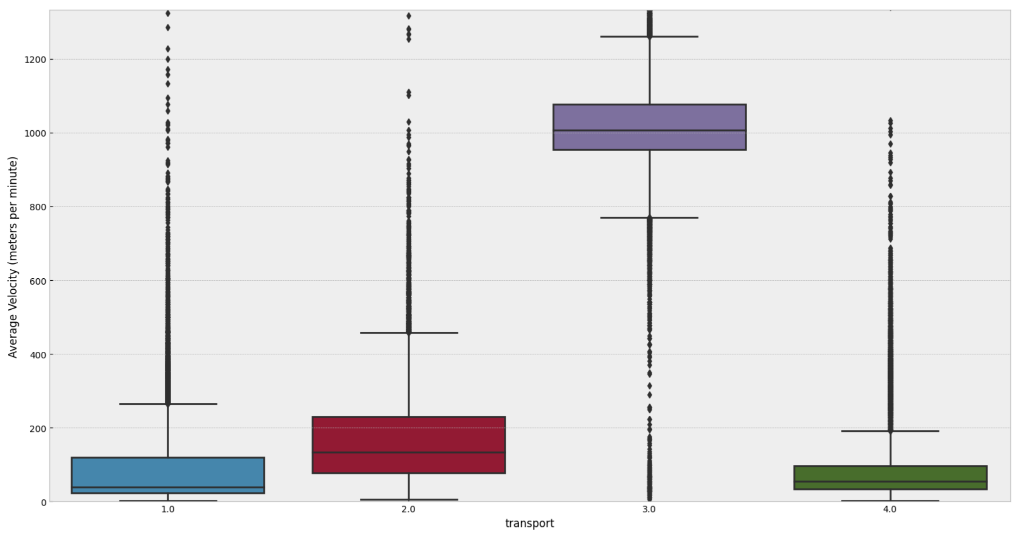


Figure 5.17: Average velocity (meters per minute) for each transport mode in a box plot.

As shown in figure 5.17, the values for the average velocity (in meters per minute) of each type of transport are represented. The conclusions that can be drawn from this are somewhat to be expected. The average velocity of Train trips differs considerably from that of other transport modes. For Vehicle velocity it also happens but not so evidently, despite its median being very close to that of Walk and Bicycle. These two modes of transport have quite coincident average velocity.

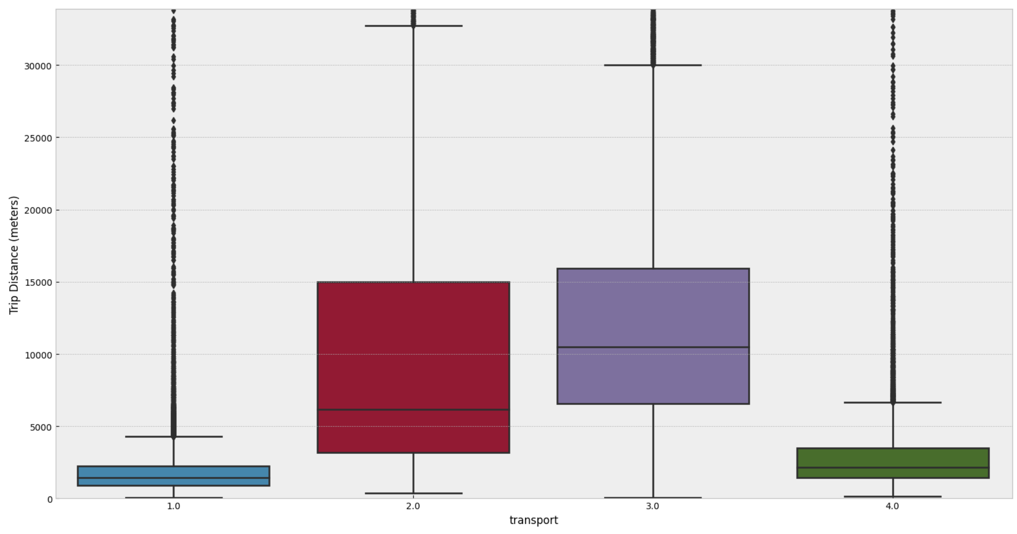


Figure 5.18: Trip distance (in meters) for each transport mode in a box plot.

Figure 5.18 presents the values for the distance covered for each mode of transport, in meters. In this case, the coincidence between Walk and Bicycle is no longer as much as at average velocity, but they still have a part where that happens. In relation to Train and Vehicle, they have very similar distance values, despite having far apart medians.

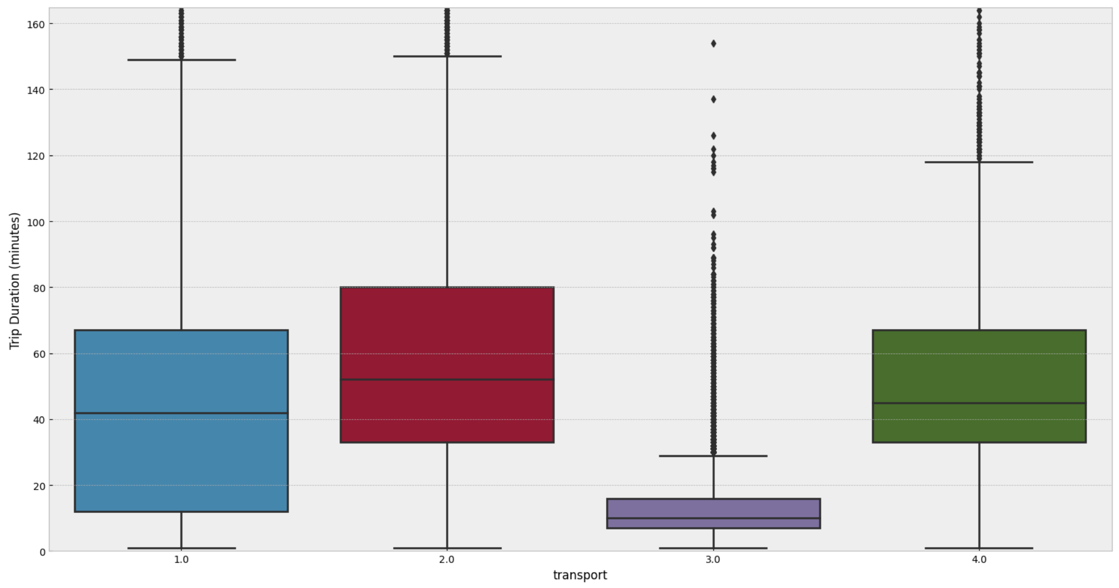


Figure 5.19: Trip duration (in minutes) for each transport mode in a box plot.

In figure 5.19 it is possible to observe another comparison, this time between the duration of trips (minutes) made by the various transport modes. Durations for both Walk, Bicycle and Vehicle have considerably coincident values and very similar medians. Despite this, the Q1 values for Walk are lower (around 15 minutes), while for Vehicle and Bicycle they start around 40 minutes. The values in Q3 for Walk and Bicycle are close to 70 minutes, while Vehicle durations at Q3 are 80 minutes.

From these box plots, it is possible to notice that the Train mode is the one that is most distinguishable in terms of numerical features. For this very reason, it is expected that the classification algorithms achieve better performance when classifying that label. In terms of Vehicle, it is noted that it is coincident both with Train, in distance, as with Walk and Bicycle in the duration of the trips. This may be due to the fact that this label represents several types of vehicles. That said, it may not be easy to classify. As for Walk and Bicycle, as expected, they always have very similar values, which can be detrimental to their classification results.

Then, the iterative process of Machine Learning began. The trip data set was not obtained at once. It was an iterative process with several experiments and adjustments during development. As Trip Detection 5.1.2 was undergoing, changes and new features to be extracted were added, the Machine Learning phase was already underway. The first features obtained were the latitudes and longitudes of origin and destination for each trip. From this it was deduced that it was not valuable to create models to classify transport modes with only these location features, since the algorithm would learn in relation to specific locations and would not be adptable to other cities. The conversion of these locations to a grid was then performed, which is explained in 5.1.2, and features derived from the timestamp were created - part of the day, starting hour and rush hour. Thus, the first tests were performed, which are shown in 5.3 and are obtained with 200 thousand trips. No scaling, outlier removal or dummy variables were used, and the fastest algorithm, Decision Tree, was used, which is a Tree-based algorithm - like random forests - it is not affected by feature scaling [75]. Though, other algorithms that exploit distances between data samples, such as KNN and SVM, are sensitive to feature transformations.

	Metric	Mean
5-fold Cross-Validation	Accuracy (%)	49.2 ± 4.4
	Precision (%)	45.9 ± 2.8
	Recall (%)	44.6 ± 3.3
	F-measure (%)	44.7 ± 3.1

Table 5.3: Decision Tree 5-fold Cross-validation results using only timestamp features.

Then, as explained in the Trip Detection chapter 5.1.2, the distance actually covered was extracted, which produced the results of the 5.4 table.

	Metric	Mean
5-fold Cross-Validation	Accuracy (%)	51.1 ± 4.7
	Precision (%)	47.7 ± 3.5
	Recall (%)	46.7 ± 3.8
	F-measure (%)	46.7 ± 3.7

Table 5.4: Decision Tree 5-fold Cross-validation results using only timestamp features and trip distance.

In addition to trip distance, the duration started to be used later. The results are shown in the table 5.5

	Metric	Mean
5-fold Cross-Validation	Accuracy (%)	77.9 ± 2.6
	Precision (%)	75.0 ± 2.8
	Recall (%)	74.9 ± 2.2
	F-measure (%)	74.8 ± 2.5

Table 5.5: Decision Tree 5-fold Cross-validation results using timestamp features, trip distance and duration.

With distance and duration of the trip, the average velocity started to be calculated. The figure 5.6 shows the results.

	Metric	Mean
5-fold Cross-Validation	Accuracy (%)	78.7 ± 2.6
	Precision (%)	75.9 ± 2.8
	Recall (%)	75.8 ± 2.2
	F-measure (%)	75.7 ± 2.5

Table 5.6: Decision Tree 5-fold Cross-validation results using all the features.

Removal of outliers via Tukey's Method was also applied to assess whether it affected the results. The removed outliers were what are called far out values. Since it did not produce major differences, the technique was not used in the following tests. What can be seen in 5.7.

Data scaling was then applied: in this case Min Max Scaling and Standardization. The results got worse - the macro F-measure went to 75.6 % in both cases, which is worse than without scaling.

Dummy Variables applied to categorical variables produced similar results 5.8. For example, the part of the day feature that has 3 categories (morning, afternoon and night) is

	Metric	Mean
5-fold Cross-Validation	Accuracy (%)	78.1 ± 2.6
	Precision (%)	75.1 ± 2.7
	Recall (%)	75.0 ± 1.9
	F-measure (%)	74.9 ± 2.3

Table 5.7: Decision Tree 5-fold Cross-validation results using all the features and with Tukey's method for outliers removal applied.

transformed into 3 binary columns.

	Metric	Mean
5-fold Cross-Validation	Accuracy (%)	78.7 ± 2.4
	Precision (%)	75.8 ± 2.6
	Recall (%)	75.7 ± 2.1
	F-measure (%)	75.6 ± 2.4

Table 5.8: Decision Tree 5-fold Cross-validation results using all the features and with categorical variables transformed into Dummy.

Clustering can also be applied to help classify the transport mode; more specifically by grouping trips by distance, duration and average speed. For this, a previous clustering is applied with k-means and k = 4 to provide a new feature corresponding to the cluster to which a given trip belongs. With this new feature together with all the others, the results were those of the 5.9 table.

	Metric	Mean
5-fold Cross-Validation	Accuracy (%)	78.8 ± 2.5
	Precision (%)	75.9 ± 2.7
	Recall (%)	75.9 ± 2.1
	F-measure (%)	75.7 ± 2.4

Table 5.9: Decision Tree 5-fold Cross-validation results using the clustering feature as well as all the other features.

With this feature produced by the clustering, it is possible to notice that the results have improved even if only minimally.

The emphasis placed on precision, recall and f-measure was due to the fact that this is a multiclass problem and accuracy is not well suited for class imbalanced problems; a good classifier only for classes in abundance, will still obtain high classification accuracy. For such problems of imbalanced classes, measures that give equal attention to all classes are then required [76]. By counting the total true positives, false negatives and false positives, the micro F-measure does not place enough emphasis on rare classes, so for class imbalanced settings it is not very useful. Therefore, macro is preferred over micro since the former gives equal importance to each class, whereas the later gives equal importance to each sample [76] - which means that the more the number of samples, more importance is attributed to it in the final score thus favoring majority classes, just like accuracy does.

Summarizing, the most popular performance metric for class imbalanced problems and the one that was used in the final tests was then the F-measure macro, which instead uses an average of the precision and recall of each label in the F-measure [76] formula. Even so, the F-measure must be used carefully as a metric to distinguish performance from algorithms.

Successfully classifying one of the classes can be more crucial than classifying another, and that is where the importance given to precision and recall comes into the F-measure [77]. For example, it would be worse to classify a sick person as healthy than to classify a healthy person as sick.

5.3 Suggestion of locations for new transport infrastructures

Before the development phase, one of the project objectives was to be able to interact with a map in order to remove/add infrastructure from the transport network and subsequently assess the consequences of changes in the transport network. Imagining that a bus stop was being repositioned; would the repositioning result in a better response to the inhabitants' demand? This is an example of a question to which an answer was sought.

5.3.1 Research

An investigation was then carried out. It was seen that Machine Learning is not one of the most used domains in the literature methods to solve problems of this nature, but optimization and graph theory are. A review of studies on the topic was made although only a few seemed to be possible to apply to this case, given that there are few studies that relate a transport network to the mobility of the people who attend it.

One included developing a supernode graph for modeling the bus transport networks. Shanmukhappa et al. (2018) [78] proposed weights for the supernodes that represented a static demand estimation. This estimation considered the density of points of interests (POIs) and the node occupying index (number of people accessing a particular bus stop) over localized zones. The edge weights represent the number of routes to operate between two supernodes. Instead of the shortest distance measure used in previous works, end-to-end delay was proposed as a parameter to measure the topological efficiency of bus networks.

Additionally, one interesting approach using two network graphs for bus transportation networks was proposed by Caminha et al. (2018) [79]. The first network graph would be for the service that the transport network gives people, where the graph nodes are the bus stops and the edges between the nodes are the number of vehicles offering routes between those stops. The second corresponded to the mobility of people between stops. The nodes would represent the bus stops and the edges the demand of bus users between two consecutive bus stops. The edge is defined as a weight that sums up the total number of users that passed through it. They used metrics to assess bottlenecks and lack of service at certain points in the transport network. In this way, problems in the network could signal places where the addition/removal of infrastructure could make sense. Also, community detection algorithms are applied to find "bus lines that are system bottlenecks (crowded bus lines) and lines where there is a waste of resources (often empty buses)". Then they would compare weight distributions of supply and demand complex networks to find such imbalances.

In several studies, many graph theory metrics were used to compare transport networks before and after changes. Metrics like degree of a node, characteristic path length, global efficiency, giant component, betweenness centrality, among others, are intended to evaluate a transport network, but without taking into account the mobility of people in that city. Some of these studies were [80], [81], [82], [83], [84], [85], [86].

In the end, we opted for a simple approach that does not follow any of these previously presented studies. The approach that was considered the most interesting was the one that used two network graphs for bus transportation networks, proposed by Caminha et al. [79] given that it allowed establishing a relationship between the topology of a transport network and the mobility of people who attend it. In this way, it would be possible to relate various configurations of bus stop locations with the mobility of the city. Even so, going for network graphs or multi objective optimization was out of question for the internship, given that the intern has no experience in this field, the time already spent on research had already been significant and therefore implementing this would consume more time than the available. In addition, the data needed to create a transport network graph in Tokyo was not enough given that there was no *bus* transport mode in the Open PFLOW data set and the translation of a mobility data set into a network graph that has users' demand was not explicit in the paper.

5.3.2 Development

Initially, one of the requirements was the removal/addition of transport network infrastructure and the consequent assessment of the effects caused by these changes in the transport network. With the data set obtained, this was not possible to do, as previously explained. That said, the suggestion of locations for new transport infrastructures arose, namely bus stops. This suggestion is made based on the most frequent daily movement of people in Tokyo, being the idea based on the assumption that a long distance and frequent walking trip may be an indicator that there is a lack of public transport supply in that area. The decision to make this suggestion was also inspired by a study by Demissie et al. [33] in which travel patterns of people are used to suggest transit routes, that is, linkages between users' origin and destination in highly needed areas are the only information used in the recommendations, as no other important metrics such as shorter travel time or higher ridership are used.

The developed process mainly used Data Mining to discover the trips between cells that are made on foot and that represent great distances covered. For this, trips from the data set that had more than a certain distance were selected. For this distance, 1.5 kilometers was thought to be a good minimum value for trips that may require transportation. Rush hour filtering can optionally be done, since in rush hours, walking trips that most likely require additional transportation can be found. Then, a clustering of cells of origin and destination is applied, through K-Modes, so that equal trips are counted. Clustering by categorical variables was used even though the *group-by* Pandas' function would also work. Hence, k equal trips that are in greater number are selected and they are replaced with new average velocities and trip durations. The average velocity at which buses travel was used to replace the average velocity of these trips that were previously made on foot. With this new velocity and trip distance, the new trip duration was calculated.

In the end, some statistics from the initial data set are compared with those from the data set with the aforementioned changes:

The following statistics refer to the suggestion of 6 bus stops when analyzing the trips that agents take on foot during rush hours.

Table 5.10 shows metrics such as affected people, affected trips and time saved in minutes, which are some of the results of suggesting locations for bus stops. Nine people/trips used the suggested locations as an origin or destination for long walking trips. With the implementation of bus stops at these sites, 197 minutes can be saved.

Affected Trips	**People Affected**	**Time Saved (minutes)**
9	9	197

Table 5.10: Table representing some metrics after the suggestion is applied to the Open PFLOW data set used. The metrics are affected people, affected trips and time saved in minutes.

	Walk	**Vehicle**	**Train**	**Bicycle**	**Velocity**	**Dur. (m)**	**Dist. (m)**
Before	37.5%	12.2%	34.5%	15.9%	24.9	41.4	8192
After	36.6%	13.1%	34.5%	15.9%	25.4	41.2	8192

Table 5.11: Table representing proportion for each transport mode and average of metrics such as distance, duration and velocity.

Table 5.11 shows the proportion for each transport mode, in addition to the average of statistics such as distance, duration and velocity. These metrics are presented for both before and after the suggested locations are changed in the Open PFLOW data set used. As expected, the percentage of trips by foot decreased, while vehicle trips increased, since bus trips are included in the Vehicle category. As a consequence, the overall speed of travel increases and the duration decreases, while distance remains the same.

5.4 Web Application

This section has as main objective the description of the entire development of the web application. It will describe how the functionalities for uploading, querying, deleting and editing files were developed. It will also describe the entire process that allows a user to visualize the mobility of the inhabitants of Tokyo, to group their trips according to the characteristics of the trip itself, to classify the transport modes that they use in their trips and to obtain suggestions of new locations for bus stops. In order to enable the objectives initially proposed to be met, the platform was developed based on the following types of data:

- Locations of people throughout the day
- Tokyo Inhabitants trips
- Bus Stops
- Train Stations
- Taxi Lines
- Ferry Stations
- Airports

5.4.1 Upload

A system-authenticated user who does not have AWS credentials may want to upload files. For that purpose, a secure way is by using a presigned URL mechanism to perform the upload.

Presigned Post

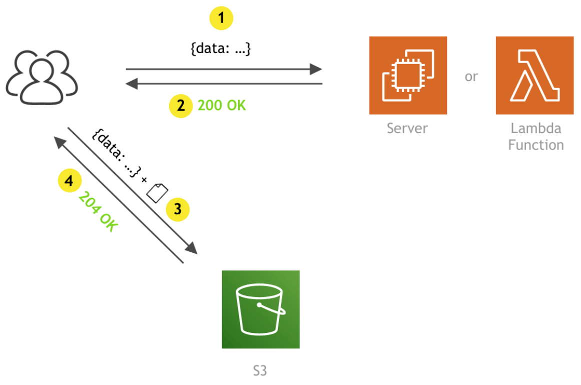


Figure 5.20: Generating a Presigned URL to Upload a File [9].

When a user enters the upload page of the desired data set, the front end immediately generates a presigned URL POST request to upload a file. This request corresponds to action 1 of the figure 5.20. The back-end Lambda Function, which deals with the request, responds (action 2 of figure 5.20) with both an upload URL and additional fields that must be saved and passed as part of the subsequent HTTP POST request.

The way used to merge all fields and the file into one object was through a FormData. A FormData is a set of key/value pairs representing form fields and their values. This object is then sent (action 3 of figure 5.20) using HTTP Multipart requests - POST body encoding used to send files and data over HTTP. This way, the request body is formatted as a series of "parts".

When the user selects the desired file to upload and hits the submit button, a POST to the upload url is issued, along with the FormData object (containing the file and fields).

Simple Storage Service (S3) receives the file and stores it. Next, since the Gateway API only allows calls of up to 30 seconds, it was necessary to create a trigger that would cause the data set POST in S3 to launch a lambda function to put this data set into DynamoDB. Thus, S3 can be uploaded to DynamoDB within 15 minutes, as the function is not invoked by the Gateway API but by an event in the S3 bucket.

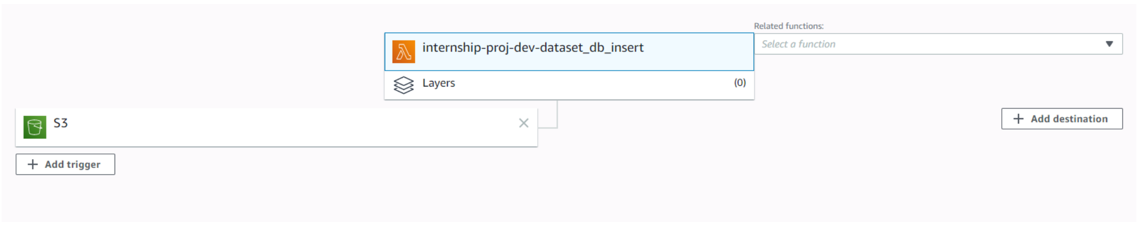
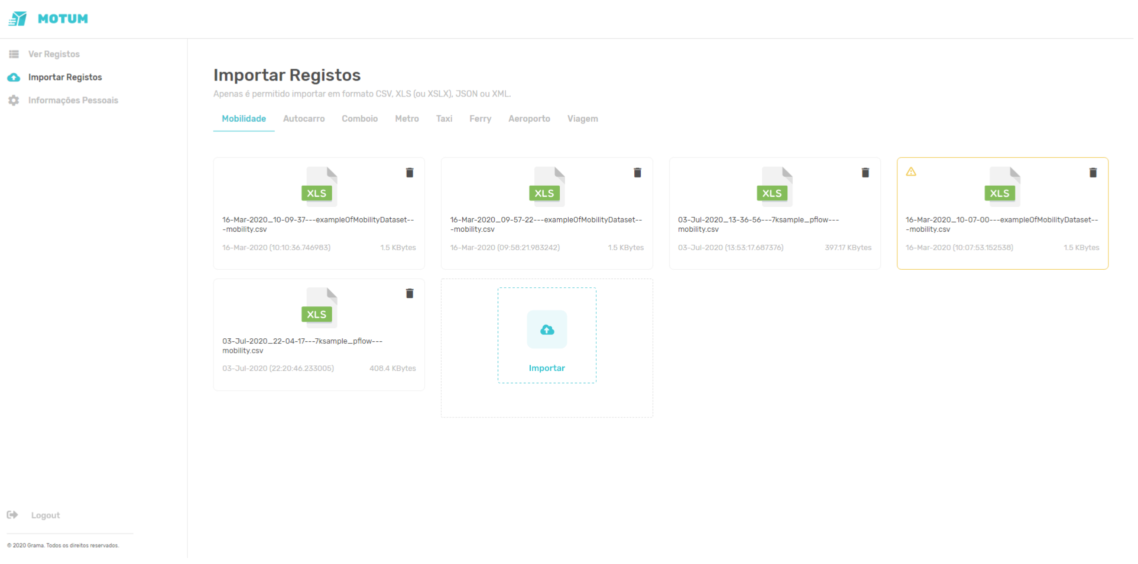


Figure 5.21: S3 triggers a Lambda function to insert a data set into DynamoDB.

The figure 5.21 shows that a trigger is invoked when a data set comes within a POST to S3. When the file upload is complete, the S3 Bucket automatically triggers the execution of a Lambda Function that initially had the function of handling and storing all the data loaded in its respective tables. However, due to the limited time of execution of 15 minutes, this approach had to be changed, since this whole process could take more than 15 minutes depending on the amount of data. In order to solve this problem, Lambda Function now triggers the start of a job on AWS Fargate responsible for handling and storing the data. AWS Fargate is an AWS technology that provides on-demand computing capacity for containers. This technology removes the need for server configuration and only charges for the time of use, which for this project is a great advantage, because the frequency of data loads is low. Therefore, a Docker container was developed. It realizes the type of data being loaded, and depending on that, does all the proper treatment and storage in the database.

In order to apply version control to the uploaded data, a metadata table was also created where information about the uploaded file is stored, such as the name, upload date, type and size. Especially for people location files, there are 2 columns referring to the oldest and most recent date present in the uploaded data set. The loading status - successful or conflictuous - and the number of conflicts in the loading are also stored.

There are two types of conflict checking. For data relating to the location of people throughout the day, having conflicts means that some date of the data being loaded is contained in the time intervals of the data already existing in the database. Thus, a data record with a date such as 7/2/2020 cannot be inserted in the database if there's already data from 2/1/2020 to 2/9/2020, for example. Regarding other types of files, conflicts occur when trying to insert a record that is already in the database. For this goal, the identifier of each record in the database is a hash of the junction of its features' values.



Importar Registos

Apenas é permitido importar em formato CSV, XLS (ou XSLX), JSON ou XML.

16-Mar-2020_10-09-37---exampleOfMobilityDataset---mobility.csv
16-Mar-2020 (10:10:36.746983) 1.5 KBytes

16-Mar-2020_09-57-22---exampleOfMobilityDataset---mobility.csv
16-Mar-2020 (09:58:21.983242) 1.5 KBytes

03-Jul-2020_13-36-56---7ksample_pflow---mobility.csv
03-Jul-2020 (13:53:17.687376) 397.17 KBytes

16-Mar-2020_10-07-00---exampleOfMobilityDataset---mobility.csv
16-Mar-2020 (10:07:53.152538) 1.5 KBytes

03-Jul-2020_22-04-17---7ksample_pflow---mobility.csv
03-Jul-2020 (22:20:46.233005) 408.4 KBytes

© 2020 Grama. Todos os direitos reservados.

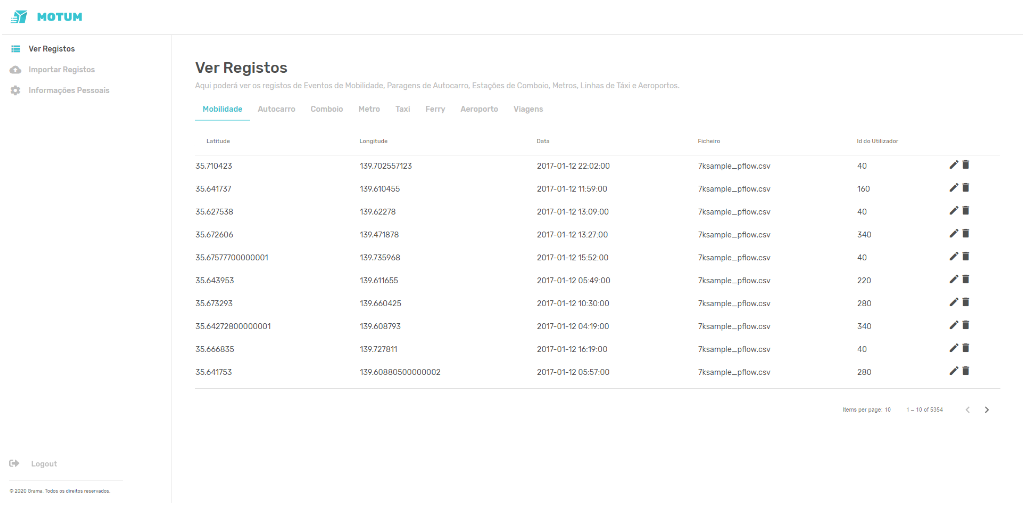
Figure 5.22: Page that allows users the upload of historical data for the first time.

Figure 5.22 shows the page that makes it possible to upload a file with csv format for the first time. In this example, it is possible to load a mobility file that contains records of people locations throughout the day.

Other types of data files can also be loaded through similar pages that can be accessed by selecting the desired type in the top bar. When a file has conflicts amidst its insertion, this is recorded in a status column. In the web application, it is shown as in figure 5.22 - a file marked by the yellow color and a warning sign.

5.4.2 Visualization

In order to facilitate the visualization of the data stored in the database, pages with tables were created to list them.



Ver Registos

Aqui poderá ver os registos de Eventos de Mobilidade, Paragens de Autocarro, Estações de Comboio, Metros, Linhas de Táxi e Aeroportos.

Latitude	Longitude	Data	Ficheiro	Id do Utilizador
35.710423	139.702557123	2017-01-12 22:02:00	7ksample_pflow.csv	40
35.641737	139.610455	2017-01-12 11:59:00	7ksample_pflow.csv	160
35.627538	139.62278	2017-01-12 13:09:00	7ksample_pflow.csv	40
35.672606	139.471878	2017-01-12 13:27:00	7ksample_pflow.csv	340
35.67577700000001	139.735968	2017-01-12 15:52:00	7ksample_pflow.csv	40
35.643953	139.611655	2017-01-12 05:49:00	7ksample_pflow.csv	220
35.673293	139.660425	2017-01-12 10:30:00	7ksample_pflow.csv	280
35.64272800000001	139.608793	2017-01-12 04:19:00	7ksample_pflow.csv	340
35.666835	139.727811	2017-01-12 16:19:00	7ksample_pflow.csv	40
35.641753	139.60880500000002	2017-01-12 05:57:00	7ksample_pflow.csv	280

Items per page: 10 1 – 10 of 5354

© 2020 Grama. Todos os direitos reservados.

Figure 5.23: Example of a page that lists a type of historical data - in this case locations of people over time.

The figure 5.23 shows the data listing page on the web app. For different types of data, the structure of the page does not change, but the content does. In this example, you can view the location of people over time data in tabular form.

Bidirectional Pagination

One of the encountered problems was the pagination of data sets on the front end. Amazon Web Services (AWS) does not offer an optimized and practical way to implement bidirectional pagination for a large amount of data.

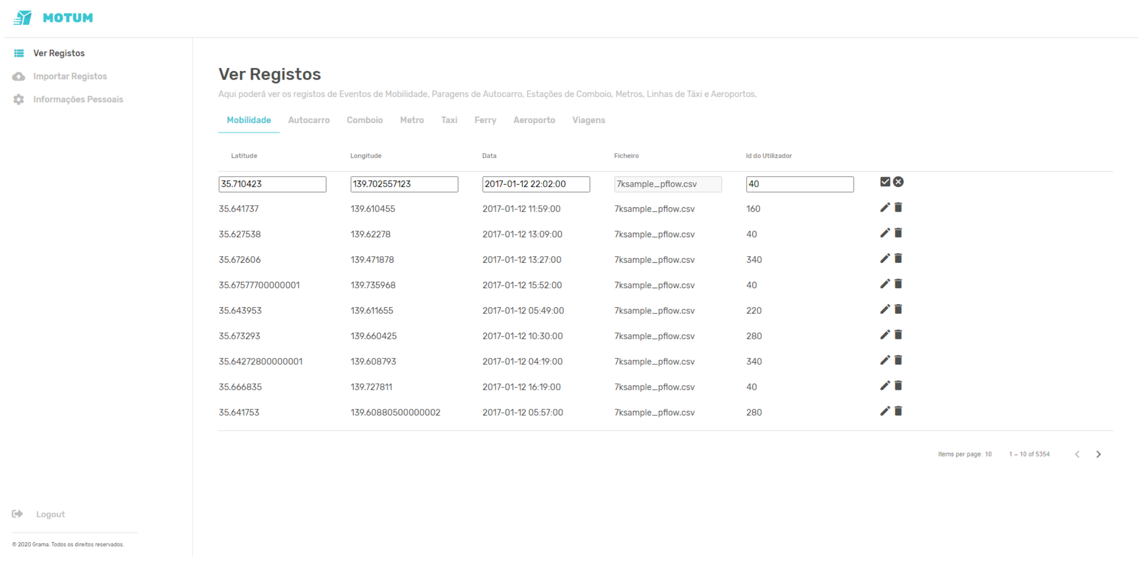
After establishing the number of results that the API returns at one time (10), it was found that an API response returned the records together with a LastEvaluatedKey field that corresponds to the last record returned. This field can then be sent as ExclusiveStartKey in a subsequent GET request to the database to return the next 10 results.

That said, the solution developed has a HashTable to store each of these LastEvaluatedKeys already visited, so that when the user needs to return there, this query is made in a more optimized way.

If a certain index does not yet exist in the HashTable, the database will be queried for all records until the desired first one, and then all the initial records will be cut until the last desired ten.

5.4.3 Editing

In order for the user to edit the data stored in the database, an icon was added to each record in the listing tables. By clicking on this icon, the attributes that can be edited become editable, thus allowing the user to modify them and subsequently save them in the database.



Ver Registos

Aqui poderá ver os registos de Eventos de Mobilidade, Paragens de Autocarro, Estações de Comboio, Metros, Linhas de Táxi e Aeroportos.

Mobilidade | Autocarro | Comboio | Metro | Taxi | Ferry | Aeroporto | Viagens

Latitude	Longitude	Data	Ficheiro	Id do Utilizador
35.710423	139.702557123	2017-01-12 22:02:00	7ksample_pflow.csv	40
35.641737	139.610455	2017-01-12 11:59:00	7ksample_pflow.csv	160
35.627538	139.62278	2017-01-12 13:09:00	7ksample_pflow.csv	40
35.672606	139.471878	2017-01-12 13:27:00	7ksample_pflow.csv	340
35.67577700000001	139.735968	2017-01-12 15:52:00	7ksample_pflow.csv	40
35.643953	139.611655	2017-01-12 05:49:00	7ksample_pflow.csv	220
35.673293	139.660425	2017-01-12 10:30:00	7ksample_pflow.csv	280
35.64272800000001	139.608793	2017-01-12 04:19:00	7ksample_pflow.csv	340
35.666835	139.727811	2017-01-12 16:19:00	7ksample_pflow.csv	40
35.641753	139.60880500000002	2017-01-12 05:57:00	7ksample_pflow.csv	280

Items per page: 10 1 – 10 of 5354

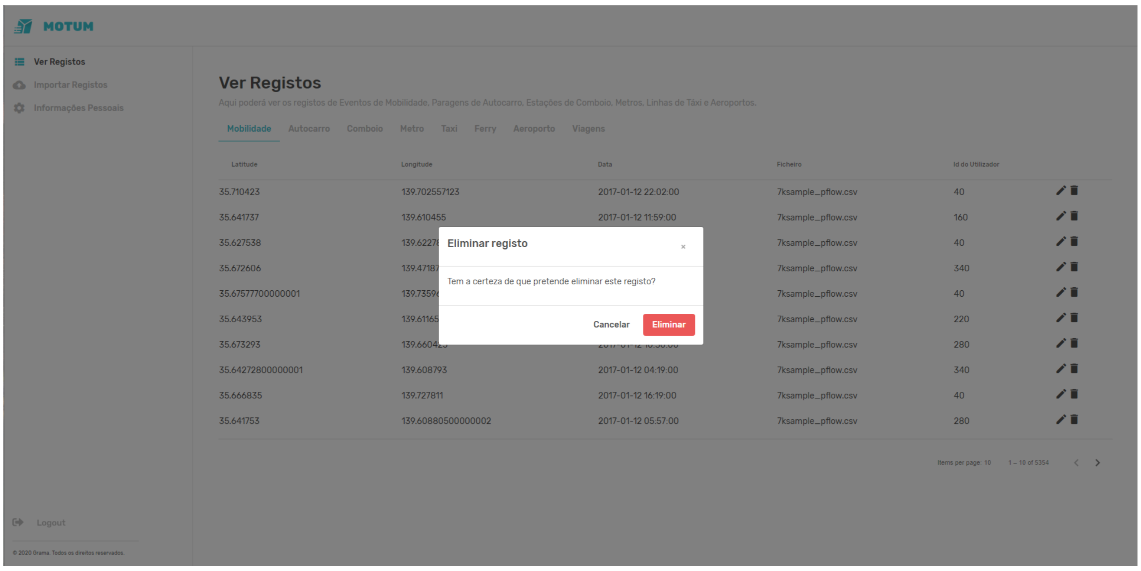
© 2020 Grama. Todos os direitos reservados.

Figure 5.24: Page that allows data to be edited line by line.

In figure 5.24, it is possible to see the list page of mobility where a line is editable. You can also see that only a few fields can be edited. The file name from which this record originated cannot be edited. This functionality is present in the visualization pages of all types of data.

5.4.4 Deletion

Another feature that facilitates the user managing the data stored in the database is the individual removal of records from the tables. For this and in a similar way to the process of editing a certain element of the table, the user needs to click on the icon that is present at the end of each line of the table.



Ver Registos

Aqui poderá ver os registos de Eventos de Mobilidade, Paragens de Autocarro, Estações de Comboio, Metros, Linhas de Táxi e Aeroportos.

Mobilidade | Autocarro | Comboio | Metro | Taxi | Ferry | Aeroporto | Viagens

Latitude	Longitude	Data	Ficheiro	Id do Utilizador
35.710423	139.702557123	2017-01-12 22:02:00	7ksample_pflow.csv	40
35.641737	139.610455	2017-01-12 11:59:00	7ksample_pflow.csv	160
35.627538	139.6227		7ksample_pflow.csv	40
35.672606	139.4718		7ksample_pflow.csv	340
35.67577700000001	139.7359		7ksample_pflow.csv	40
35.643953	139.6116		7ksample_pflow.csv	220
35.673293	139.6604		7ksample_pflow.csv	280
35.64272800000001	139.608793	2017-01-12 04:19:00	7ksample_pflow.csv	340
35.666835	139.727811	2017-01-12 16:19:00	7ksample_pflow.csv	40
35.641753	139.60880500000002	2017-01-12 05:57:00	7ksample_pflow.csv	280

Eliminar registo

Tem a certeza de que pretende eliminar este registo?

Items per page: 10 1 – 10 of 5354

© 2020 Grama. Todos os direitos reservados.

Figure 5.25: Modal that is opened to confirm a register deletion. This is triggered by a click on the trash icon.

In order not to accidentally delete data, when the user clicks on the delete record icon, a modal opens in which it is possible for the user to confirm or cancel the operation, as shown in figure 5.25.

5.4.5 Authentication

As only a data provider can perform the functions mentioned above, a web application authentication system had to be implemented. To this end, the AWS Cognito service was used to store all data relating to these data providers and to facilitate the entire login process, such as creating the session token and validating it.

Five functions were developed: registration, login, reset password, edit personal data and edit password. To register a new data provider, an API Gateway endpoint was developed for that purpose. This end point can only be accessed by those who have access to the AWS account where this project was developed. This limitation is due to the fact that only data providers need a registration, and thus their control is easier, since only a person with access to the AWS account can register a new provider. Having access to this account, the user only needs to define an email, a name and a password to register a new data provider. With the existence of at least one user already registered through the AWS account, login can now be made.

A login page was developed in order to give access to all pages that need authorization, such as data upload, visualization, edit and removal pages.

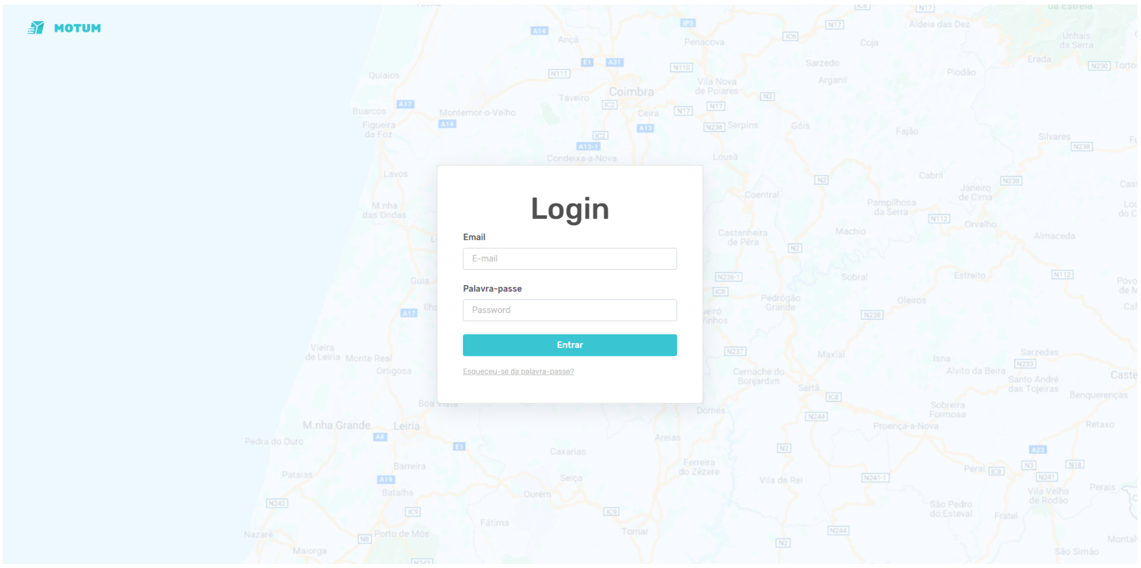
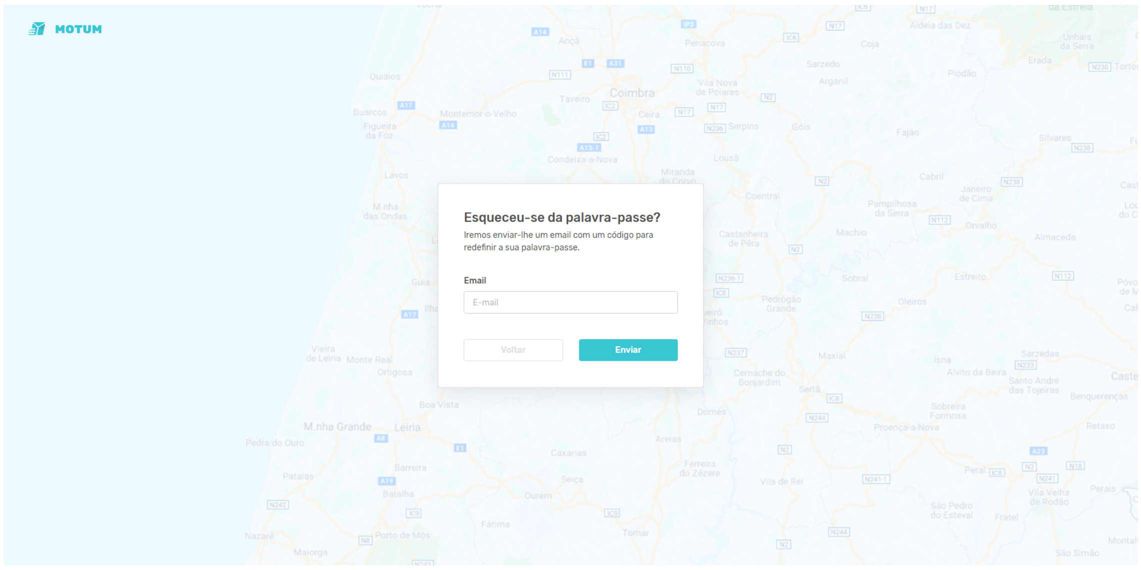


Figure 5.26: Login page.

In figure 5.26 it is possible to see the login page. When a user enters his email and password, the server validates the credentials and, if the credentials are valid, returns a token. This session token will be responsible for identifying whether or not a particular user is authorized to access a page. For this, the front end starts sending the token on all subsequent request headers sent to the back end. In case the user enters invalid credentials, an error message is displayed notifying the user that the email and/or password are incorrect.

In order for the user to recover his password, two other pages were implemented.

By clicking "Esqueceu-se da palavra-passe?", the user is redirected to the page represented in figure 5.27. On this page, the user needs to insert his account email in order to receive


Esqueceu-se da palavra-passe?

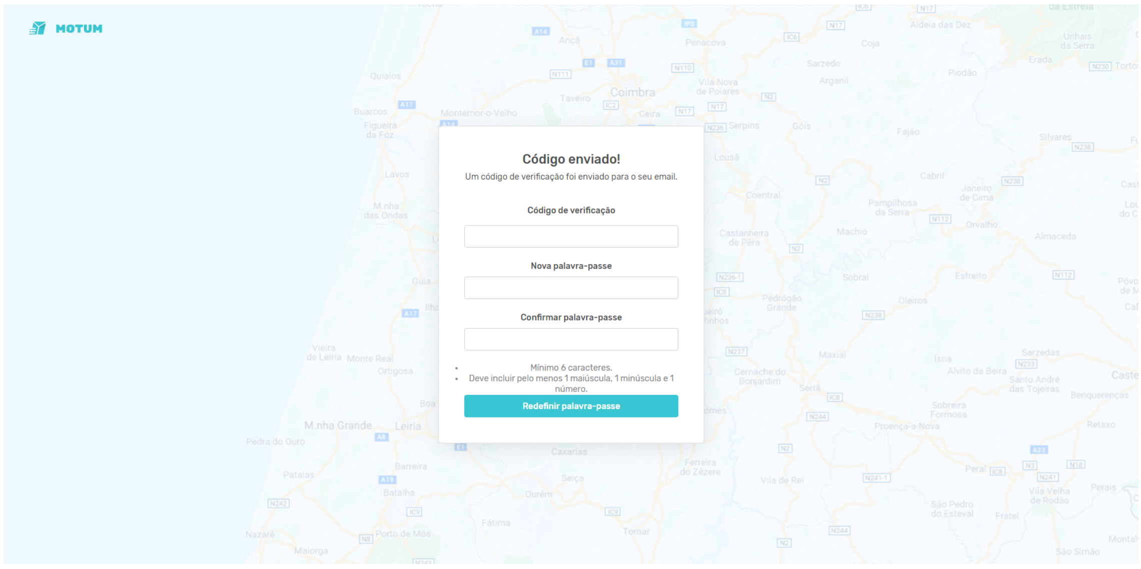
Iremos enviar-lhe um email com um código para redefinir a sua palavra-passe.

Email

E-mail

Figure 5.27: First forgot password page where a Data Provider must enter the email.

in that email a code sent by the back end. This code is generated by AWS Cognito, which saves it so that it can be compared with the code entered by the user on the following page.


Código enviado!

Um código de verificação foi enviado para o seu email.

Código de verificação

Nova palavra-passe

Confirmar palavra-passe

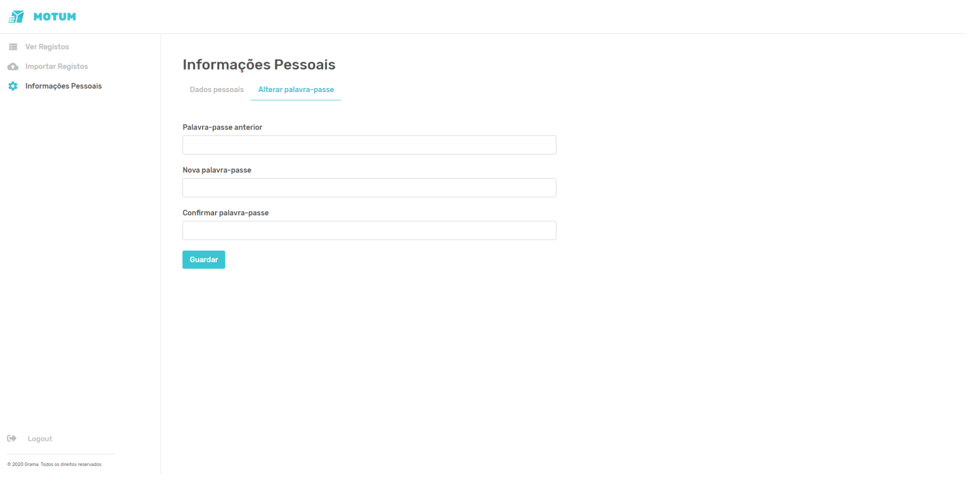
- Mínimo 6 caracteres.
- Deve incluir pelo menos 1 maiúscula, 1 minúscula e 1 número.

Figure 5.28: Second forgot password page where a Data Provider must enter the received code and set his new password.

As previously mentioned, after the user receives a code in the email, that code must be inserted on the page represented in figure 5.28. On that same page you also need to enter a new password and confirm it. When everything is confirmed and sent, the code and password are received by the back end, where the veracity of the code will be verified. If so, the password is changed to the new one sent by the user.

That said, the user can login with the new password. After logging in, the user can change his password in the personal settings page. To do this, you need to navigate to the account settings and click on change password.

In figure 5.29, it is noticeable that for the user to change his password, the old password



Informações Pessoais

Dados pessoais | Alterar palavra-passe

Palavra-passe anterior

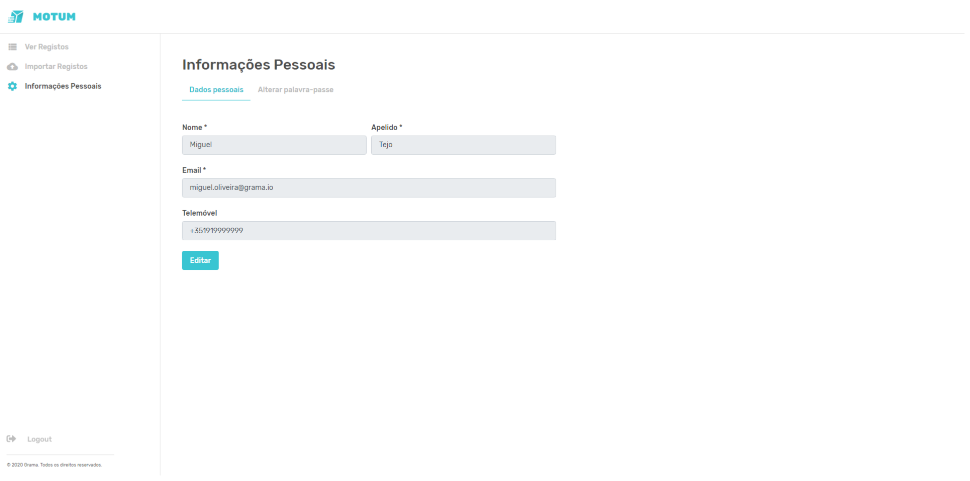
Nova palavra-passe

Confirmar palavra-passe

© 2020 Grama. Todos os direitos reservados.

Figure 5.29: Change password page for a logged in Data Provider.

and a new one are necessary. The back end, upon receiving this data, validates the old password, and if so, changes the password to the new one provided by the user. Also in the account settings tab, the user has the functionality to change his personal data, such as the first and last name.



Informações Pessoais

Dados pessoais | Alterar palavra-passe

Nome * Miguel

Apelido * Tejo

Email * miguel.oliveira@grama.io

Telemóvel +351919999999

© 2020 Grama. Todos os direitos reservados.

Figure 5.30: Page in which a Data Provider can change his personal settings.

As can be seen in figure 5.30, when clicking on personal data, all editable fields are filled with the current information to inform the user about the data present in Cognito and also to allow users to change only one of the fields. When clicking save, all new personal data is updated in AWS Cognito.

5.4.6 Tokyo inhabitants locations visualization

Data visualization is one of Motum Web App's strengths and, therefore, one of the requirements of the project was the representation of historical data, both for transport network infrastructures and people's mobility. To that end, a map was developed using the ngx-leaflet [87] package, which is open source for Angular IO. The used map layer has a custom style in from MapBox [88]. MapBox allows the selection of everything that a map needs to represent a use case, such as buildings, points of interest, road network, etc.

Thus, by using heat maps and marker clusters, it was possible to represent on a map the location of the inhabitants of Tokyo throughout the day and visually assess where it was accentuated. Google has a research paper on human mobility representation [89]. They claim that de-identified aggregate flows of populations around the world can now be computed from phones' location sensors at a uniform spatial resolution. Then they create mobility maps, which can be considered heatmaps. Hence the inspiration to also create heatmaps of the mobility records that the Open PFLOW data set has. Jenny et al. [90] also study the representation of mobility in a map. They compare straight lines to curved lines between origins and destinations. Their findings showed that (a) curved flows are more effective than straight flows, (b) arrows indicate direction more effectively than tapered line widths, and (c) flows between nodes are more effective than flows between areas. Given that in the case of the Motum web app, the library used only contained polylines, straight lines were implemented between the middle point of the areas in Tokyo, which can be interpreted as nodes. In addition, in [91] it is said that "flow on 2D OD flow maps is commonly shown by a straight line from origin to destination with line width encoding magnitude of flow and an arrowhead showing direction", but it turns out that there may inevitably be visual clutter and line crossings. Using curved instead of straight lines is a way to overcome such problems. That is why, in the chapter of Future Work 7.2, the implementation of curved lines is given as a future implementation, since the final product of this internship is only a first version. For this, a library was later found, Canvas-Flowmap-Layer, which extends the ArcGIS API for JavaScript (Esri JSAPI) to map the flow of objects from an origin point to a destination point by using Bezier curves. In the github of the Canvas-Flowmap-Layer, the advantages of using Bezier curves over other curves to represent mobility are substantiated.

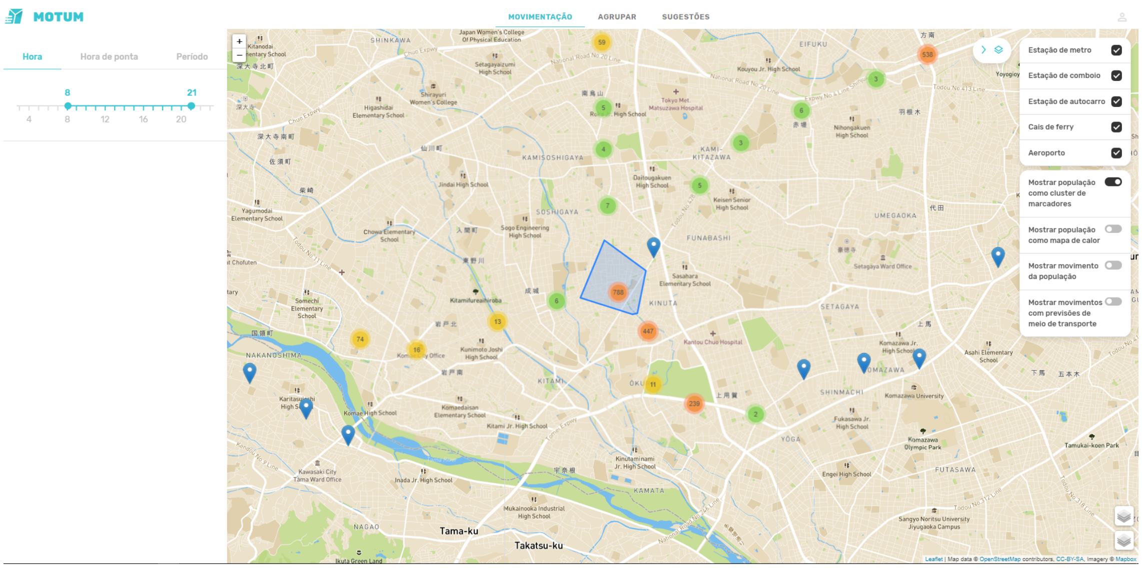
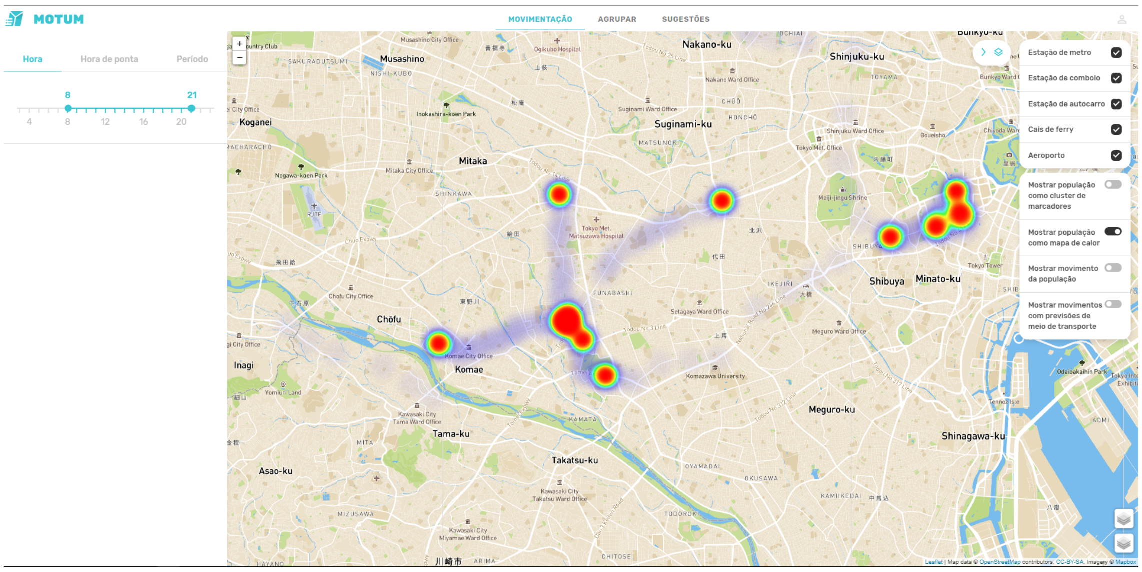

MOTUM
MOVIMENTAÇÃO
AGRUPAR
SUGESTÕES
Hora
Hora de ponta
Período
8
21
4
8
12
16
20
Estação de metro
Estação de comboio
Estação de autocarro
Cais de ferry
Aeroporto
Mostrar população como cluster de marcadores
Mostrar população como mapa de calor
Mostrar movimento da população
Mostrar movimentos com previsões de meio de transporte
Tama-ku
Takatsu-ku

Figure 5.31: Clusters of markers represented in a map.

In figure 5.31, it is possible to see a map in which the numbered circles contain the number of inhabitants' locations throughout the day in that specific place. It should be noted that on the left there are filters that allow filtering by hour, rush hour or not, and part of the day. Requests for daily mobility or trips that the front end makes to the back end are always sent with these filter values. In this specific feature, it is also possible to hover the circle groups and get an approximation of the area where these locations are.

In figure 5.32, it is clear that it is a heat map in which there is a color coding to represent different values. A more reddish color represents a high number of records of mobility at



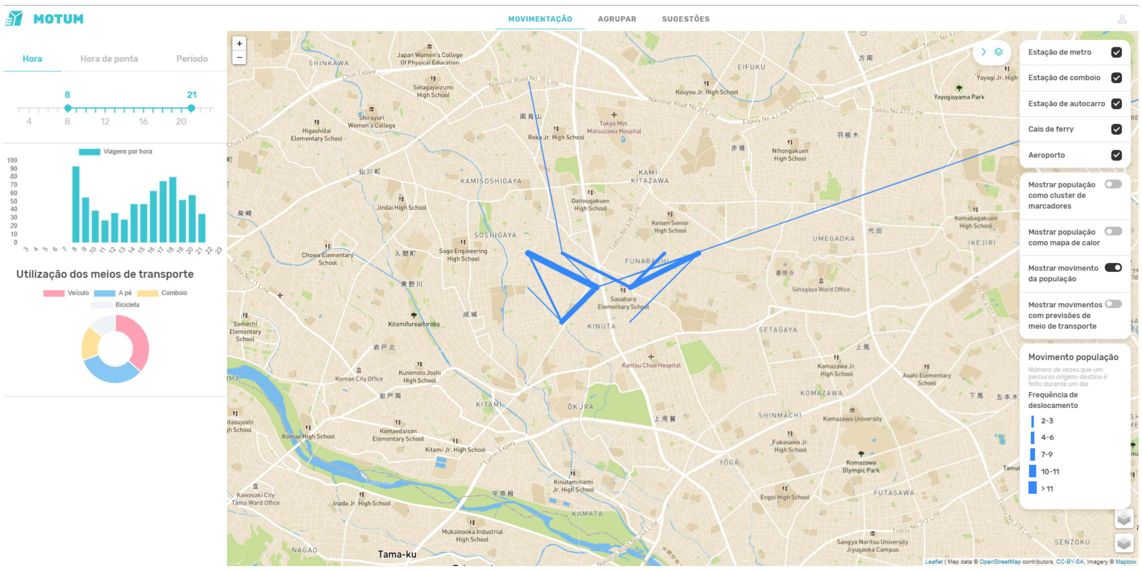
Estação de metro ✓
Estação de comboio ✓
Estação de autocarro ✓
Cais de ferry ✓
Aeroporto ✓
Mostrar população como cluster de marcadores
Mostrar população como mapa de calor ✓
Mostrar movimento da população
Mostrar movimentos com previsões de meio de transporte

Figure 5.32: Heat map for the locations of people throughout the day.

that site, while less intense and more opaque colors represent the opposite.

5.4.7 Tokyo inhabitants trips visualization

With the large number of trips that may possibly exist, these had to be previously grouped by their origin and destination cells, that is, trips with these two equal characteristics are represented by only one record that contains the number of trips made between a certain origin and certain destiny. In addition to the grouping, these trips are aggregated in terms of part of the day, transport, average velocity, average duration and distance. This in order to obtain only one record that contains the summary of several trips between the same origin and the same destination and thus not having to contain all trips in the front end. It would be inconceivable to have all the trips on the front end - the map would not hold up.

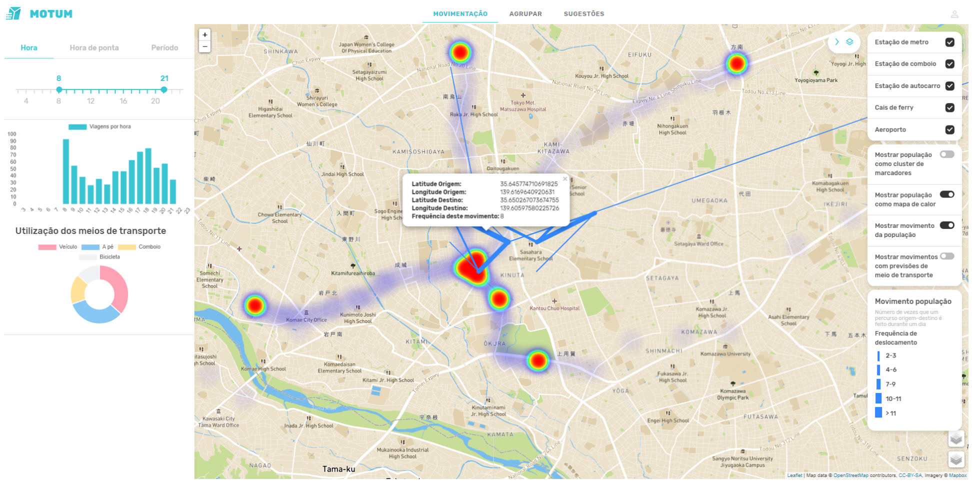


Estação de metro ✓
Estação de comboio ✓
Estação de autocarro ✓
Cais de ferry ✓
Aeroporto ✓
Mostrar população como cluster de marcadores
Mostrar população como mapa de calor
Mostrar movimento da população ✓
Mostrar movimentos com previsões de meio de transporte

Movimento população
Número de vezes que um percurso origem-destino é feito durante um dia
Frequência de deslocamento
2-3
4-6
7-9
10-11
>11

Figure 5.33: Map with people's trips according to the selected filters on the left.

To represent the mobility of people in Tokyo, it is visible in figure 5.33 that polylines were used. The thicknesses of these lines visually indicate the number of times that these trips with the same origin and destination were made. The zoom that is done on the map helps a lot in the perception of what is happening on the map. The more you zoom in, the better you can see which area of Tokyo is represented and a better analysis can be made.



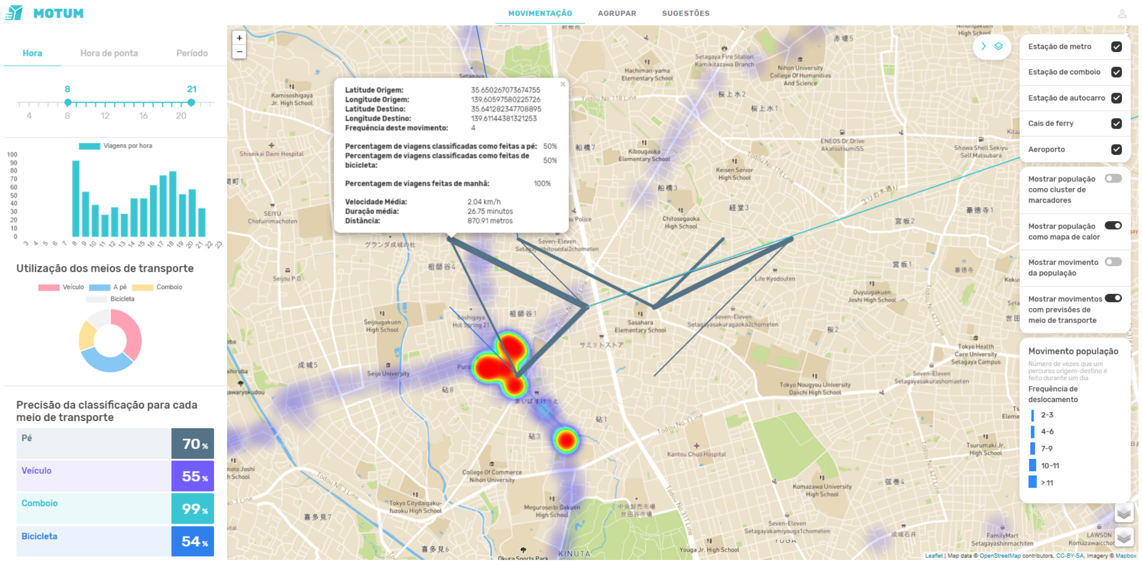
Latitude Origem: 35.645774710691825
Longitude Origem: 139.6169640920631
Latitude Destino: 35.650267073674755
Longitude Destino: 139.60597580225726
Frequência deste movimento: 8

Estação de metro ✓
Estação de comboio ✓
Estação de autocarro ✓
Cais de ferry ✓
Aeroporto ✓
Mostrar população como cluster de marcadores
Mostrar população como mapa de calor ✓
Mostrar movimento da população ✓
Mostrar movimentos com previsões de meio de transporte

Figure 5.34: Map with trips together with the heatmap of people's daily locations.

In figure 5.34 not only are trips visible, but it is also possible to view them along with other map features - heatmap or clusters of markers. Also, by clicking on the polyline of a trip, a popup opens with information about the origin, destination and frequency which this trip is made. On the left side, using charts, statistics are shown for trip count per hour and with what mode of transport. These charts are based on the trip data currently being filtered on the left.

5.4.8 Trips transport mode classification



Latitude Origem: 35.650267073674755
Longitude Origem: 139.60597580225726
Latitude Destino: 35.641282347708895
Longitude Destino: 139.61144381321253
Frequência deste movimento: 4

Percentagem de viagens classificadas como feitas a pé: 50%
Percentagem de viagens classificadas como feitas de bicicleta: 50%

Percentagem de viagens feitas de manhã: 100%

Velocidade Média: 2.04 km/h
Duração média: 26.75 minutos
Distância: 870.91 metros

Estação de metro ✓
Estação de comboio ✓
Estação de autocarro ✓
Cais de ferry ✓
Aeroporto ✓
Mostrar população como cluster de marcadores
Mostrar população como mapa de calor ✓
Mostrar movimento da população
Mostrar movimentos com previsões de meio de transporte ✓

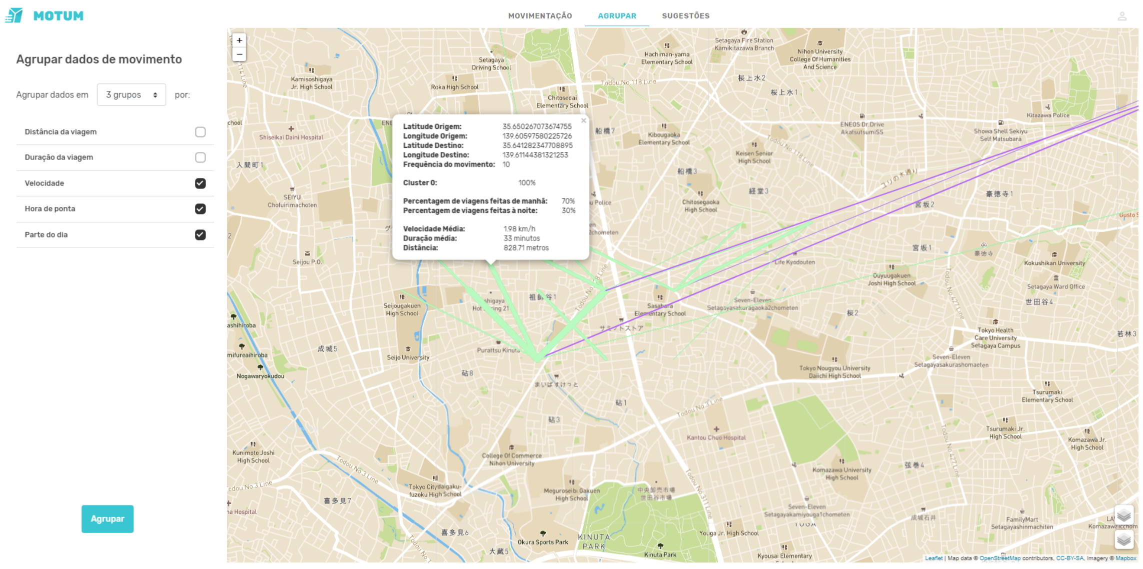
Precisão da classificação para cada meio de transporte

Pé	70%
Veículo	55%
Comboio	99%
Bicicleta	54%

Figure 5.35: Map with trips already classified with the transport modes that are most likely to have been used.

In figure 5.35 we have the page where the user can request the transport modes classifications. These are being done in a very small data set and using a Decision Tree model trained with trips of 40 thousand agents, just to demonstrate the functionality. All due to performance reasons. To request the classifications, there is a switch button on the right side of the screen with the label "Mostrar movimentos com previsões de meio de transporte". When this button is clicked, all movements between different cells in Tokyo are retrieved with their modes of transport classifications. The pop-up of each movement on the map shows the percentage of movements that were classified with a certain transport mode and that make up this large movement. Along with these classifications, performance metrics come from the back end. In this case, the precision of each classification for each mode of transport is shown on the left side of the screen.

5.4.9 Clustering of Trips



Agrupar dados de movimento

Agrupar dados em 3 grupos por:

Distância da viagem	☐
Duração da viagem	☐
Velocidade	✓
Hora de ponta	✓
Parte do dia	✓

Latitude Origem:	35.650267073674755
Longitude Origem:	139.60597580225726
Latitude Destino:	35.641282347708895
Longitude Destino:	139.61144381321253
Frequência do movimento:	10
Cluster 0:	100%
Percentagem de viagens feitas de manhã:	70%
Percentagem de viagens feitas à noite:	30%
Velocidade Média:	1.98 km/h
Duração média:	33 minutos
Distância:	828.71 metros

Figure 5.36: Page where users can select combinations of features and cluster by them.

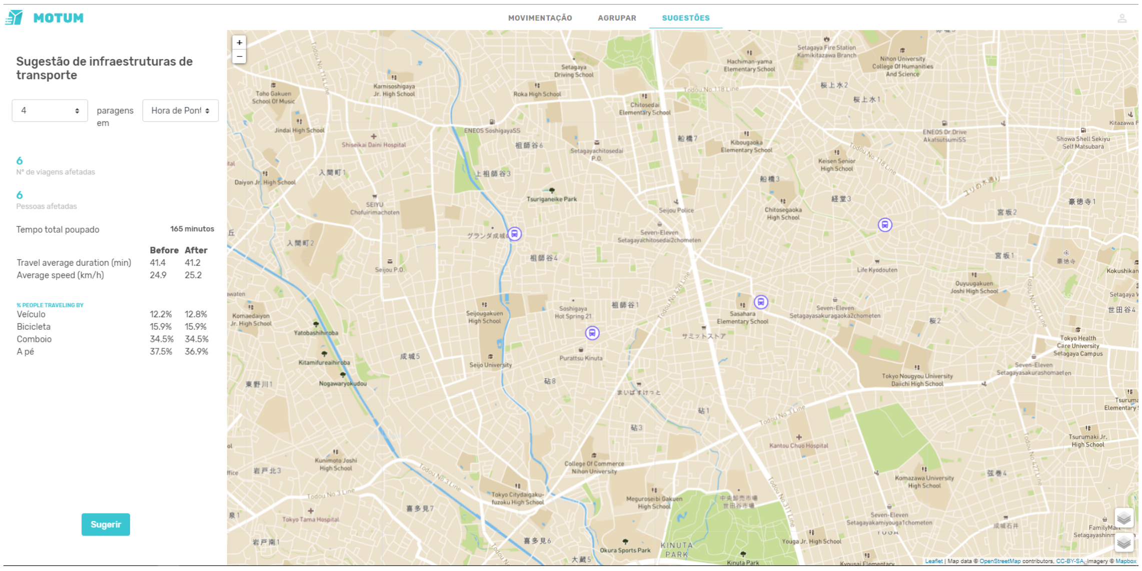
In figure 5.36 the page where it is possible to group trips in a customized way is shown. User can select the characteristics of the trips he wants to be grouped and also the number of clusters to be formed.

Also in this feature, trips, before being sent to the front end, are also grouped by origin and destination again for performance reasons. For this reason, trips that are made from the same origin to the same destination are aggregated by the cluster label to which they belong. For example, there can be 10 trips from origin A to destination B with different cluster labels and their aggregation has 70% trips belonging to cluster 1 and 30% belonging to cluster 2. In this case, these trips would be represented with the color of cluster 1, since it represents the majority.

5.4.10 Suggestion of locations for new transport infrastructures

More specifically, the suggestion developed was related to bus infrastructure.

Figure 5.37 represents the page where it is possible to ask for suggestions of locations for new bus infrastructures. It is possible to select the desired number of suggestions and whether this suggestion is to be made based on trips made during normal or peak hours. It was thought to be interesting to analyze the movements with a certain minimum



Sugestão de infraestruturas de transporte

4 paragens em Hora de Pont

6
Nº de viagens afetadas

6
Pessoas afetadas

Tempo total poupado 165 minutos

	Before	After
Travel average duration (min)	41.4	41.2
Average speed (km/h)	24.9	25.2

% PEOPLE TRAVELING BY		
Veículo	12.2%	12.8%
Bicicleta	15.9%	15.9%
Comboio	34.5%	34.5%
A pé	37.5%	36.9%

Figure 5.37: Page where users can ask for new transport infrastructure locations.

distance and made on foot at peak hour. These can possibly indicate cases where there is no modes of transport at the origins or destinations where a person has moved. This page also shows statistics for before and after the hypothetical insertion of the infrastructures in the network and the consequent alteration of the existing trips. Thus, it is possible to assess the impact that these new bus stops would have on people's movement. Finally, infrastructures like current bus stops in Tokyo are not being shown due to struggles caused by the large amount of records there is in the city. The map was having crashes caused by those infrastructures.

Chapter 6

Tests and Results

This chapter presents the test approaches and their results for both the Clustering phase and the Classification phase.

6.1 Clustering of trips

To test the algorithms developed for clustering trips data, around 100 thousand trips from the data set already described in 5.1 were used. These trips assume that the zones of Tokyo are represented by cells of 500 meters.

A very important problem that arises after the clustering of trips is the evaluation of their results to find the partitioning that best fits the underlying data. Cluster Validity is the name given to this process. For this, in order to substantiate the choices made in this part of the project, a literature review on clustering validation techniques was consulted (Halkidi et al, [92]).

In case we are in the presence of data sets with more than two dimensions, visualizing the data set becomes a difficult task. In 2D data sets, it is usually easier to evaluate the results visually.

Also depending on the features of the data set and the parameters of the clustering algorithms, the term "optimal" clustering scheme can be defined as "the outcome of running a clustering algorithm that best fits the inherent partitions of the data set" [92]. If the clustering algorithm parameters have improper values, the clustering may result in a clustering scheme that is not optimal for the specific data set and consequently bad decisions will be made from it.

According to Halkidi et al. [92], there are three approaches to investigate cluster validity. First, there is one based on **external criteria**. According to this approach, results of a clustering algorithm are evaluated based on a "pre-specified structure, which is imposed on a data set and reflects our intuition about the clustering structure of the data set". The second one is based on **internal criteria**. Results of a clustering algorithm may be evaluated in terms of" quantities that involve the vectors of the data set themselves (eg proximity matrix) ". The third approach is based on **relative criteria** Here, the clustering structure is evaluated by comparing it to other clustering schemes, resulting by the same algorithm but with different parameter values.

Two criteria were proposed for clustering evaluation and selection of an optimal clustering scheme in Berry and Linoff, 1996 [93]:

- **Compactness**, which dictates that the members of each cluster should be as close to each other as possible.
- **Separation**, which dictates that clusters themselves should be widely spaced.

Since the problem is not supervised, validity indices which fall within internal criteria based approaches were used. Arbelaitz et al. [94] 2013 have exhaustively tested a variety of indices. The study concludes that Davies-Bouldin, Silhouette and Calinski Harabasz perform well in a substantial variety of situations.

- **Silhouette**, guided by the maximum class spread/variance. Clusters' cohesion is measured based on the distance between all the points in the same cluster and the separation is measured based on the nearest neighbour distance. Its formula takes into account an average distance between each point within a cluster and an average nearest-cluster distance for each point (Rousseeuw 1987 [95]). Values can range from -1 to 1. While 1 is the best possible result, -1 is the worst. Values close to 0 indicate overlapping clusters.
- **Davies-Bouldin**, which calculates the average similarity measure of each cluster with its most similar cluster. The similarity measure is the ratio of within-cluster distances to between-cluster distances (Davies and Bouldin 1979 [96]). With a minimum value of zero, lower values indicating better clustering.
- **Calinski-Harabasz**, which utilizes both the ratio between the within-cluster dispersion and the between-cluster dispersion (Calinski and Harabasz 1974 [97]). The higher its values, the better the result of clustering.
- **Inertia**, which scikit-learn KMeans' website defines as the sum of squared distances of samples to their closest cluster center [98].
- **Clustering cost**, defined by Nico de Vos in his Python implementation [99] as the sum of distances of all points to their respective cluster centroids.

When calculating Silhouettes, the hardware used for testing had some struggles. The maximum available memory has been reached using both Ubuntu 18.04 and Windows 10. In the Silhouette calculation, all data is iterated twice and therefore, its complexity is O(N^2n) [100]. For large data sets of at least hundreds of thousands of observations, the complexity of the calculation can be even greater than that of the clustering task. However, good scores in the internal criteria mentioned above do not necessarily mean that a clustering has good effectiveness in an application.

For the various Clustering algorithms used, tests that varied their respective hyperparameters were performed. For k-prototypes, as it is written in the authors' paper [70], it is important to vary the number of clusters to be created and the gamma, which is a weighing factor that determines relative importance of numerical against categorical features. There are also two variations of this algorithm: Cao and Huang. Both were tested repeatedly to evaluate Silhouette. The one with acceptable results was Cao, while Huang's k-modes produced negative silhouettes.

DBScan was also one of the algorithms used. Several minimum points were tested and the epsilon was always calculated, as previously explained. From there, the number of clusters is decided by the algorithm. The automatic epsilon calculation was done as follows. The k value, which corresponds to mininum points, is previously chosenand nearest neighbours algorithm is then applied for each point. Then, the kth distances of every point are gathered and sorted in ascending order. Then, these k distances are plotted in an ascending order in order to determine the *knee* - where the optimal epsilon parameter lies.

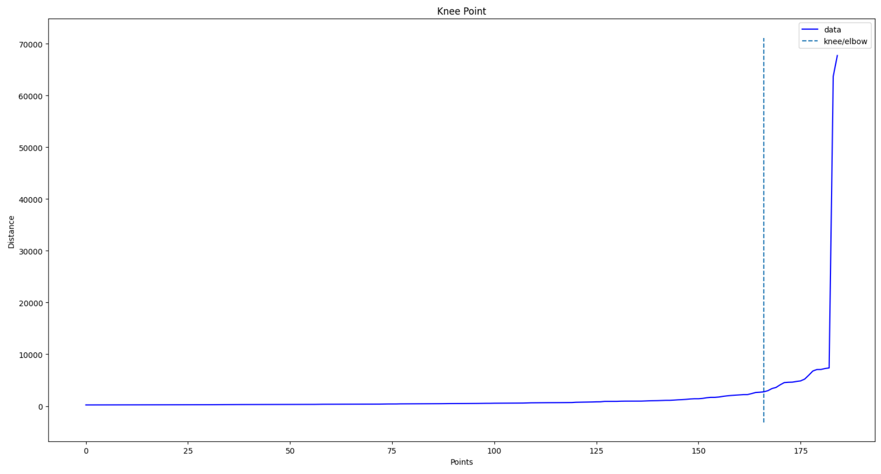


Figure 6.1: Example of a knee point.

This *knee*, presented in figure 6.1, corresponds to a threshold where an abrupt change occurs along the distances curve. This was done systematically by using the *kneed* technique presented in a paper by Ville Satopaa et al. [101]. In that paper, the technique aims to find the elbow point of an increasing progression of values. The *kneed* python library that implements this technique was used [102].

Better results were obtained when few clusters were formed. Finally, the distance metric between the various objectives to be clustered was varied between Euclidean and Manhattan.

With the use of k-means, several values for k were tested, thus verifying which one would produce the best performance metrics. Various combinations of features have also been tested.

The first test presented is a test where clustering validation metrics tell us that clusters do not represent well-separated groups. The test was done with two features of different types: starting hour and duration. The results obtained for k = 2 were as follows.

Algorithm	**Cost**	**Silhouette**	**Davies-Bouldin**	**Calinski-Harabasz**
k-prototypes	97679.7	-0.107	254.2	6.68

Table 6.1: Bad k-prototypes results with k = 2 where starting hour and trip duration were used.

In table 6.1, a negative silhouette and close to 0 is seen, which respectively indicates that there are badly assigned clusters and that there is overlapping clusters. In addition, the Davies-Bouldin value should be low but is quite high, while the Calinski-Harabasz value should be high but is very low.

Figure 6.2 represents the features that have been clustered by k-prototypes. It is possible

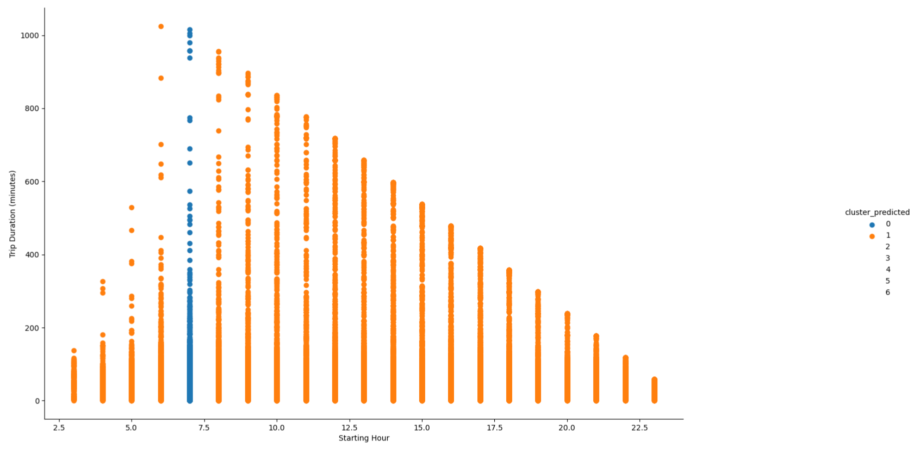


Figure 6.2: Two clusters identified in trip duration and starting hour data.

to see that the division does not seem very well done, as the validation metrics of this clustering also indicate. Trips made at 7 pm are in a different group from every other trip.

While the representation of the previous clusters was about poor results, next is a clustering that seems to have been successful, in which we can notice a segregation of trips.

Algorithm	**Cost**	**Silhouette**	**Davies-Bouldin**	**Calinski-Harabasz**
k-prototypes	158565545.9	0.895	0.420	156060.3

Table 6.2: Successful k-prototypes results with k = 2 where starting hour and trip duration were used.

Algorithm	**Inertia**	**Silhouette**	**Davies-Bouldin**	**Calinski-Harabasz**
k-means	155474735.1	0.8957	0.419	156062.8

Table 6.3: Successful k-means results with k = 2 where starting hour and trip duration were used.

In tables 6.2 and 6.3, results are very similar, even though those are performed by different algorithms. It is seen a positive and close to 1 silhouette, which means clusters are correctly assigned and that there is close to no overlap of clusters. In addition, the Davies-Bouldin value is quite low and the Calinski-Harabasz value is really high, which is what we want.

Figure 6.3 represents the features that have been clustered by k-prototypes with the gamma parameter being automatically calculated by the algorithm. The same representation, but for k-means, is equal to the naked eye. It can be seen that the algorithm chose to perform the division mostly taking average velocity into account, and the results were good, as the result metrics already indicated.

Then k-means clustering was applied with k = 6. Trip duration, distance and average velocity were used as features. Six clusters are represented in a different color in figure 6.4. These do not have much inter-cluster distance, but it is assumed that the within-cluster distance is even smaller, given that the metrics tell us that it was an acceptable clustering. In more extreme cases in terms of both duration and average velocity, there are trips that are overlapped with trips from another cluster.

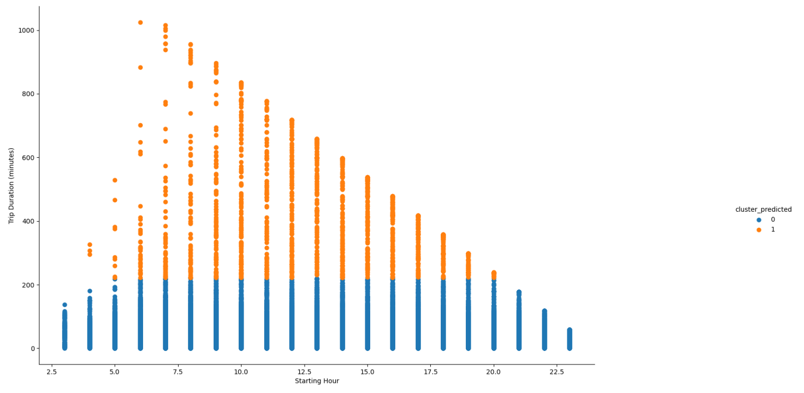


Figure 6.3: Two clusters identified in trip duration and starting hour data.

Algorithm	Inertia	Silhouette	Davies-Bouldin	Calinski-Harabasz
k-means	951256600268.7	0.669	0.529	293685.2

Table 6.4: Successful k-means results with k = 6 where trip duration, distance and average velocity were used.

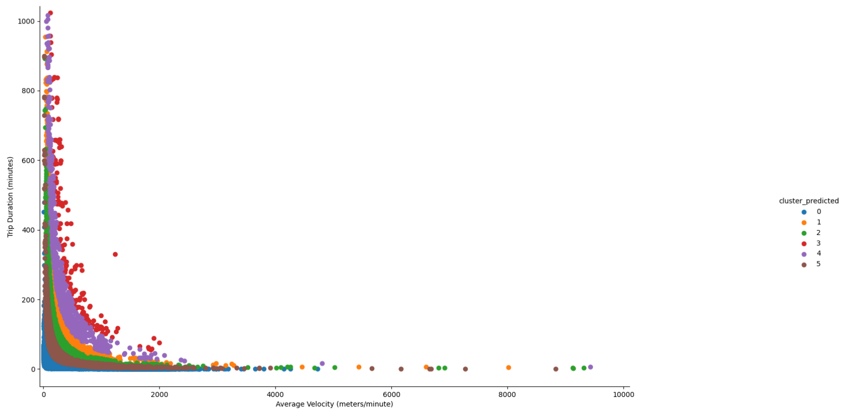


Figure 6.4: Six clusters identified in trip duration and starting hour data.

A clustering by three features can be analyzed differently. The following clustering was performed on trip duration, distance and average velocity.

Algorithm	Inertia	Silhouette	Davies-Bouldin	Calinski-Harabasz
k-means	5745502024890	0.803	0.5114	155109.5

Table 6.5: K-means results with k = 2 where trip duration, distance and average velocity were used.

As 3 features were used to perform this k-means, three three-dimensional charts were created for the clusters produced. Each one of them represents a different angle to emphasize each of the features used in clustering.

In the angle of figure 6.5, it is possible to notice that the cluster with reddish color has

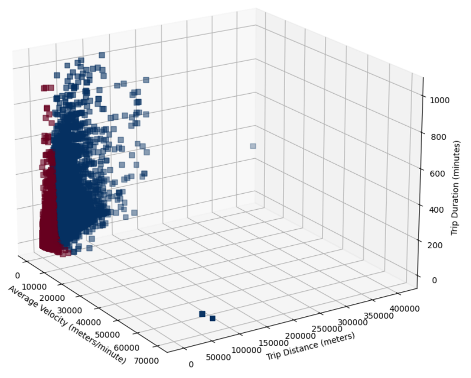


Figure 6.5: Angle 1 from tridimensional chart of two clusters identified in trip duration, distance and average velocity.

relatively low values of average velocity and distance, while its travel duration values may fluctuate between the lower half of the value spectrum. The duration ones are between 0 and 500 minutes mostly - there are higher values but quite spread out up to 900 minutes.

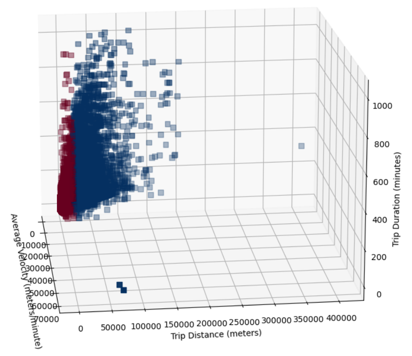


Figure 6.6: Angle 2 from tridimensional chart of two clusters identified in trip duration, distance and average velocity.

In the angle of figure 6.6, it is possible to see that the distance values are between 0 and 50000 meters.

In the angle of figure 6.7 the average velocity values seem to be mostly lower in the red cluster compared to the blue. The red cluster only has high speed values when the trip duration is short.

For mixed type of features, another technique was used for the analysis, which will be

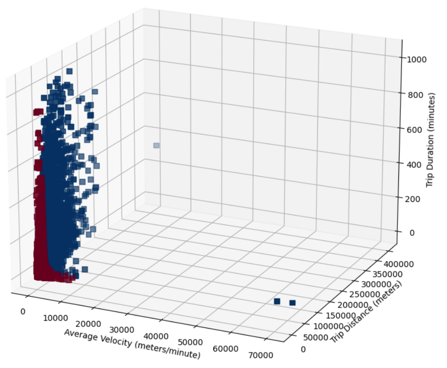


Figure 6.7: Angle 3 from tridimensional chart of two clusters identified in trip duration, distance and average velocity.

presented next. The significance of the clusters has to be discovered, as the algorithm gives no indication of this. A summary or characterization of each cluster needs to be done, so **parallel coordinates plots** were used to visually check what values each feature takes in the different clusters. Sagar et al. [103] also use it for multivariate visualization. They utilize D equidistant parallel axes to visualize each feature of simple K-means clusters and project the D-features on a two-dimensional surface. The axes correspond to the features and are scaled from the minimum to the maximum value of the corresponding feature. Feature values are presented as polygonal lines, intersecting each of the axes at a point that corresponds to the value of the considered feature.

The following clustering is also done via k-means with k = 4. The features used were average velocity, distance, duration, part of the day and rush hour.

Algorithm	**Silhouette**	**Davies-Bouldin**	**Calinski-Harabasz**
k-means	0.697	0.518	229646.7

Table 6.6: K-means metrics results with k = 4 where trip duration, distance, average velocity, part of the day and rush hour were used.

In table 6.6, it is seen a positive and close to 1 silhouette, which means clusters are correctly assigned and that there is close to no overlap of clusters. In addition, the Davies-Bouldin value is low and the Calinski-Harabasz value is high, which is what is appropriate.

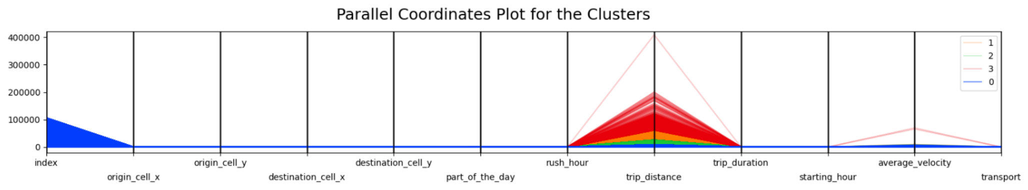


Figure 6.8: Parallel coordinates plot for 4 k-means clusters.

Through the parallel plot 6.8, it is possible to see differences between the features - mainly in the numerical ones. The biggest difference lies in the trip distance. Categorical variables have very different scales from numerical variables, so this plot does not show differences for this type of features. It is possible to see that cluster 3 has extremely high values for both average velocity and trip distance. Perhaps because the range of trip duration values is lower than that of velocity and distance, the difference of this feature across the 4 clusters produced also cannot be seen.

One parallel plot for each cluster label was created. In each one, the respective cluster label is emphasized, although from only a parallel plot an idea of the values of each feature for each label cluster can be obtained. To check the differences between numerical features, a 3D chart was also produced.

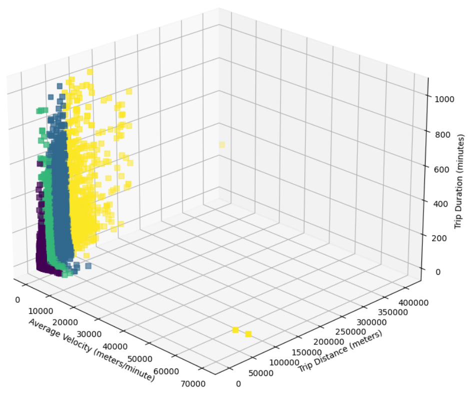


Figure 6.9: Tridimensional scatter plot of four clusters identified in trip duration, distance, average velocity part of the day and rush hour.

In figure 6.9 it can be seen that the purple cluster has values across the entire average velocity spectrum, while presenting relatively small values for duration and trip distance. The other three clusters have very similar average velocity and duration values. The only thing that distinguishes them is the distance values for their trips. It is also possible to conclude that the yellowish cluster, which has the longest trip distance, is the least compact cluster; also contains all trips with extreme values for their characteristics.

To get an idea about the relationship between categorical variables and to clarify the doubts about numerical ones, the only way would be to inspect the centroids that k-means produced, so that is what will be presented next.

Cluster #	**Distance (meters)**	**Duration (minutes)**	**Avg. Velocity (km/h)**
0	2748.5	37.0	15.2
1	37238.1	121.1	42.7
2	14039.8	32.3	50.0
3	85955.2	297.6	40.4

Table 6.7: K-means centroids for k = 4. Centroids' values for trip duration, distance and average velocity are shown.

With this analysis, several conclusions were drawn that could not be drawn by inspecting the previously developed charts. For example, contrarily to what was thought, there is

Cluster #	Morning	Afternoon	Evening
0	0.457	0.338	0.304
1	0.468	0.341	0.189
2	0.438	0.332	0.228
3	0.454	0.404	0.140

Table 6.8: K-means centroids for k = 4. Centroids' part of the day values are shown.

Cluster #	Not Rush Hour	Rush Hour
0	0.588	0.411
1	0.633	0.366
2	0.565	0.434
3	0.692	0.307

Table 6.9: K-means centroids for k = 4. Centroids' rush hour values are shown.

a difference in numerical features other than trip distance. It can be said that cluster 0 and 2 have low durations (37 and 32.3 minutes), cluster 1 has a median duration (of 121.1 minutes) and cluster 3 has the highest (297.6 minutes).

In short, cluster 0 can represent a group of people who travel for a short time, for short distances and at a medium average velocity. Cluster 1 can represent a group of people who make trips of medium distance and duration, but at a medium-high speed. Cluster 2 can represent a group of people who make trips of short distance and duration but at a great average speed. Cluster 3 can represent a group of people who travel very long distances and for high durations, at a medium-high speed.

It is possible to notice that a division was not made taking into account the part of the day. All clusters have more trips in the morning than in other parts of the day. Regarding whether they are rush hour trips or not, all clusters have more trips in non-rush hour hours than in those that are. Cluster 3 has most of the trips outside rush hours - almost 70 % of them. Cluster 2 has the most rush hour travel - around 43 %.

In this section, both analytical and visual ways of evaluating clustering results and interpreting them were presented. Many of these charts can serve as an aid to detect patterns in the Motum web app, such as groups of people who have a characteristic and similar activity across the population, which is what was tried to do in this section. At the moment they are not being used in the web app, but in the future they may serve as support to clarify the user about the clustering produced in real time.

6.2 Transport Mode Classification

6.2.1 Testing Approach

To test the algorithms developed for transport mode classification, around 415 thousand trips from the data set already described in 5.1 were used. These trips assume that the zones of Tokyo are represented by cells of 500 meters. However, there are other parameters that have nothing to do with the Machine Learning (ML) algorithm itself and that can be mixed, such as categorical variables being transformed into Dummy Variables or not, the ML algorithm itself, rescaling of data and features to be used (Table 6.10).

To test the models and optimize their hyperparameters, the data set was divided into three

Scaler	no scaling, min max
Dummy variables applied to categorical	yes, no
Algorithm	Random Forest, Decision Tree, LightGBM
Clustering label	yes, no
Bayesian Optimization number of evaluations	10, 15, 20
Features	agent id, origin cell x, origin cell y, destination cell x, destination cell y, part of the day, transport, rush hour, trip distance, trip duration, starting hour, average velocity

Table 6.10: Parameters of the Machine Learning process to vary.

subsets: 1) training set, 2) validation set and 3) test set. Initially the data is randomized and then the division into 3 subsets is performed: 70% for both training and validation sets, and 30% for the test set. The data for the three sets is never intersected. Thus, it is guaranteed that the test set is never used and works as a final test of the models chosen in the validation set model evaluation. This division of the data set is defended by seminal texts in the field, such as the book by Ripley [104], page 354. There, we have training, validation, and test sets defined as:

- **Training** set: As the name suggests, it uses examples to learn and fit the parameters of a classifier.
- **Validation** set: Examples are used to tune the hyperparameters of a classifier. For instance, the number of hidden layers in a neural network, or the number of estimators in a Random Forest.
- **Test** set: A set of examples is used only to evaluate the performance of a fully-specified classifier.

Keeping a test set completely separate is also reiterated by Russell and Norvig in their seminal textbook [105], page 709. There, they advise locking the test set away completely until all model tuning is complete. Then, if the results are not acceptable, a completely new test set as to be locked away once again. Russell and Norvig also comment that "the training data set used to fit the model can be further split into a training set and a validation set, and that it is this subset of the training data set, called the validation set, that can be used to get an early estimate of the skill of the model".

The machine learning algorithms, along with the various combinations of their hyperparameters, are then trained with the training data set. These models are then evaluated using the validation set. This process is iterative so it can include any necessary adjustments to a model based on the results that are being obtained. Hyperparameter tuning with Bayesian Optimization was performed for Decision Tree (DT), Random Forest (RF), Light Gradient Boosting Machines (LGBM) and Artificial Neural Network (ANN). It is proven that ML algorithms' hyperparameters need to be tuned to achieve optimal performances (Bergstra et al. [19] 2012; Duarte et al. [20] 2017; Hutter et al. [21] 2011).

For this phase, 3 hyperparameter optimization techniques were studied, which are explained in the Background Knowledge chapter 2.

Hyperopt-specific distribution functions were used to create the search space of hyperparameters [106]. First, the developed objective function simply takes in a set of hyperparameters and returns a score that indicates how well a set of hyperparameters performs on the validation set, with the macro F-measure being the metric of choice for the evaluation. Therefore, the aim was then to minimize the objective function, since the returned cross validation macro F-measure is a negative value. This is due to the fact that the Hyperopt optimiser requires a value to minimize, thus, maximizing the macro F-measure is equal to minimizing the negative macro F-measure.

There are two variants of the Bayesian Optimization: one based on the Gaussian process and the other on the Tree Parzen Estimator. The package used for implementation was the HyperOpt package [106], which implements the Tree Parzen Estimator algorithm. It takes less training steps in order to achieve a comparable result when compared to random search and grid search with a sufficiently high number of experiments. Therefore, Bayesian Optimization was chosen, as it is the technique that has the best time-quality ratio. A few number of evaluations were ran, due to previously mentioned hardware limitations.

More specifically, the training and validation phases were performed using k-fold cross validation, with k = 10. In cross-validation the data sets are divided into several subsets and the learning models are trained and evaluated in these subsets. The k-fold cross validation is considered a resampling-based technique (James et al. (2013) [13]), in which the data set is divided into k subsets (folds) that are used for training and validation purposes for k iteration times. Each subset will be used at least once as a validation data set and the remaining (k-1) as a training data set. Once all the iterations are completed, the average of the chosen evaluation metric is calculated. This for each model of the k iterations.

The 10 folds are due to the consensus found in the literature. According to James et al. (2013) [13], there will always be a trade-off between bias and variance associated to the k choice in k-fold. Both k = 5 and k = 10 have been "shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance".

For every hyperparameter setting presented in Figure 6.12, model performance (macro F-measure) was computed and averaged across the folds of Group K-fold. The hyperparameter setting with the highest mean F-score result across all folds was used to train a model on 70% of data (training and validation sets). The selected model was then evaluated on the test set to ensure that overfitting did not occur and to assess the generalization capability of the model. It was also important to ensure that, in cases where the results are good, the test/validation sets were not leaking information to the training data set. For instance, the Group K-Fold cross validation was used. This variation of K-Fold cross validation was used since each trip has an ID that refers to the agent to which it belongs. Therefore, trips of the same agent which were in the training fold and validation fold could cause information leakage. Then, the Group KFold uses the agent ID to create groups and each group is entirely in the training or test set, never distributed in the two different sets.

Following this experimental setup, all parameters of the 6.10 table were varied in tests with Group 10-fold cross validation and Bayesian Optimization. For each algorithm, the table 6.12 contains the limits predefined for the respective hyperparameters. The purpose of these tests was to choose the model with the best performance metrics of all - namely precision, recall and F-measure. As already explained, the macro variants of these metrics were used, given that we are in the presence of a problem with class imbalance - which can be seen in figure 6.11.

Table 6.11 shows that the data set is class imbalanced. Walk and Train transport modes have more values than the other modes.

Walk	**Vehicle**	**Train**	**Bicycle**
35.82%	18.66%	29.02%	16.43%

Table 6.11: Percentages of transport modes.

Algorithm	**Hyperparameter**	**Value**	**Start**	**End**	**Step**	**Distribution**
Decision Tree	max_depth	-	2	150	2	-
Random Forest	n_estimators	-	200	900	100	-
	max_depth	-	10	20	5	-
	min_samples_split	[2, 5]	-	-	-	-
	min_samples_leaf	[1, 2]	-	-	-	-
	bootstrap	[True, False]	-	-	-	-
Light GBM	learning_rate	-	0.01	1	-	log uniform
	max_depth	-	4	20	4	-
	n_estimators	-	50	800	50	-
	num_leaves	-	10	150	10	-
	boosting_type	['gbdt', 'dart']	-	-	-	-
	colsample_bytree	-	0.6	1.0	-	uniform
	reg_lambda	-	0.0	1.0	-	uniform

Table 6.12: Hyperparameter limits for every Machine Learning algorithm.

With regard to DT, it was already known that its results would be the worst of the 3 algorithms, so only the maximum depth that the tree can take was varied. Between RF and LGBM, given that the latter was a more novel algorithm and with evidence that it could outperform RF, more belief was placed in this, that is, more hyperparameters were selected to test this algorithm, as can be seen in table 6.12. The limits for each hyperparameter have to do with the time restrictions that existed for the testing phase, since the hardware used for testing was the same that was used both for the implementation of code or writing of the report during the day, therefore the tests were only performed in idle time. There was a fairly trial and error component to define acceptable hyperparameter limits for the time available for the maximum duration of the tests.

6.2.2 Results

From the phase of choosing the best model and hyperparameters tuning, the best model corresponding to each algorithm was selected.

The one that showed the best macro F-measure average throughout the model tuning phase was LGBM using 500m cells, Min Max scaling of numerical features and Dummy Variables transformation on categorical ones.

	LightGBM	
Bayesian Optimization with 10-fold Cross-Validation	F-measure (%)	85.59
Final test with hold-out set	Precision (%)	86.35
	Recall (%)	84.95
	Accuracy (%)	87.94
	F-measure	85.57

Table 6.13: Model with which the best test results were obtained.

Table 6.13 shows that LGBM produced results with a high accuracy - near 80% - but with a lower but still good macro F-measure - 85.57%. For this test all features were used. Regarding hyperparameters, the boosting type used was the traditional Gradient Boosting Decision Tree and the algorithm selected 98.8% of features before training each tree, using the *colsample_ by_ tree* hyperparameter. The learning rate by default is 0.1, but in this case 0.38 was used, which made the learning much more faster. The maximum depth for trees was 8, the number of estimators was 700 and the maximum number of leaves for each tree was 110.

	Random Forest	
Bayesian Optimization with 10-fold cross-validation	F-measure (%)	84.26
Final test with hold-out set	Precision (%)	85.36
	Recall (%)	83.02
	Accuracy (%)	86.82
	F-measure	83.94

Table 6.14: Best test results for the model using Random Forest algorithm.

In 6.14 table, it is noted that the results of the best RF model are similar to those obtained with the best LGBM. Even so, LGBM results are better along with its great speed of training, which makes LGBM better than RF, for this case. For RF, recall dropped a bit and, as a result, the F-measure macro also dropped. Regarding hyperparameters, bootstrap samples were used when building trees, instead of using the whole data set. The maximum depth of trees was set at 35 and the number of trees in the forest was 380. The square root of the number of features was the number of features used to consider when looking for the best split.

	Decision Tree	
Bayesian Optimization with 10-fold cross-validation	F-measure	82.73
Final test with hold-out set	Precision (%)	83.41
	Recall (%)	81.79
	Accuracy (%)	85.35
	F-measure	82.48

Table 6.15: Best test results for the model using Decision Tree algorithm.

As can be seen in table 6.15 that represents the best results obtained with a Decision Tree, the worst results of all the algorithms were obtained. This DT had a maximum depth of 15. Although the accuracy remains close to the results obtained with other algorithms, the recall dropped significantly. As a result, the F-measure macro also drops.

Origin and destination cells were being used for the previous tests, although it is important to perform tests without these features, so that models do not just learn about trips of one place in the world, but produce models for any type of trips data set.

For the test provided in table 6.16, features used were: agent id, part of the day, rush hour, trip distance, trip duration, starting hour and average velocity. Dummy Variables transformation was applied to categorical ones, but no scaling was used. With this change in features, the results dropped a bit, but nothing significant. Regarding hyperparameters, the boosting type used was the traditional Gradient Boosting Decision Tree and the algorithm selected 88.6% of features before training each tree, using the *colsample_ by_ tree* hyperparameter. The learning rate by default is 0.1, but in this case 0.138 was used, which

	LightGBM	
Bayesian Optimization with 10-fold Cross-Validation	F-measure (%)	83.11
Final test with hold-out set	Precision (%)	84.02
	Recall (%)	82.32
	Accuracy (%)	85.75
	F-measure	83.03

Table 6.16: LGBM test results with no grid cells being used as features.

made the learning faster. The maximum depth for trees was 8, the number of estimators was 350 and the maximum number of leaves for each tree was 90.

LGBM model reached a macro f-measure of 83.03%, combined with a precision of 84.02%, a recall of 82.32% and accuracy of 85.75%. In comparison to state-of-the-art studies, the results obtained were considered good, since for the same problem, Apichon Witayangkurn et al. [40] obtained results similar to those of this work - 87.80% and 84% for precision and recall, respectively, while using Bagging and spatial features, which were not used in this work. In another study of the state of the art, Qiuhui Zhu et al. [41] reached an overall accuracy of 82.85% for identifying transportation modes, and, especially for car mode, 91.44% precision and 92.91% recall were reached, which are really good results that might result from their postprocessing method of correcting misidentified modes. Even so, it is noted that the results are not balanced, some transport modes have much better results than others, which does not happen in the work described in this document.

ANN with hyperparameter tuning was also implemented with Keras. However, very few evaluations could be made since they are the algorithm that takes the longest to be trained. Having few hours in a row to perform tests, it was not possible to test them. Though, isolated test results during the development of the project were similar to RF.

6.3 Web Application

Given that the internship is more focused on Artificial Intelligence and the intern already had some experience in front and back end, less emphasis is placed on this part.

Tests on the application were done by several people in the company in order to guarantee its quality. Also, the source code for each part of the application was divided by repositories in GitLab and was revised iteratively by people from the company of different specializations. For each new task developed, a merge request was made, which was only subsequently accepted after review. This Git workflow process was used in all parts of the development, from the front end to the back end.

The design and the assurance of quality of the application were not the intern's objectives and members of the company responsible for these areas were responsible for carrying them out. All functional requirements were met, with the exception of two that had to be changed - the addition and removal of infrastructure from the map and the consequent assessment of the transport network. These requirements have been replaced by the suggestion of locations for bus stops requirement.

Chapter 7

Conclusion

Transport infrastructures determine the degree of accessibility of places and regions, thus contributing to the decision-making process for the location of companies and families. They will always choose regions and places with a greater degree of accessibility due to the possibility of transforming this accessibility into value [107]. That said, having a data visualization platform that a) detects patterns in the movement of the population, b) determines which transport modes the inhabitants have a preference for and c) suggests locations that may possibly require more transport infrastructure is an asset for planners of transport networks of a city. Despite the various features it brings, the great asset of the Motum web app is the Data Visualization strand that it makes available to its users.

In the development phase, several machine learning techniques were applied to develop algorithms capable of two tasks: detecting patterns in the movement of the population and classifying the means of transport that travel uses. In the clustering phase, given the computational requirement of the algorithms, a set of 100 thousand trips from the OpenPFLOW data set was used in order to try to understand what patterns exist in the movement of the population. In the classification phase, a set with 400 thousand trips of the OpenPFLOW artificial data set was used in order to understand the best algorithm and best configuration of hyperparameters to obtain the smallest possible macro F-measure. As for the suggestion of locations for new bus stops, Data Mining techniques were the only applied techniques and these were based in common sense. The suggestions can be of use, but this is not proven. These suggestions can benefit specific OpenPFLOW data set trips that can use the suggested stops. Then, the user can visually assess whether the stops make sense or not, from the map.

The clustering phase was initially intended only to group the population's movements in terms of origin and destination so that they could be represented on a map. This was extended to all features that were extracted for each trip. By looking at previous research and for the limits of values that the clustering metrics used can take, the results of the internal validation of the clustering carried out were good. Examples of this are the silhouette scores close to 1 obtained, Davies-Bouldin scores with very small values and very high Calinski-Harabasz scores; which follows what the literature suggests for these values. As for the interpretation of clusters through plots, it was possible to break them down further, although not as much as intended, given that the clustering algorithms when in the presence of several features, could not divide the data looking at all the features, but using one of them mainly.

After testing, the best algorithm for the clustering of trips was k-means, which obtained the following internal validation metrics for k=2, trip starting hour and trip duration being used: Silhouette of 0.895, Davies-Bouldin of 0.420 and Calinski-Harabasz of 156060. Then, an attempt to interpret the results in search of relevant patterns in each group of trips produced by the clustering was done. For this, more tests were done for the algorithm and hyperparameters with the best results from the first test phase. Not very successful, possibly due to the fact that the trips were found to be very compact in the multidimensional space of the features. In the classification of transport modes, the best result was obtained with a LGBM model - macro f-measure of 85.57%, combined with a precision of 86.35%, a recall of 84.95% and accuracy of 87.94%. However, these results were obtained using Tokyo cells as features, so the model may have learned about these cells. It was then important to perform tests without these features, so that models do not just learn about trips of one place in the world, and instead produce models for any type of trips data set. For that, the best results were obtained with a LGBM model - macro f-measure of 83.03%, combined with a precision of 84.02%, a recall of 82.32% and accuracy of 85.75%. In comparison to state-of-the-art studies, the results obtained were considered good, since for the same problem, few studies have been found that exceed the results of this work. Even so, there are studies in which this happens and because of that, improvements in terms of data set and ML process can be done to reach higher levels. For instance, hyperparameters of the ML algorithms can be increased so that better results are obtained or other trips data sets can be obtained to assess the generalization capability of the produced models.

Both in the development and in the validation phase, several knowledge was acquired in the area of machine learning, such as data analysis and processing, operation of the various models, techniques for optimizing hyperparameters, among others. The development methodology chosen allowed that the requirements could change and adapt to the customer's needs whenever necessary. Furthermore, it allowed that at the end of each sprint a functional product was presented to the customer in order to obtain feedback and rethink certain features. The risk analysis, which was carried out at the beginning of the second semester, allowed that important risks had still been identified and that the respective mitigation strategies had been thought out.

Informed decision-making through the maps and charts present at Motum was one of the objectives achieved and which still has a lot of potential and can be enhanced. The application area of these decisions can range from the simple extraction of patterns from large movements of the population to the planning of transport network infrastructure. The addition/removal of transport network infrastructures fell short of expectations, given that the suggestion being made may not be of much use. However, in the near future, this does not prevent further investigation of optimization or evolutionary algorithms so that such functionalities are implemented.

7.1 Difficulties overcome

In general, the difficulties overcome have to do with the investigation and learning of fields previously unknown to the intern. When this happened, a careful dive into literature that was peer reviewed was necessary, which was the case. Here are the main difficulties that were overcome during the project:

- For large data insertions in the website, the Lambda function that was being used was not being able to run for the desired time. In the second semester it was necessary to research technologies that would allow jobs to run without being in real-time. Fargate was chosen and, for that, Docker was also used as a container for the Python code that the job runs.

- It was the first time working with geographic coordinates. Firstly, Haversine distance was used to convert latitude and longitude into grid cells, but it was necessary to reverse them back into latitude and longitude, and haversine distance was a hard equation to handle. It was then decided to use distances relating the great circles of the Earth.
- Facing the problem of grouping trips in large movements of the population, a lot of research had to be done, since the intern had no experience in clustering. For each algorithm, there are different hyperparameters that were taken into account when defining the tests for clustering. The fact that the clustering results are not very conclusive may be due to this, but I believe that the possible was done with the data that the intern had. Many of the attempted Clustering tests were canceled due to the computer hardware used for them. These were interrupted due to problems with maximum memory exceeded. These happened in the calculation of the silhouette score and in the calculation of the most appropriate eps for DBSCAN, which are two computationally expensive processes.
- Contrarily, basics on testing supervised learning algorithms were already known to the intern. The use of pickle and Amazon Web Services to deploy these ML models to the could was something new. These ML models are already trained in S3 and the lambda functions will only fetch it for rapid use. In the case of clustering, whenever it is used in the web app, it is necessary to fit the data with the algorithm. This is a very time consuming process and it has been noticed that for a real-time feature it is not ideal. In future developments, another solution may have to be considered.
- Time spent researching network graphs due to the optimization of the transport network with addition/removal of infrastructure was the biggest setback but that enriched the work and the intern. It was concluded that the learning curve would be slow and would run away from the technologies used. In addition, it could take more time than was actually available.
- In the front end, various experimentations of maps to represent trips and locations were done. The Leaflet NGX framework has advantages over others, as it allowed heat maps and marker clusters and was available in TypeScript, which is great given that the intern already had knowledge of Angular.

To conclude, it was an interesting project in all its stages. All of them provided the learning of new knowledge in several areas, and it is even possible to say that it was the project in which the intern learned the most so far. It also provided the first contact with the business world and what it is like to work in an environment such as that of Grama, where all the elements, from the first day of the internship, were available to help with whatever was needed.

7.2 Future work

Given that this is a first version of Motum that could evolve into something much bigger, there are points to be improved in future work:

- Agnostic to the web application, during the trip detection phase in Tokyo's over-time location of people, more complex calculation features can be extracted, such as acceleration and percentage of the segment that is superimposed with a car/rail road. The second is more complex than the first and possibly requires geometric calculations with data about the terrain in question. Acceleration and Jerk are also extracted in another work [108].

- Straight lines are the trips representation for this first version, although curved lines are a better representation for mobility. For this, a library was later found, Canvas-Flowmap-Layer, which extends the ArcGIS API for JavaScript (Esri JSAPI) to map the flow of objects from an origin point to a destination point by using Bezier curves. In the Canvas-Flowmap-Layer github, the advantages of using Bezier curves over other curves to represent mobility are substantiated.
- In terms of transport infrastructures representation, there were struggles in representing them in the map. A file with 8673 bus stops was found for Tokyo in the Statistics Bureau of Japan. When represented on the map, the web application's response time increases dramatically, causing map crashes. Lighter solutions should be investigated later as all Tokyo infrastructures must be presented to the user.
- At the moment, techniques to analyze clusters 6.1 are not being used in Motum. In the future they may serve as support to clarify the user about the clustering produced in real time.
- In order to validate those clusters as actual predictors of human activity, these could be compared with clusters formed using ground truth variables, like presence of people, buildings, points of interest, bus and taxi movement [32], since the latter clusters can give information about points of interest that people visit most of the time.
- Further investigation of optimization problems related to transport networks and evolutionary algorithms may be done in the near future in order to implement the functionality of addition/removal of transport network infrastructures and network assessment.
- Another role may be necessary in the continuation of this project so that data can be consulted also using authentication. For this stage, it was not strictly necessary, so the intern did not implement it as he had no experience in role-based access control with AWS Cognito.

References

[1] P. P. US, "k-nearest neighbor." https://thenounproject.com/term/k-nearest-neighbor/2424488/.

[2] T. M. Mitchell, *Machine Learning.* USA: McGraw-Hill, Inc., 1 ed., 1997.

[3] A. Verikas, E. Vaiciukynas, A. Gelzinis, J. Parker, and M. Olsson, "Electromyographic patterns during golf swing: Activation sequence profiling and prediction of shot effectiveness," *Sensors*, vol. 16, no. 4, p. 592, 2016.

[4] MATLAB, "Artificial neural network image." https://www.mathworks.com/content/mathworks/www/en/discovery/deep-learning/jcr:content/mainParsys/band_2123350969_copy_1983242569/mainParsys/columns_1635259577/1/image_2128876021_cop_1731669336.adapt.full.high.svg/1569936707114.svg.

[5] S. Banerjee, "Recurrent neural network image." https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912.

[6] "Remix website." https://www.remix.com.

[7] "Optibus website." https://www.optibus.com/.

[8] T. Kashiyama, Y. Pang, and Y. Sekimoto, "Open pflow: Creation and evaluation of an open dataset for typical people mass movement in urban areas," *Transportation Research Part C: Emerging Technologies*, vol. 85, p. 249–267, 2017.

[9] "Image of a presigned url generation process." https://miro.medium.com/max/1530/1*K5yybcjSoP_jMZUlZ_n2lw.png.

[10] "Cold start latencies table with cloud providers comparison." https://kruschecompany.com/serverless-providers-comparison/, author=K&C Team.

[11] "Machine learning (ml) vs. ai and their important differences." https://medium.com/towards-artificial-intelligence/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6.

[12] "Nvidia - what's the difference between supervised, unsupervised, semi-supervised and reinforcement learning?." https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/#:~:text=In+a+supervised+learning+model,and+patterns+on+its+own.?ncid=afm-chs-44270&ranMID=44270&ranEAID=a1LgFw09t88&ranSiteID=a1LgFw09t88-9gesBoGobGg1dXh3I9MBOQ.

[13] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An introduction to statistical learning," *Springer Texts in Statistics*, 2013.

[14] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction.* The MIT Press, 2018.

[15] M. Kuhn and K. Johnson, *Applied Predictive Modeling.* Springer New York, 2013.

[16] P. Grover, "Gradient boosting from scratch." `https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d`.

[17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 3149–3157, Curran Associates Inc., 2017.

[18] "Lightgbm's documentation." `https://lightgbm.readthedocs.io/en/latest/`.

[19] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, p. 281–305, Feb. 2012.

[20] E. Duarte and J. Wainer, "Empirical comparison of cross-validation and internal metrics for tuning svm hyperparameters," *Pattern Recognition Letters*, vol. 88, p. 6–11, 2017.

[21] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," *Lecture Notes in Computer Science Learning and Intelligent Optimization*, p. 507–523, 2011.

[22] "Bayesian optimization article from arimo." `https://arimo.com/data-science/2016/bayesian-optimization-hyperparameter-tuning/`.

[23] M. Kraus, "Bayesian optimization article from vantage ai." `https://medium.com/vantageai/bringing-back-the-time-spent-on-hyperparameter-tuning-with-bayesian-optimisat`

[24] P.-N. T. Michael Steinbach and V. Kumar, "Cluster analysis: Basic concepts and algorithms," in *Introduction to Data Mining*, ch. 8, pp. 488–568, 2005.

[25] A. K. Jain and R. C. Dubes, *Algorithms for clustering data.* Prentice Hall, 1988.

[26] J. Han, "Cluster analysis: Basic concepts and methods," in *Data Mining: Concepts and Techniques*, ch. 10, pp. 444–495, 2000.

[27] "What is deep learning?." `https://machinelearningmastery.com/what-is-deep-learning/`.

[28] "Recurrent neural networks - técnico lisboa." `http://web.tecnico.ulisboa.pt/~andreas.wichert/12_RN.pdf`.

[29] S. Hochreiter and J. Schimdhuber, *Long short term memory*, vol. Neural Computation 9(8):1735-1780. Inst. fur Informatik, December 1997.

[30] G. D. Lorenzo, M. L. Sbodio, F. Calabrese, M. Berlingerio, R. Nair, and F. Pinelli, "Allaboard," *Proceedings of the 19th international conference on Intelligent User Interfaces - IUI 14*, 2014.

[31] D. Levinson and A. Kumar, "Multimodal trip distribution: Structure and application," *Transportation Research Record*, 1994.

[32] *Extracting urban activities through aggregate cellphone usage*, 2014.

[33] M. G. Demissie, S. Phithakkitnukoon, T. Sukhvibul, F. Antunes, R. Gomes, and C. Bento, "Inferring passenger travel demand to improve urban mobility in developing countries using cell phone data: A case study of senegal," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, p. 2466–2478, 2016.

[34] M. G. Demissie, F. Antunes, C. Bento, S. Phithakkitnukoon, and T. Sukhvibul, "Inferring origin-destination flows using mobile phone data: A case study of senegal," *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2016.

[35] E. Fabbiani, S. Nesmachnow, J. Toutouh, A. Tchernykh, A. Avetisyan, and G. Radchenko, "Analysis of mobility patterns for public transportation and bus stops relocation," *Programming and Computer Software*, vol. 44, no. 6, p. 508–525, 2018.

[36] M. K. E. Mahrsi, E. Come, L. Oukhellou, and M. Verleysen, "Clustering smart card data for urban mobility analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, p. 712–728, 2017.

[37] E. Thuillier, L. Moalic, S. Lamrous, and A. Caminada, "Clustering weekly patterns of human mobility through mobile phone data," *IEEE Transactions on Mobile Computing*, vol. 17, p. 817–830, Jan 2018.

[38] M. B. Rojas, E. Sadeghvaziri, and X. Jin, "Comprehensive review of travel behavior and mobility pattern studies that used mobile phone data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2563, no. 1, p. 71–79, 2016.

[39] K. Zhao, S. Tarkoma, S. Liu, and H. Vo, "Urban human mobility data mining: An overview," *2016 IEEE International Conference on Big Data (Big Data)*, 2016.

[40] T. H. N. O. R. S. Apichon Witayangkurn, Teerayut Horanont and R. Shibasaki, "Trip reconstruction and transportation mode extraction on low data rate gps data from mobile phone," July 2013.

[41] Q. Zhu, M. Zhu, M. Li, M. Fu, Z. Huang, Q. Gan, and Z. Zhou, *Identifying Transportation Modes from Raw GPS Data*, vol. 623, p. 395–409. Springer Singapore, 2016.

[42] A. T. A. Z. R. K. João Carreira, Pedro Fonseca, "A case for serverless machine learning,"

[43] C. Drumond, "Atlassian - scrum." `https://www.atlassian.com/agile/scrum`.

[44] *IEEE standard glossary of software engineering terminology: IEEE std 610.12-1990.* Institute of Electrical and Electronics Engineers, 1990.

[45] "Non-functional requirements: Examples, types, how to approach." `https://www.altexsoft.com/blog/non-functional-requirements/`.

[46] "Usability 101: Introduction to usability - nielsen norman group." `https://www.nngroup.com/articles/usability-101-introduction-to-usability/`.

[47] "Owasp top ten." `https://owasp.org/www-project-top-ten/`.

[48] "Response times: The 3 important limits - nielsen norma group." `https://www.nngroup.com/articles/response-times-3-important-limits/`.

[49] "Measuring web performance." https://www.keycdn.com/blog/measuring-web-performance.

[50] "The c4 model for visualising software architecture." https://c4model.com/.

[51] M. von Mörner, "Application of call detail records - chances and obstacles," *Transportation Research Procedia*, vol. 25, pp. 2233 – 2241, 2017. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.

[52] Y. S. akehiro Kashiyama, Yanbo Pang, "Openpflow github." https://github.com/sekilab/OpenPFLOW.

[53] "Introduction to postgis - geography." https://postgis.net/workshops/postgis-intro/geography.html.

[54] A. Ali, J. Kim, and S. Lee, "Travel behavior analysis using smart card data," *KSCE Journal of Civil Engineering*, vol. 20, 07 2015.

[55] "Distance calculation between geographic coordinates." https://www.mkompf.com/gps/distcalc.html.

[56] D. S. Birney, G. Gonzalez, and D. Oesper, *Observational astronomy.* Cambridge Univ. Press, 2010.

[57] "Github with distance formulas comparison." https://github.com/skaringa/DistanceCalculation/blob/master/Sample.cpp.

[58] G. Cich, L. Knapen, T. Bellemans, D. Janssens, and G. Wets, "Trip/stop detection in gps traces to feed prompted recall survey," *Procedia Computer Science*, vol. 52, p. 262–269, 2015.

[59] "Understanding q-q plots." https://data.library.virginia.edu/understanding-q-q-plots/#:~:text=A%20Q%2DQ%20plot%20is%20a,truly%20come%20from%20Normal%20distributions..

[60] J. W. Tukey, *Exploratory data analysis.* Pearson, 1970.

[61] S. Seo, "A review and comparison of methods for detecting outliers in univariate data sets," Master's thesis, University of Pittsburgh, 2006.

[62] E. C. Blessie and E. Karthikeyan, "Sigmis: A feature selection algorithm using correlation based method," *Journal of Algorithms Computational Technology*, vol. 6, no. 3, p. 385–394, 2012.

[63] "Point-biserial correlation coefficient." https://en.wikipedia.org/wiki/Point-biserial_correlation_coefficient.

[64] A. Richardson, "Nonparametric statistics for non-statisticians: A step-by-step approach by gregory w. corder, dale i. foreman," *International Statistical Review*, vol. 78, no. 3, p. 451–452, 2010.

[65] "Google developers - k-means advantages and disadvantages." https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages.

[66] "University of uppsala - advantages disadvantages of k-means and hierarchical clustering." http://santini.se/teaching/ml/2016/Lect_10/10c_UnsupervisedMethods.pdf.

[67] P. H. Ahmad and S. Dang, "Performance evaluation of clustering algorithm using different datasets," *Journal of Information Engineering and Applications*, vol. 5, pp. 39–46, 2015.

[68] B. Borah and D. K. Bhattacharyya, "An improved sampling-based dbscan for large spatial databases," pp. 92 – 96, 02 2004.

[69] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, p. 283–304, 1998.

[70] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," 1997.

[71] M. R. Rezaee, B. Lelieveldt, and J. Reiber, "A new cluster validity index for the fuzzy c-mean," *Pattern Recognition Letters*, vol. 19, no. 3-4, p. 237–246, 1998.

[72] "How to use dbscan effectively." `https://towardsdatascience.com/how-to-use-dbscan-effectively-ed212c02e62`.

[73] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," *Ecological Modelling*, vol. 406, p. 109–120, 2019.

[74] J. Hagenauer and M. Helbich, "A comparative study of machine learning classifiers for modeling travel mode choice," *Expert Systems with Applications*, vol. 78, 02 2017.

[75] K. Markham, "Comparing supervised learning algorithms." `https://www.dataschool.io/comparing-supervised-learning-algorithms/`.

[76] H. Narasimhan, W. Pan, P. Kar, P. Protopapas, and H. G. Ramaswamy, "Optimizing the multiclass f-measure via biconcave programming," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1101–1106, 2016.

[77] D. Hand and P. Christen, "A note on using the f-measure for evaluating record linkage algorithms," *Statistics and Computing*, vol. 28, no. 3, p. 539–547, 2017.

[78] T. Shanmukhappa, I. W.-H. Ho, and C. K. Tse, "Spatial analysis of bus transport networks using network theory," *Physica A: Statistical Mechanics and its Applications*, vol. 502, p. 295–314, 2018.

[79] C. Caminha, V. Furtado, V. Pinheiro, and C. Ponte, "Graph mining for the detection of overcrowding and waste of resources in public transport," *Journal of Internet Services and Applications*, vol. 9, no. 1, 2018.

[80] R. Żochowska and P. Soczówka, "Analysis of selected transportation network structures based on graph measures," *Scientific Journal of Silesian University of Technology. Series Transport*, vol. 98, p. 223–233, 2018.

[81] H. Zhang, "Structural analysis of bus networks using indicators of graph theory and complex network theory," *The Open Civil Engineering Journal*, vol. 11, no. 1, p. 92–100, 2017.

[82] T. Shanmukhappa, I. W.-H. Ho, C. K. Tse, and K. K. Leung, "Recent development in public transport network analysis from the complex network perspective," *IEEE Circuits and Systems Magazine*, vol. 19, no. 4, p. 39–65, 2019.

[83] O. Kuz'Kin, "Graph theory methods in analysing commuting networks of municipal electric transport," *Eastern-European Journal of Enterprise Technologies*, vol. 2, no. 4(80), p. 19, 2016.

[84] G.-L. Jia, R.-G. Ma, and Z.-H. Hu, "Urban transit network properties evaluation and optimization based on complex network theory," *Sustainability*, vol. 11, no. 7, p. 2007, 2019.

[85] J. Hong, R. Tamakloe, S. Lee, and D. Park, "Exploring the topological characteristics of complex public transportation networks: Focus on variations in both single and integrated systems in the seoul metropolitan area," *Sustainability*, vol. 11, no. 19, p. 5404, 2019.

[86] S. Guze, "Graph theory approach to the vulnerability of transportation networks," *Algorithms*, vol. 12, no. 12, p. 270, 2019.

[87] "Ngx-leaflet package for angular." `https://github.com/Asymmetrik/ngx-leaflet`.

[88] "Mapbox studio." `https://studio.mapbox.com/`.

[89] A. Sadilek and X. Dotiwalla, "New insights into human mobility with privacy preserving aggregation." `https://ai.googleblog.com/2019/11/new-insights-into-human-mobility-with.html`, November 2019.

[90] B. Jenny, D. M. Stephen, I. Muehlenhaus, B. E. Marston, R. Sharma, E. Zhang, and H. Jenny, "Design principles for origin-destination flow maps," *Cartography and Geographic Information Science*, vol. 45, no. 1, p. 62–75, 2016.

[91] Y. Yang, T. Dwyer, B. Jenny, K. Marriott, M. Cordeil, and H. Chen, "Origin-destination flow maps in immersive environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, p. 693–703, 2019.

[92] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2/3, p. 107–145, 2001.

[93] G. S. Linoff and M. J. A. Berry, *Data mining techniques for marketing, sales, and customer relationship management.* Wiley Pub., Inc., 2011.

[94] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, p. 243–256, 2013.

[95] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, p. 53–65, 1987.

[96] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[97] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics - Theory and Methods*, vol. 3, no. 1, p. 1–27, 1974.

[98] "Scikit-learn kmeans." `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`.

[99] N. de Vos, "Kprototypes python implementation." `https://github.com/nicodv/kmodes/blob/master/kmodes/kprototypes.py`.

[100] J. Hämäläinen, S. Jauhiainen, and T. Kärkkäinen, "Comparison of internal clustering validation indices for prototype-based clustering," *Algorithms*, vol. 10, no. 3, p. 105, 2017.

[101] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, pp. 166–171, 2011.

[102] "Python kneed library." `https://pypi.org/project/kneed/`.

[103] S. S. De, M. Mishra, and S. Dehuri, "Mvclustviz," *International Journal of System Dynamics Applications*, vol. 2, no. 4, p. 19–32, 2013.

[104] B. D. Ripley, "Pattern recognition and neural networks," 1996.

[105] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.* USA: Prentice Hall Press, 3rd ed., 2009.

[106] "Hyperopt." `http://hyperopt.github.io/hyperopt/`.

[107] C. Ferrari, A. Bottasso, M. Conti, and A. Tei, *Economic role of transport infrastructure: theory and models.* Elsevier, 2019.

[108] S. Dabiri and K. Heaslip, "Inferring transportation modes from gps trajectories using a convolutional neural network," *Transportation Research Part C: Emerging Technologies*, vol. 86, p. 360–371, 2018.

Appendices

Functional Requirements

This type of requirements specify which behaviors the system should have.

User Login: REQ01

This feature allows users to login into their account.

Dependencies	Configured Amazon Cognito
Preconditions	User has an account registered in the system User is in the login page
Success Scenario	1. Insert an e-mail 2. Insert the correct password
Success Outcome	User is redirected to the homepage
Possible Failure Outcomes	1. The message "Invalid e-mail or password" is presented 2. The message "E-mail is mandatory" 3. The message "Password is mandatory"
Priority	Must Have

Table 1: Login of an already registered user in the Web Application: REQ 01.

User Logout: REQ02

This feature allows users to logout of their account.

Dependencies	Configured Amazon Cognito
Preconditions	User has an account registered in the system User is logged in
Success Scenario	1. User clicks the logout button
Success Outcome	User is redirected to the login page
Possible Failure Outcomes	None
Priority	Must Have

Table 2: Perform Logout of the Web Application: REQ 02.

Reset Password: REQ03

This feature allows users to reset their account's password in case they forget it.

Dependencies	Configured Amazon Cognito
Preconditions	User has an account registered in the system User is in the login page
Success Scenario	1. Click the reset password button 2. Insert a valid and existent email
Success Outcome	User is redirected to a page with the message: "message was sent to your email with a new password"
Possible Failure Outcomes	1. The message "Invalid or non-existent email" is presented
Priority	Must Have

Table 3: Reset a Web Application account's password: REQ 03.

Change Password: REQ 04

This feature allows users to change the password of their account.

Dependencies	Configured Amazon Cognito
Preconditions	User has an account registered in the system User is logged in User is in the My Account page
Success Scenario	1. Click the change password button 2. Insert the current password twice 3. Insert new password
Success Outcome	User is redirected to the Home Page
Possible Failure Outcomes	1. The message "The current password is wrong" is presented 2. The message "The new password is invalid" is presented
Priority	Must Have

Table 4: Change the password of a Web Application's account: REQ 04.

Change users personal settings: REQ 05

This feature allows users to change their personal settings.

Dependencies	Configured Amazon Cognito
Preconditions	User has an account registered in the system User is logged in User is in the My Account page
Success Scenario	1. Click to change a certain attribute 2. Insert the attribute's new value 3. Click to save
Success Outcome	User personal settings are updated
Possible Failure Outcomes	1. The message "The new attribute value is invalid" 2. The message "Changes to that attribute are not allowed"
Priority	Must Have

Table 5: Change personal settings of a Web Application's account: REQ 05.

Load historical data about Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations and Airports: REQ 06

This feature allows users to load transport infrastructures data.

Dependencies	User has an account registered in the system as a Data Provider
Preconditions	Data Provider is logged in
Success Scenario	1. Insert an appropriate dataset file to upload This data must include at least the following data points: . Location of the station, taxi line or airport . When it started operating
Success Outcome	The system can now display that historical data on screen and feed it to machine learning algorithms to improve predictions. An authenticated data consumer can now access this data in the website.
Possible Failure Outcomes	1. The message "Invalid data format." is presented
Priority	Must Have

Table 6: Load historical data about several points of interest: REQ 06.

Load historical data about spatiotemporal positions of individuals and trips: REQ 07

This feature allows users to load spatiotemporal positions of individuals and trips data.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider is logged in
Success Scenario	1. Insert an appropriate spatiotemporal positions of individuals and trips file to upload This data must include at least the following data points: . Timestamp . Longitude . Latitude . Agent ID . Transport
Success Outcome	The system can now display that historical data on screen and feed it to machine learning algorithms to improve predictions. A authenticated data consumer can now access this data in the website.
Possible Failure Outcomes	1. The message "Invalid data format." is presented
Priority	Must Have

Table 7: Load historical data about spatiotemporal positions of individuals and trips: REQ 07.

Store data in internal data structures: REQ 08

This feature allows users to store all data in reliable data repositories.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	A Data Provider is logged in This Data Provider has uploaded a file containing adequate data
Success Scenario	1. The intelligent system successfully stores the uploaded data in data repositories that are reliable and allow quick access to its consumers.
Success Outcome	The system can now display that historical data on screen and feed it to machine learning algorithms to improve predictions. An authenticated data consumer can now access this data in the website.
Possible Failure Outcomes	1. The message "Invalid data format." is presented
Priority	Must Have

Table 8: Store data in internal data structures that are fit for purpose: REQ 08.

Removing noise in the input data: REQ 09

This feature allows users to remove all noise that input data can possibly have.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider is logged in This Data Provider has uploaded a file containing adequate data
Success Scenario	1. The intelligent system cleans up values in the input data that are irrelevant for the machine learning algorithms.
Success Outcome	The system can now make more accurate predictions due to a prior Data Cleaning
Possible Failure Outcomes	None
Priority	Must Have

Table 9: Noise removal on input data: REQ 09.

Input Data Normalization: REQ 10

This functionality allows the system to normalize input data. This is something that is not needed if data always comes in the same format and same features have the same measurements.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider is logged in This Data Provider has uploaded a file containing adequate data
Success Scenario	1. The intelligent system should correct of each feature values to the same measurement and type. This normalization also includes cases where data needs to be categorized.
Success Outcome	The system can now make more accurate predictions due to a prior Data Normalization
Possible Failure Outcomes	None
Priority	Must Have

Table 10: Normalizing input data: REQ 10.

Input Data Formats: REQ 11

This functionality allows the system to receive uploads of various input data formats.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider is logged in This Data Provider has uploaded a file containing adequate data
Success Scenario	1. The intelligent system allows input data to be from one of 4 types: CSV, XLS (XSLX), JSON or XML.
Success Outcome	Data is easily loaded to the system.
Possible Failure Outcomes	1. The message "Invalid data type." is presented.
Priority	Nice to have

Table 11: Allowing various input data formats: REQ 11.

Group and represent large population movements: REQ 12

This feature allows the system to cluster population trips and produce a representation of it to show in the web application.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Consumer The system has all the necessary historical data
Preconditions	Data Consumer is logged in
Success Scenario	1. User selects how many groups he wants generated by clustering. 2. User specifies which features the clustering will use.
Success Outcome	The system generates all the needed information to create map data representation of groups of large population movements.
Possible Failure Outcomes	Not defined
Priority	Must Have

Table 12: The system generates map data representations of groups of large population movements as well as some statistics about them: REQ 12.

Classification of most likely used transport infrastructures in population trips: REQ 13

This feature allows the system to classify trips by the transport modes most likely used.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Consumer The system has all the necessary historical data
Preconditions	Data Consumer is logged in
Success Scenario	1. By using a previously trained machine learning model, the system assigns most likely used transport modes to trips.
Success Outcome	System infers which transport infrastructure was most likely used in the population movements.
Possible Failure Outcomes	Not defined
Priority	Must Have

Table 13: Estimation of most likely used transport modes in population trips: REQ 13.

Suggestion of new transport infrastructures by analyzing population movements: REQ 14

This functionality provides the system with the capacity of suggesting new transport infrastructures locations.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Consumer The system has all the necessary historical data
Preconditions	Data Consumer is logged in
Success Scenario	1. User selects how many suggestions he wants the system to produce. 2. User also selects if the suggestion should be done by taking rush hours into account.
Success Outcome	System generates suggestions data to present it in a map.
Possible Failure Outcomes	Not defined
Priority	Must Have

Table 14: Influence of hypothetical creation of new transport infrastructures on population movements: REQ 14.

Read historical data about Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations and Airports: REQ 15

This feature allows users to consult data about transport infrastructures.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider has logged in
Success Scenario	1. The data provider is in the desired page and chooses to read the desired data.
Success Outcome	The desired data is obtained.
Possible Failure Outcomes	1. The message "There's no data in the system" is presented.
Priority	Must Have

Table 15: Read data about all loaded points of interest: REQ 15.

Update historical data about Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations and Airports: REQ 16

This feature allows users to update data about transport infrastructures.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider has logged in.
Success Scenario	1. The data provider is in the desired page and chooses to update the desired data.
Success Outcome	Data is updated in the database.
Possible Failure Outcomes	1. The message "Data couldn't be updated." is presented. 2. The message "Bad format." is presented.
Priority	Must Have

Table 16: Update already loaded points of interest data: REQ 16.

Delete historical data about Train Stations, Bus Stations, Subway Stations, Taxi Lines, Ferry Docking Stations and Airports: REQ 17

This feature allows users to delete data about transport infrastructures.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider has logged in
Success Scenario	1. The data provider is in the desired page and chooses to delete the desired data.
Success Outcome	Data is deleted from the database.
Possible Failure Outcomes	1. The message "Data couldn't be deleted." is presented.
Priority	Must Have

Table 17: Delete loaded points of interest data: REQ 17.

Read historical data about spatiotemporal positions of individuals and trips: REQ 18

This feature allows users to consult data about spatiotemporal positions of individuals and trips.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider is logged in.
Success Scenario	1. The data provider is in the desired page and chooses to consult the data.
Success Outcome	The desired data is obtained.
Possible Failure Outcomes	1. The message "There's no data in the system" is presented.
Priority	Must Have

Table 18: Read data about loaded spatiotemporal positions of individuals and trips: REQ 18.

Update historical data about spatiotemporal positions of individuals and trips: REQ 19

This feature allows users to consult data about spatiotemporal positions of individuals and trips.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider has logged in.
Success Scenario	1. The data provider is in the desired page and chooses to update data.
Success Outcome	Data is updated in the database.
Possible Failure Outcomes	1. The message "Data couldn't be updated." is presented. 2. The message "Bad format." is presented.
Priority	Must Have

Table 19: Update already loaded spatiotemporal positions of individuals and trips data: REQ 19.

Delete historical data about spatiotemporal positions of individuals and trips: REQ 20

This feature allows users to delete data about spatiotemporal positions of individuals and trips.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Provider
Preconditions	Data Provider has logged in.
Success Scenario	1. The data provider is in the desired page and chooses to delete data.
Success Outcome	Data is deleted from the database.
Possible Failure Outcomes	1. The message "Data couldn't be deleted." is presented.
Priority	Must Have

Table 20: Delete loaded spatiotemporal positions of individuals and trips: REQ 20.

Display historical data in a map: REQ 21

This feature allows users to visualize data about spatiotemporal positions of individuals and trips in a map.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Consumer The system has all the necessary historical data
Preconditions	None
Success Scenario	1. From historical data, the system displays a map of population trips, locations over time and transport infrastructures
Success Outcome	An authenticated data consumer can easily and visually analyze large population movements, locations over time and transport infrastructures
Possible Failure Outcomes	Not defined
Priority	Must Have

Table 21: Display historical data in a map: REQ 21.

Display groups of population movement: REQ 22

This feature allows users to visualize groups of trips in a map.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Consumer The system has all the necessary historical data
Preconditions	None
Success Scenario	1. From historical movement patterns extraction, the system uses a clustering algorithm to assign a cluster to which trip 2. Trips with the same origin and destination are represented in a map. Each trip line has a pop-up click which shows information such as the cluster that trip belongs to
Success Outcome	An authenticated data consumer can see trips between the same origin and destination and from their features, draw conclusions about which ones influence cluster attribution
Possible Failure Outcomes	Not defined
Priority	Must Have

Table 22: Display groups of population movement in a map: REQ 22.

Display transport modes that most likely were used in population trips: REQ 23

Display transport mode classifications of all the trips. Performance metric for each transport mode is also showed.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Consumer The system has all the necessary historical data
Preconditions	None
Success Scenario	1. From historical movement patterns extraction, the system uses a previously trained machine learning model to calculate which transport infrastructure was most likely used in the population trips 2. Trips with the same origin and destination are represented in a map. Each trip line has a pop-up click which shows information such as transport mode predictions
Success Outcome	An authenticated data consumer can know which transport mode was most likely used in population trips and therefore draw conclusions about which transport modes influence large population movements
Possible Failure Outcomes	Not defined
Priority	Must Have

Table 23: Display transport modes that most likely were used in population trips: REQ 23.

Display suggestions of new transport infrastructures: REQ 24

Display location suggestions of new transport infrastructures in a map.

Dependencies	Configured Amazon Cognito User has an account registered in the system as a Data Consumer The system has all the necessary historical data
Preconditions	None
Success Scenario	1. From historical population trips new locations for transport infrastructures are calculated for the suggestion 2. Location for the suggested infrastructures are represented in a map. 3. Hypothetical statistics are shown for before and after the suggested infrastructures are used in historical trips
Success Outcome	An authenticated data consumer can consult these suggestion and decide if the new location is acceptable by analyzing what is represented on the map
Possible Failure Outcomes	Not defined
Priority	Must Have

Table 24: Display suggestions of new transport infrastructures: REQ 24.