André Miguel Pereira Pinho

Inteligência no Negócio para aplicação em Telecomunicações

Relatório de Estágio
Mestrado em Engenharia de Software
orientada por Pedro Furtado, Helena Margarida, Ricardo Ângelo
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Julho de 2018

· U C ·

Universidade de Coimbra

Mestrado em Engenharia Informática
Estágio
Relatório Final

Inteligência no Negócio para aplicação em Telecomunicações

André Miguel Pereira Pinho
apinho@student.dei.uc.pt

Orientadores:
Prof. Doutor Pedro Furtado (DEI)
Eng. Helena Margarida (Altice Labs)
Eng. Ricardo Ângelo (Altice Labs)

Data: 2 de Julho de 2018

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

Hoje, as telecomunicações fazem parte da vida das pessoas, empresas e organizações. A conectividade entre os operadores de telecomunicações e os clientes gera uma enorme quantidade de dados. Esta informação gerada e armazenada regularmente é de grande importância, pois pode revelar conhecimento importante quando analisada com técnicas avançadas de análise de dados. O principal objetivo deste trabalho é o estudo e desenvolvimento de uma plataforma com modelos preditivos e descritivos neste contexto. A plataforma consiste num back-end fornecido na forma de uma API, que recolhe e analisa dados existentes em outros produtos com o objetivo de extrair conhecimento. Usando diversas técnicas de data mining, a plataforma desenvolvida é capaz de determinar o perfil de clientes aderentes e não aderentes a uma campanha de marketing, prever vários tipos de consumo e descobrir relações de market basket nos serviços subscritos pelos clientes. O conhecimento extraído pelas funcionalidades fornece uma excelente ferramenta no apoio à tomada de decisão e avaliação de resultados. Com o propósito de validar os objetivos e requisitos propostos, dados reais de um operador de telecomunicações foram utilizados neste trabalho. O trabalho em si, resultou uma plataforma com um conjunto de funcionalidades que foram desenvolvidas para ajudar os gestores a analisar os dados. Inclui opções de configuração e parametrização desenvolvidas para integração nos produtos da empresa.

Palavras-Chave: Análise de dados, Clustering, Data mining, Regras de associação, Séries temporais, Telecomunicações

Abstract

Today, telecommunications are part of the lives of people, companies and organizations. The connectivity between Telecom operators and customers generates a huge amount of data. This information generated and stored regularly is of great importance, as it can reveal important knowledge when analyzed with advanced data analysis techniques. The main objective of this work is the study and development of a platform with predictive and descriptive models in that context. The platform consists of a back-end provided in the form of an API, which retrieves and analyzes existing data in other products with the purpose of extracting knowledge. Using several data mining techniques, the platform developed is able to determine the profile of adherent and non-adherent customers to a marketing campaign, predict various types of consumption, and discover market basket relationships in the subscribed services by the customers. The knowledge extracted by the functionalities provides an excellent tool in support of decision making and evaluation of results. For the purpose of validating the proposed objectives and requirements, real data from a Telecom operator was used in this work. The work itself resulted in a platform with a set of functionalities that were developed to help managers analyze the data. It includes configuration and parameterization choices and it was developed for further integration in the products of the company.

Keywords: Clustering, Data analysis, Data mining, Association rules, Time series, Telecommunications

Reconhecimentos

É de enorme justiça reconhecer publicamente e agradecer à instituição e pessoas que contribuíram para que fosse possível concluir este estágio.
Começo por agradecer à minha família pelo incentivo e apoio, principalmente nos momentos de maior dificuldade.
Ao Professor, Pedro Furtado, pela ajuda, apoio, sabedoria, inteligência e empenho que colocou em todas as fases deste projeto de estágio. Aos Orientadores da empresa pelo acompanhamento e validação do trabalho.
À Altice Labs pela oportunidade de realizar o estágio em ambiente empresarial e forma como me acolheu e integrou numa equipa. Pelas excelentes condições, recursos disponibilizados e flexibilidade para dar respostas às solicitações.
Finalmente aos amigos em especial o excelente grupo de estagiários da empresa.

Índice de Conteúdos

Lista de Figuras

Lista de Tabelas

Glossário

ACF: Autocorrelation Function
ACM: Active Campaign Manager
AR: Auto-Regressive
ARIMA: Auto-Regressive Integrated Moving Average
ARPU: Average Revenue Per User
API: Application Programming Interface
BI: Business Intelligence
BIT: Business Intelligence Tools
CRISP-DM: Cross-Industry Standard Process for Data Mining
DEI: Departamento de Engenharia Informática
ETL: Extract, Transform, Load
FP: Frequent Pattern
GC: Grupo de Controlo
HDD: Hard Disk Drive
HTTP: Hypertext Transfer Protocol
IDE: Integrated Development Environment
JSON: JavaScript Object Notation
KPI: Key Performance Indicator
MA: Moving Average
MAPE: Mean Absolute Percentage Error
RMSE: Root Mean Squared Error
NGIN PCC: Next Generation Intelligent Network (Policy, Charging and Control)
OLAP: Online Analytical Processing
ORM: Origem na Rede Móvel
PA: Público-Alvo
PACF: Partial Autocorrelation Function
RAM: Random Access Memory
RBF: Radial Basis Function
REST: Representational State Transfer
RF: Rede Fixa
SMS: Short Message Service
SSE: Sum of Squared Error
SVM: Support Vector Machines
WEKA: Waikato Environment for Knowledge Analysis

Churn: desistência de serviço por parte de clientes
Clustering: agrupamento de elementos descritos por um conjunto de atributos
Cluster: grupo de elementos semelhantes após a aplicação da técnica de clustering
Grupo de controlo: grupo de clientes em condições de entrar numa campanha, selecionados aleatoriamente para não serem incentivados
Lift over grupo de controlo: métrica que mede e avalia o sucesso da campanha, relacionando a taxa de adesão dos clientes do público-alvo (incentivados por notificação) com a taxa de adesão dos clientes do grupo de controlo (não incentivados)
Market basket: processo que analisa os itens comprados em conjunto pelos clientes
Waterfall: metodologia tradicional de desenvolvimento de software em cascata
Testes black-box: teste das funcionalidades de um software em função do input e output
Testes white-box: teste da estrutura interna de um software

Capítulo 1

Introdução

O presente relatório de estágio descreve o trabalho desenvolvido pelo aluno André Miguel Pereira Pinho, no âmbito da unidade curricular de Estágio do Mestrado em Engenharia de Software, na Universidade de Coimbra. O estágio teve lugar na Altice Labs, empresa com sede em Aveiro, especializada no desenvolvimento de tecnologia para os operadores de telecomunicações em vários pontos do mundo.

1.1. Enquadramento

Vivemos no mundo das tecnologias de informação onde os serviços de telecomunicações desempenham um papel fundamental na vida das pessoas, empresas e organizações. Hoje, sem a capacidade de comunicar pelos meios tecnológicos, a vida das pessoas é impensável e nas empresas seria o caos completo. Podemos considerar que estamos em presença de uma nova era digital nas comunicações móveis, com impactos significativos para os utilizadores e operadores.

Esta nova era digital proporciona por um lado, a criação de novos negócios, por outro lado o fim de outros que perdem preponderância. Neste contexto, os conteúdos de streaming e as aplicações de comunicação por internet assumem grande importância. Estas últimas representam 80% do tráfego de mensagens e mais de um terço dos minutos de tráfego de voz internacional [1].

Esta nova realidade nas telecomunicações, obriga as empresas a possuir e criar ferramentas de suporte à decisão, que disponibilizem indicadores e informação em tempo real, por um lado, das tendências e necessidades do mercado, por outro lado, dos recursos de rede disponíveis. Uma oportunidade relevante é analisar a enorme e valiosa quantidade de dados gerada e armazenada regularmente com recurso às tecnologias atuais de análise de dados. Este projeto vem na sequência desta nova aposta da empresa e enquadra-se na área de inteligência de negócio.

1.2. Ecossistema NGIN PCC da Altice Labs

Neste contexto a Altice Labs possui o ecossistema Next Generation Intelligent Network - Policy, Charging and Control (NGIN PCC), solução de controlo de tráfego e cobrança em tempo real para clientes do operador de telecomunicações. Tem como objetivo dar resposta a estes novos desafios, ao nível do aumento da procura e consumo de serviços de dados, que são colocados aos operadores de telecomunicações. Constituído por um conjunto de aplicações integradas, que permitem aos operadores disponibilizar e gerir novos serviços de uma forma rápida, económica e que satisfaça as expectativas dos clientes. Dentro deste existem duas aplicações que serão as fontes de dados deste projeto, o Business Intelligence Tools (BIT) e o Active Campaign Manager (ACM).

Os BIT são uma framework de inteligência de negócio, que realiza o processo de recolha, tratamento, organização e análise da informação de suporte à gestão do negócio. Reporta e

fornece indicadores operacionais e de negócio, do que está a acontecer no momento na atividade de negócio, nas áreas de informação de clientes, tráfego e transações.
O ACM é uma plataforma para criar, lançar, executar e monitorizar campanhas e promoções. Permite aos operadores desenhar e lançar campanhas, promoções em tempo real direcionadas aos seus clientes, contribuindo para o aumento da receita do negócio e da satisfação do cliente. Fornece uma interface intuitiva de desenho do ciclo de vida da campanha, um sistema de execução e gestão da campanha flexível e ainda um conjunto de relatórios que permitem acompanhar os seus resultados em tempo real. Um projeto em desenvolvimento neste produto importante de ser mencionado, consiste na identificação do público-alvo a uma campanha com recurso a classificadores.

1.3. Problema a resolver

As novas funcionalidades a desenvolver neste projeto pretendem contribuir para a resolução de um conjunto de problemas. O primeiro problema resulta da forte concorrência, que provoca a migração dos clientes e a quebra nos lucros. Em segundo nas atividades de marketing, existe uma taxa de adesão de cerca de 5%, originando um desperdício de oportunidades de negócio e recursos financeiros. Finalmente, ocorrem dificuldades relacionadas com a gestão de rede, sobrecarga ou desperdício de recursos.

1.4. Objetivo

O estágio tem como finalidade o estudo e desenvolvimento de uma solução de suporte à decisão para alcançar os seguintes três objetivos:

- Fornecer insights durante as campanhas para identificar com maior precisão o seu público-alvo;
- Prever consumos para ajudar no planeamento e gestão de recursos de rede, e fornecer uma melhoria na qualidade de serviço;
- Extrair relações entre serviços subscritos em simultâneo pelos clientes.

A solução a desenvolver será uma plataforma com modelos preditivos e descritivos. Esta consiste num back-end que recolhe e analisa dados provenientes das fontes de dados existentes, com o propósito de extrair conhecimento. O conhecimento será disponibilizado por uma API em formatos de dados que possam ser reconhecidos por outros produtos. Tendo em conta a recente aposta da empresa nesta área de data mining, este projeto consiste numa prova de conceito. Baseia-se em abordagens de referências internacionais que apoiam as funcionalidades desenvolvidas.

O trabalho divide-se nas seguintes etapas:

- Estudo dos conceitos fundamentais e análise do Estado da Arte relacionado ao tema do trabalho;
- Levantamento e especificação dos requisitos da plataforma a desenvolver;
- Desenho da arquitetura da solução que suporte todos os requisitos especificados;
- Desenvolvimento dos módulos da plataforma;
- Experimentação e desenvolvimento de modelos dos requisitos;
- Testes de validação à plataforma desenvolvida.

1.5. Contribuições

Deste trabalho resultou um artigo, Experimental Comparison and Tuning of Time Series Prediction for Telecom Analysis, submetido na International Conference on Time Series and Forecasting (ITISE 2018) a realizar a 19-21 de setembro de 2018, em Granada, Espanha. O artigo encontra-se no anexo C deste relatório.

1.6. Estrutura do relatório

O presente relatório está dividido em nove capítulos. O capítulo 2 de Gestão de Projeto, descreve as metodologias de desenvolvimento e ferramentas usadas durante o projeto de estágio. Segue-se a apresentação do planeamento das tarefas dos dois semestres e os desvios ocorridos ao longo do estágio. Por fim apresenta a análise de riscos relacionados com este projeto. O capítulo 3 de Conhecimento Base, expõe o estudo dos conceitos gerais de telecomunicações e de data mining importantes ao desenvolvimento deste projeto. O capítulo 4 de Estado da Arte, apresenta o levantamento e análise dos sistemas de suporte à decisão concorrentes, e o estudo de abordagens de implementação dos requisitos funcionais. Por fim apresenta o levantamento e comparação de ferramentas de data mining. O capítulo 5 de Especificação de Requisitos, apresenta a plataforma a desenvolver, os seus requisitos funcionais, não funcionais, e suas restrições técnicas e de negócio. O capítulo 6 de Arquitetura, divide-se em desenho e especificação detalhada da arquitetura proposta para a plataforma. O desenho descreve a arquitetura da plataforma com um conjunto de vistas relevantes, que demostram como os requisitos deste projeto são suportados. A especificação detalhada descreve os detalhes técnicos dos módulos da plataforma. O capítulo 7 de Experimentação e Desenvolvimento de Modelos, apresenta o estudo, análise comparativa e avaliação de modelos dos requisitos funcionais. O capítulo 8 de Testes, expõe o plano e os resultados de execução dos testes de aceitação, requisitos funcionais e não funcionais. Finalmente o capítulo 9 de Conclusão, apresenta as conclusões do trabalho realizado, sugestões para trabalho futuro e o balanço do estágio.

Capítulo 2

Gestão do Projeto

Este capítulo começa por descrever as metodologias de desenvolvimento e as ferramentas de gestão de projeto selecionadas e usadas ao longo do projeto de estágio. De seguida apresenta o planeamento para o primeiro e segundo semestre. Finalmente a última secção expõe a análise de riscos deste projeto.

2.1. Metodologia de desenvolvimento

O desenvolvimento deste projeto seguiu uma metodologia baseada em Waterfall (modelo em cascata) [2], constituída por uma sequência linear de fases, em que uma fase apenas começa quando a anterior for concluída. A escolha desta metodologia, deveu-se à natureza das tarefas seguirem uma ordem sequencial de execução e reduzida probabilidade dos requisitos sofrerem modificações. A Figura 1 ilustra a adaptação desta metodologia ao desenvolvimento deste projeto com as seguintes fases:

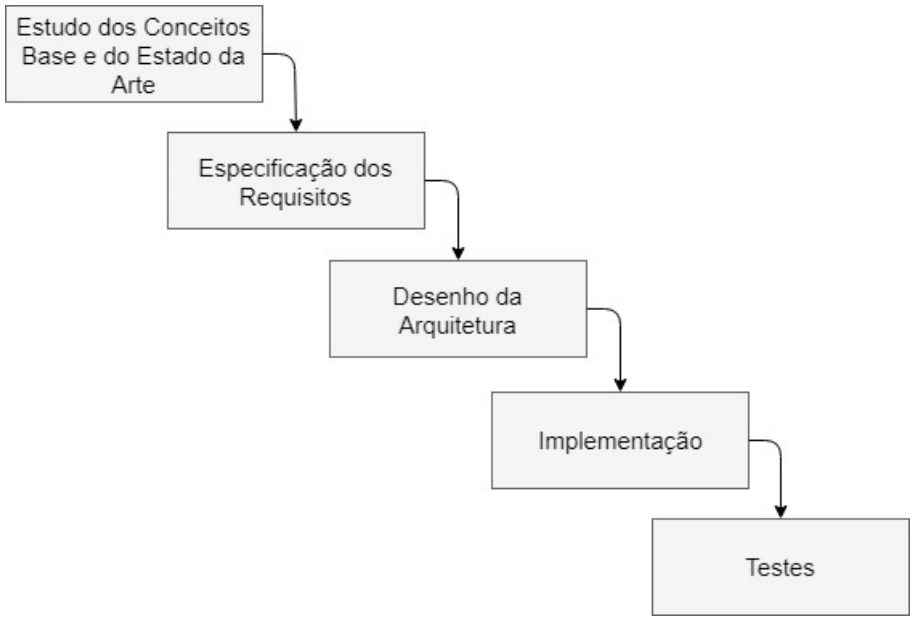


Figura 1 - Metodologia de desenvolvimento Waterfall

A fase de implementação seguiu a metodologia iterativa Cross-Industry Standard Process for Data Mining (CRISP-DM) [3] de forma adaptada. A escolha desta metodologia nesta fase, deveu-se à necessidade de experimentação e desenvolvimento de modelos dos requisitos de data mining, os quais requerem um refinamento iterativo. A metodologia CRISP-DM fornece uma abordagem estruturada no desenvolvimento destes requisitos, dividindo o seu ciclo de vida em seis fases como ilustra a Figura 2 [4].

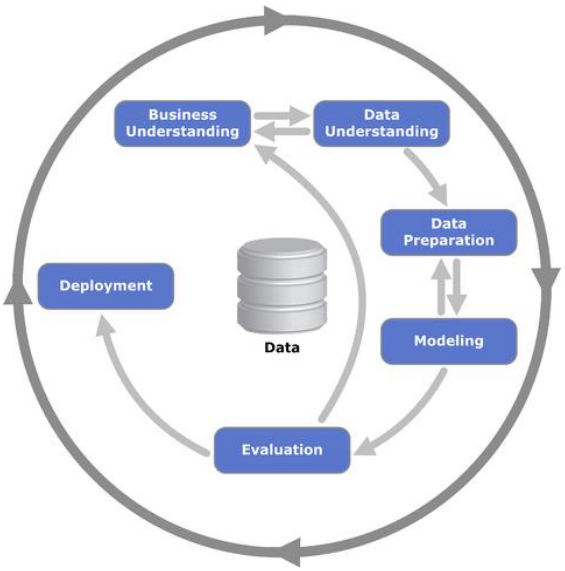


Figura 2 - Metodologia iterativa CRISP-DM

1. Compreensão do negócio: consiste na compreensão dos objetivos e requisitos do projeto, com o propósito de utilizar esse conhecimento na definição do problema e de critérios de sucesso [3]. A primeira fase, foi realizada nos estágios iniciais do projeto e não na fase de implementação.
2. Compreensão dos dados: tem por objetivo familiariza-se com os dados que serão usados, recolhendo informação sobre os seus atributos e quantidade de dados. Também uma primeira análise exploratória nos dados é realizada, com o objetivo de identificar valores em falta, outliers, e entender a distribuição dos valores dos atributos [3].
3. Preparação dos dados: é constituída por um conjunto de atividades que permitem construir a estrutura de dados de entrada do modelo, a partir do conhecimento adquirido nas fases anteriores. Inclui as tarefas de limpeza, integração e transformação dos dados [3]. A secção 3.3 deste relatório, descreve as atividades de preparação dos dados realizadas nesta fase.
4. Modelação: inicia-se após a fase anterior, e consiste na seleção de técnicas e respetivos algoritmos a usar na resolução do problema. O método de teste do modelo também é definido nesta fase [3].
5. Avaliação dos modelos: comparação, avaliação do desempenho dos modelos e validação dos objetivos definidos na primeira fase. Depois disto toma-se uma decisão, se o modelo está pronto para a fase seguinte ou necessita de ser revisto numa nova iteração no processo [3].
6. Deployment: o modelo escolhido é integrado na aplicação, com a implementação de um processo genérico capaz de dar resposta aos pedidos do utilizador ou de outras aplicações de suporte à decisão [3].

Durante o estágio foram realizadas reuniões semanais com os Orientadores da empresa e reuniões mensais com o Orientador do Departamento de Engenharia Informática (DEI). As reuniões semanais tiveram o objetivo de fazer a monitorização do progresso do trabalho e dos riscos, esclarecimento de dúvidas, discussão das dificuldades encontradas, e validação dos objetivos de negócio das funcionalidades. As reuniões mensais foram agendadas com o objetivo de discutir os avanços realizados, validar a documentação produzida e orientar as próximas tarefas do trabalho.

2.2. Ferramentas

As ferramentas de gestão de projeto utilizadas ao longo do projeto foram as usadas internamente na empresa, o Jira e a Wiki. O Jira é uma plataforma de desenvolvimento que permite fazer o registo e atribuição de tarefas. A Wiki serve para o registo de notas, partilha de conhecimento e documentação interna do trabalho. Foi utilizado o diagrama de Gantt [5] no planeamento, calendarização das tarefas e para visualização do progresso do projeto. Finalmente utilizou-se o Integrated Development Environment (IDE) Pycharm no desenvolvimento do código.

2.3. Planeamento

Esta secção apresenta o planeamento dos dois semestres e os respetivos desvios ocorridos em relação ao planeamento inicial. Recorreu-se à técnica de estimação de esforço das tarefas por analogia [2], baseada na experiência do estagiário em tarefas semelhantes de outros projetos realizados no curso.

2.3.1. Primeiro semestre

O trabalho do primeiro semestre teve como principal objetivo o estudo dos conceitos envolvidos e a elaboração da proposta de estágio. A Tabela 1 apresenta o planeamento e a calendarização das tarefas do primeiro semestre, com a data de início, de fim, e duração estimada em dias. A monitorização do progresso do trabalho foi realizada semanalmente. O diagrama de Gantt encontra-se no anexo A.

Tarefa	Data de início	Data de fim	Estimativa
Gestão do projeto	18-09-2017	19-01-2018	-
Conhecimento base	18-09-2017	13-10-2017	20
• Telecomunicações	18-09-2017	22-09-2017	5
• Data mining	25-09-2017	06-10-2017	10
• Preparação dos dados	09-10-2017	13-10-2017	5
Estado da arte	16-10-2017	10-11-2017	20
• Sistemas de suporte à decisão concorrentes	16-10-2017	20-10-2017	5
• Abordagens analíticas em telecomunicações	23-10-2017	07-11-2017	12
• Ferramentas de data mining	08-11-2017	10-11-2017	3
Especificação de requisitos	13-11-2017	24-11-2017	10
• Requisitos funcionais	13-11-2017	17-11-2017	5
• Requisitos não funcionais	20-11-2017	22-11-2017	3
• Restrições técnicas e de negócio	23-11-2017	24-11-2017	2
Desenho da arquitetura	27-11-2017	15-12-2017	15
Familiarização com as tecnologias	18-12-2017	22-12-2017	5
Início do desenvolvimento de modelos	26-12-2017	12-01-2018	13
Detalhes	15-01-2018	19-01-2018	5
Preparação da defesa	22-01-2018	26-01-2018	5
Total	18-09-2017	26-01-2018	93

Tabela 1 - Planeamento do primeiro semestre

O trabalho relativo ao primeiro semestre dividiu-se nas seguintes tarefas:

- **Conhecimento base**: estudo de conceitos de telecomunicações e de data mining;
- **Estado da arte**: estudo dos produtos existentes na empresa que fornecem dados a este projeto, e de abordagens analíticas relacionadas com a implementação dos requisitos funcionais. Segue-se o levantamento e análise comparativa de sistemas de suporte à decisão concorrentes e de ferramentas de data mining;
- **Especificação de requisitos**: inicia-se após o estudo dos Conceitos Base e do Estado da Arte, e consiste no levantamento e especificação de requisitos;
- **Desenho da arquitetura**: inicia-se após a definição dos requisitos, e consiste em propor e desenhar uma arquitetura para a plataforma que suporte todos requisitos. Inclui a escolha de tecnologias e frameworks a usar na fase implementação;
- **Familiarização com as tecnologias**: tem como objetivo a realização de tutorias de adaptação às tecnologias e frameworks selecionadas;
- **Início do desenvolvimento de modelos dos requisitos**: início da experimentação e análise comparativa de modelos do primeiro requisito funcional;
- **Preparação da defesa**: consiste na preparação da defesa de estágio intermédia.

O relatório foi desenvolvido ao longo do semestre, no entanto a última semana foi destinada para a sua revisão.

2.3.2. Segundo semestre

O trabalho do segundo semestre teve como principal objetivo a implementação da plataforma especificada no primeiro semestre. A Tabela 2 apresenta o planeamento e a calendarização das tarefas do segundo semestre. O registo da duração real das tarefas e a monitorização do progresso do trabalho, foram realizadas semanalmente. O diagrama de Gantt encontra-se no anexo A.

Tarefa	Data de início	Data de fim	Estimativa
Gestão do projeto	30-01-2018	29-06-2018	-
Revisão do projeto	30-01-2018	02-02-2018	4
Implementação	05-02-2018	18-05-2018	74
• Codificação do módulo	05-02-2018	18-05-2018	-
• Experimentação e desenvolvimento de modelos	05-02-2018	18-05-2018	-
• Integração na plataforma	05-02-2018	18-05-2018	-
Testes	21-05-2018	15-06-2018	20
• Requisitos funcionais	21-05-2018	01-06-2018	10
• Requisitos não funcionais	04-06-2018	15-06-2018	10
Detalhes	18-06-2018	29-06-2018	10
Preparação da defesa	02-07-2018	06-07-2018	5
Total	30-01-2018	06-07-2018	113

Tabela 2 - Planeamento do segundo semestre

O trabalho relativo ao segundo semestre dividiu-se nas seguintes tarefas:

- **Revisão do projeto**: consiste na revisão geral do trabalho segundo as orientações fornecidas pelos júris na defesa intermédia do estágio;
- **Implementação**: nesta fase como mencionado anteriormente, seguiu-se a metodologia iterativa CRISP-DM. Consiste no desenvolvimento dos módulos da

aplicação e na experimentação e desenvolvimento de modelos dos requisitos funcionais;

- **Testes**: planeamento e realização de testes de aceitação, aos requisitos funcionais e não funcionais. Também inclui a correção dos defeitos encontrados.
- **Preparação da defesa**: consiste na preparação da defesa de estágio final.

O artigo foi desenvolvido durante os meses de maio e junho em simultâneo com as outras tarefas. O relatório, ao longo do semestre e as últimas duas semanas destinadas para sua revisão e acerto de detalhes.

2.3.3. Desvios

Durante o primeiro semestre seguiu-se o planeamento sem grandes dificuldades. Apenas registou-se um pequeno desvio no estudo do Conhecimento Base e de Estado da Arte. Este desvio deveu-se à especificação dos requisitos ter sido realizada após o seu estudo, e terem surgido alguns novos conceitos de data mining importantes de serem estudados que inicialmente não foram planeados por desconhecimento.

O planeamento do segundo semestre foi necessário efetuar alguns ajustes para lidar com alguns atrasos e questões. A Tabela 3 apresenta o planeamento final das tarefas do segundo semestre. O seu diagrama de Gantt encontra-se no anexo A.

Tarefa	Data de início	Data de fim	Estimativa
Gestão do projeto	30-01-2018	29-06-2018	-
Revisão do projeto	30-01-2018	02-02-2018	4
Estudo de séries temporais	05-02-2018	09-02-2018	5
Implementação	12-02-2018	18-05-2018	69
• Codificação do módulo	12-02-2018	18-05-2018	-
• Experimentação e desenvolvimento de modelos	12-02-2018	18-05-2018	-
• Integração na plataforma	12-02-2018	18-05-2018	-
Testes	21-05-2018	15-06-2018	20
• Requisitos funcionais	21-05-2018	05-06-2018	12
• Requisitos não funcionais	06-06-2018	15-06-2018	8
Detalhes	18-06-2018	29-06-2018	10
Preparação da defesa	02-07-2018	06-07-2018	5
Total	30-01-2018	06-07-2018	113

Tabela 3 - Planeamento final do segundo semestre

O primeiro desvio registado deveu-se ao estudo de séries temporais importante ao desenvolvimento do requisito de previsão de consumo, não ter sido realizado no primeiro semestre. Este desvio foi colmatado com a redução do tempo alocado à implementação em uma semana. Outro desvio verificado foi na implementação do primeiro requisito, ter levado mais tempo que o esperado, devido à necessidade de adicionar algumas restrições, realizar algumas correções e experimentar novas abordagens no caso da seleção de atributos com maior importância. Todas estas questões que surgiram, deveram-se à inexperiência do estagiário no desenvolvimento de um primeiro requisito de data mining. Este desvio foi colmatado com a alocação de esforço adicional. Importa referir que estes desvios ocorridos não impediram o projeto de alcançar os objetivos propostos.

2.4. Riscos

Um risco é um evento que quando ocorre pode impactar nos objetivos do projeto [6]. A presente secção descreve o processo genérico de gestão de riscos a realizar ao longo de todas as fases de um projeto. Segue-se a aplicação deste processo neste projeto, com a identificação, análise, monitorização e registo dos riscos verificados.

2.4.1. Processo genérico de gestão de riscos

O processo de gestão de riscos é constituído por um conjunto de etapas, identificação, análise de exposição, e monitorização dos riscos identificados [2] [6]. A identificação dos riscos deve ser realizada nos estágios iniciais do projeto, tendo em conta os seus objetivos e cronograma. A análise dos riscos identificados divide-se em duas vertentes de análise, a qualitativa e quantitativa [6]. A análise qualitativa, avalia os riscos pelo seu impacto e probabilidade de ocorrer.

O impacto de um risco indica a perda ou efeito sobre os objetivos do projeto. Inclui os seguintes três níveis:

- Alto: não permite alcançar os objetivos do projeto;
- Médio: permite alcançar os objetivos do projeto, com a alocação de esforço adicional ao projeto;
- Baixo: permite alcançar os objetivos, sem a necessidade de alocar esforço adicional ao projeto.

A probabilidade de ocorrência indica a possibilidade que o risco tem de ocorrer tendo determinado impacto. Inclui os seguintes três níveis:

- Alta: previsível de vir a ocorrer;
- Média: pode vir a ocorrer;
- Baixa: não é previsível de vir a ocorrer.

A Tabela 4 apresenta a matriz de exposição dos riscos, que relaciona o seu impacto e a probabilidade de ocorrer.

		Probabilidade		
		Baixa	Média	Alta
Impacto	Baixo	Baixa	Baixa	Média
	Médio	Baixa	Média	Alta
	Alto	Média	Alta	Alta

Tabela 4 - Matriz de exposição dos riscos

A análise quantitativa, tem como objetivo priorizar os riscos a partir da análise qualitativa realizada e pela janela temporal. A janela temporal indica o momento que o risco poderá vir a ocorrer, ou que será necessário lidar com ele. Inclui os seguintes três níveis:

- Longo prazo: o risco ocorre na fase de testes e validação do projeto;
- Médio prazo: o risco ocorre na fase de implementação do projeto;
- Curto prazo: o risco ocorre nos estágios iniciais do projeto.

Finalmente a etapa de monitorização dos riscos identificados, verifica se os riscos identificados estão a ser mitigados ou não com o plano criado, e se novos riscos são necessários de analisar e avaliar.

2.4.2. Identificação e análise de riscos

Os riscos relacionados com o desenvolvimento deste projeto foram descritos utilizando a seguinte estrutura, nome do risco, a sua consequência no caso de vir a ocorrer, o seu impacto, a sua probabilidade de ocorrer e o plano de mitigação a usar para evitar que o risco se torne um problema. Tanto o impacto do risco como a probabilidade de o risco ocorrer, foram classificados pela perceção do estagiário e orientadores.

ID: R1
Nome: alteração dos requisitos funcionais
Consequência: atraso no projeto
Impacto: alto
Probabilidade: baixa
Plano de mitigação: redução do scope do projeto

ID: R2
Nome: inexperiência no uso das tecnologias ou frameworks escolhidas
Consequência: atraso no início da fase de implementação do projeto
Impacto: alto
Probabilidade: baixa
Plano de mitigação: após definidas as tecnologias e frameworks a usar, estudar e realizar antecipadamente tutoriais com o propósito de aprender e estar familiarizado com estas

---

ID: R3
Nome: compreensão limitada da área de negócio das telecomunicações
Consequência: implementação incorreta dos requisitos funcionais
Impacto: alto
Probabilidade: alta
Plano de mitigação: comunicação com os elementos da equipa para compreender e esclarecer dúvidas sobre os dados de telecomunicações

---

ID: R4
Nome: dificuldade em perceber o modelo de dados dos sistemas fonte dos produtos
Consequência: implementação incorreta dos requisitos funcionais
Impacto: médio
Probabilidade: média
Plano de mitigação: alocar tempo para exploração dos dados dos sistemas fonte, e pedir ajuda a quem desenvolveu sempre que necessário

---

ID: R5
Nome: fraca qualidade dos dados
Consequência: uma vez que a qualidade dos dados de entrada no modelo tem grande impacto nos resultados obtidos, a fraca qualidade dos dados pode levar a que um determinado requisito não consiga alcançar os seus objetivos
Impacto: alto
Probabilidade: baixa
Plano de mitigação: alocar mais esforço à etapa de preparação dos dados, e alargar as fontes de dados a outros operadores

---

A Tabela 5 apresenta a matriz de exposição dos riscos identificados, pelo seu impacto e probabilidade de ocorrer.

		Probabilidade		
		Baixa	Média	Alta
Impacto	Baixo			
	Médio		R4	
	Alto	R1, R2, R5		R3

Tabela 5 - Matriz de exposição dos riscos identificados

Depois de identificar e analisar a exposição dos riscos, segue-se a sua análise quantitativa. A Tabela 6 apresenta os riscos ordenados pela sua exposição ao projeto e pela janela temporal, que pode vir a ocorrer.

Risco	**Exposição**	**Janela temporal**
R3	Alta	Curto prazo
R4	Média	Curto prazo
R1	Média	Médio prazo
R2	Média	Médio prazo
R5	Média	Médio prazo

Tabela 6 - Priorização dos riscos identificados

2.4.3. Monitorização e registo de riscos

Ao longo do projeto, os riscos identificados foram monitorizados semanalmente. Dos riscos identificados, no primeiro semestre mitigou-se a ocorrência do Risco 3, compreensão limitada da área de negócio das telecomunicações, quando o estagiário começou a conhecer e a familiarizar-se com dados que seriam usados na implementação deste projeto. O risco 4, dificuldade em perceber o modelo de dados dos sistemas fonte também foi mitigado. Os planos de mitigação criados previamente para cada um destes dois riscos, permitiram que estes fossem efetivamente mitigados. O Risco 1, alteração dos requisitos funcionais, registaram-se pequenas alterações e correções no primeiro requisito funcional no início do segundo semestre, no entanto sem grandes problemas. Quanto aos restantes riscos, a sua probabilidade de ocorrer foi diminuindo até que deixou de ser um problema, à medida que o estagiário ganhou experiência e conhecimento no domínio em que trabalhou. Concluindo, os planos de mitigação criados foram eficazes na mitigação dos riscos verificados, de modo que estes não impediram o projeto de alcançar os objetivos definidos.

Capítulo 3

Conhecimento Base

Este capítulo apresenta o estudo dos conceitos gerais de telecomunicações e data mining. Tem como objetivo compreender e perceber a sua aplicação no desenvolvimento deste projeto de estágio.

3.1. Conceitos gerais de telecomunicações

Dada a importância de compreender alguns dos principais conceitos de telecomunicações na realização deste trabalho, esta secção define e caracteriza os conceitos introdutórios desta temática.

3.1.1. Serviço

Os serviços de telecomunicações definem-se como o conjunto de capacidades de comunicação disponibilizados aos clientes. Atualmente, a indústria de telecomunicações oferece múltiplos serviços de comunicação, tanto locais como de longa distância, de modo a proporcionar uma diversidade de serviços. Estes serviços incluem chamadas, mensagens de texto, internet móvel e fixa, televisão, transmissão de informação via web. Os serviços disponibilizados estão subdivididos genericamente em três tipos de tarifário, o pré-pago paga um serviço que permite uma utilização durante um limite temporal, o pós-pago utiliza e paga à posteriori, e o híbrido tem um plafond inicial incluído na mensalidade pós-paga e se eventualmente esgotar pode recarregar e continuar a utilizar o serviço em modo Pré-Pago.

3.1.2. Segmentação de clientes

Atualmente os operadores de telecomunicações mantêm uma grande quantidade de dados nos seus sistemas operacionais sobre os seus clientes. A análise destes dados permite desenvolver e aumentar a eficiência e eficácia das suas atividades de marketing. Existem duas estratégias possíveis de marketing, a do produto e a do cliente [7]. A primeira apesar de menos utilizada pelo facto de se focar apenas no produto, tem interesse quando a empresa consegue criar um produto inovador, que crie enorme interesse de compra no mercado. Atualmente, os operadores de telecomunicações adotam a segunda estratégia, que consiste em criar serviços e campanhas de marketing à medida dos seus clientes (por exemplo oferecer um bónus no carregamento num determinado período temporal). De maneira a conseguir esta estratégia de marketing direcionada aos clientes, a segmentação de clientes tem um papel fundamental. A segmentação de clientes consiste no processo de dividir os clientes em grupos, de acordo com as suas características e hábitos de consumo. Isto permite conhecer os seus clientes, proporcionando uma atividade de marketing mais eficiente e direcionada para um determinado público-alvo. Neste processo de segmentação de clientes é necessário ter em conta alguns fatores, nomeadamente a distinção entre clientes normais e empresariais, verificar se um determinado cliente possui ou não o serviço e, se não tem atrasos nas suas obrigações contratuais [7].

3.2. Data mining

Dada a importância do data mining neste projeto, na compreensão dos hábitos de consumo dos utilizadores, estudo de preferências e análises de mercado preditivas, esta secção apresenta o estudo deste tópico. O data mining define-se como o processo que transforma os dados provenientes das fontes operacionais em informação útil no apoio à tomada de decisão ou avaliação de resultados. Constituído por um conjunto de etapas, começando pela preparação dos dados, extração de conhecimento com recurso a algoritmos de aprendizagem computacional, avaliação de modelos e finalmente apresentação ou disponibilização através de uma aplicação do conhecimento extraído [8].

O data mining disponibiliza um conjunto de técnicas que permitem extrair conhecimento dos dados, dividindo-se em dois grupos, as supervisionadas incluem a regressão, classificação e seleção de atributos, e as não-supervisionadas incluem o clustering e associação [8]. As supervisionadas são usadas para prever um resultado de uma classe-alvo e para calcular a importância de atributos (ou features) em relação à classe-alvo. Necessitam de dados de treino que contenham esta classe, no entanto apenas a seleção de atributos vai ser aplicada neste projeto. As não-supervisionadas não contêm a classe-alvo nos dados de treino, e aplicam-se na descoberta de relações, padrões e agrupamentos nos dados. O restante desta secção descreve a área de aplicação, os requisitos e métodos das técnicas de clustering, associação e seleção de atributos. A última secção apresenta a aplicação de métodos de séries temporais a este projeto.

3.2.1. Clustering

Na indústria das telecomunicações, o clustering aplica-se na identificação do perfil de clientes com maior e menor adesão a uma determinada campanha de marketing e na deteção de anomalias de consumo [8]. O clustering baseia-se no princípio de agrupar elementos descritos por valores numéricos em grupos ou clusters, cuja semelhança entre elementos do mesmo grupo é elevada e a semelhança entre elementos de grupos diferentes é baixa. A semelhança entre elementos é calculada por uma função de proximidade, que recebe dois elementos e devolve a distância que os separa, calculada de acordo com a sua similaridade. Duas funções populares usadas no cálculo da semelhança entre dois elementos, é a Euclidean e Manhattan. A distância Euclidean [8], calcula a distância em linha reta entre os atributos de dois elementos, $i = (x_{i1}, x_{i2} \dots x_{in})$ e $j = (xj_1, x_{j2} \dots x_{jn})$, ambos descritos por um conjunto de n atributos, usando a seguinte fórmula:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}$$

A distância Manhattan [8], calcula a distância em blocos entre dois elementos usando a seguinte fórmula:

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|$$

Dos métodos de clustering existentes na literatura apresenta-se dois dos mais conhecidos e usados na indústria [9], o método de clustering hierárquico e particionamento K-Means.

3.2.1.1. Método hierárquico

O método hierárquico agrupa os elementos em grupos (ou clusters) pela criação de hierarquias ou árvores de relacionamentos entre os elementos. Neste método a similaridade entre dois grupos é medida pela função de proximidade, usando o par de elementos mais próximo entre si pertencentes a grupos diferentes.

Possui duas versões do método, a aglomerativa e a divisiva. A abordagem aglomerativa parte dos elementos isolados, vai recursivamente combinando os elementos ou grupos semelhantes pela função de proximidade. O processo recursivo termina quando todos os grupos pertencerem a um grande grupo (o topo da hierarquia) ou até satisfazer uma condição relacionada com o número de grupos. A versão divisiva, começa com um grande grupo contendo todos os elementos, vai dividindo recursivamente os elementos em grupos menores, até cada elemento ficar isolado num grupo. As decisões tomadas pelos métodos hierárquicos ao longo do processo recursivo de combinação ou divisão dos elementos são irreversíveis e não podem ser desfeitas ou melhoradas.

3.2.1.2. Método de particionamento

O método de particionamento agrupa os elementos num determinado número de grupos (ou clusters) definido previamente. O agrupamento é realizado com base na similaridade dos elementos, calculada pela função de proximidade. A função de proximidade assegura que os elementos dentro de um grupo são similares entre si, e heterogêneos em relação a outros grupos vizinhos. A maioria das aplicações que recorre a este método, usa o algoritmo K-Means [8]. O K-Means baseia-se no princípio de minimizar a distância de todos os elementos a um conjunto de k elementos definidos como centros dos grupos. O algoritmo começa por escolher arbitrariamente k elementos como centros dos grupos. De seguida de forma iterativa vai atribuindo cada elemento ao grupo mais próximo ou de maior similaridade, baseado nos valores médios dos elementos do grupo. Em cada iteração atualiza o cálculo do valor médio dos atributos dos grupos. Depois usando os novos centros dos grupos, os elementos são reatribuídos. O processo iterativo termina até que os grupos não apresentem alterações em relação à iteração anterior [8].

Comparando a complexidade temporal dos dois métodos, o de particionamento K-Means é superior ao hierárquico sendo capaz de lidar com grandes conjuntos de dados. O método de particionamento K-Means tem complexidade O(nkt) em que, n representa o número total de elementos, k o número de grupos, t o número de iterações necessárias no processo iterativo. O método hierárquico possui complexidade $O(n^2)$ [8] [9].

3.2.2. Associação

A associação nas telecomunicações aplica-se no estudo e análise de preferências de serviços comprados ou subscritos pelos clientes. Permite encontrar regras nos dados com a forma de $A \rightarrow B$ [8], ou seja, se comprar um ou mais itens que constituem o conjunto A, com um determinado grau de confiança também compra um ou mais itens que constituem o conjunto B. O grau de confiança de uma regra é definido por três parâmetros, o suporte, a confiança e o lift. O suporte [8] indica a percentagem de transações que contém os itens de A e B comprados em conjunto, ou seja, a probabilidade de A reunido com B:

$$\text{Suporte } (A \rightarrow B) = P(A \cup B)$$

A confiança [8] de uma regra $A \rightarrow B$, indica a percentagem de transações contendo os itens de A, que também contém os de B, ou seja, a probabilidade condicional de B sabendo que contém A:

$$\text{Confiança } (A \rightarrow B) = P(B \mid A) = \frac{\text{Suporte } (A \cup B)}{\text{Suporte } (A)}$$

O lift [8] é uma medida de correlação capaz de descartar regras de associação desinteressantes, cujas medidas de suporte e confiança não conseguem. O lift entre A e B é calculado usando a seguinte fórmula:

$$\text{Lift}\,(A, B) = \frac{P(A\ U\ B)}{P(A).P(B)}$$

Segundo [8] o valor do lift deve ser superior a um para garantir que os conjuntos A e B estão correlacionados, ou seja, a ocorrência de um deles implica a existência do outro.

O conjunto de regras geradas na associação possui um valor maior ou igual aos parâmetros de suporte, de confiança e do lift. O restante desta secção apresenta dois algoritmos que permitem extrair regras do conjunto de dados, começando por um algoritmo mais simples, o Apriori e um segundo mais eficiente que o primeiro, o Frequent Pattern Growth. Ambos recebem três parâmetros de entrada, o conjunto de transações, o suporte e a confiança. Quanto ao parâmetro do lift, este deve ser definido com o valor de um [8].

3.2.2.1. Apriori

O algoritmo Apriori é constituído por duas fases, a primeira encontra os conjuntos de itens frequentes e a segunda faz a geração de regras. Na primeira fase, o algoritmo a partir do conjunto de transações, seleciona os conjuntos de itens frequentes, correspondentes a subconjuntos de transações juntamente com o seu valor de suporte, ou seja, a percentagem de transações que contém esse conjunto de itens. A seleção destes conjuntos de itens é feita de forma repetitiva, ou seja, testa com um item, depois dois e dai em diante. Os conjuntos de itens que possuem valor inferior ao suporte são descartados. Na segunda fase o algoritmo faz a geração das regras a partir dos conjuntos de itens frequentes encontrados na primeira fase. Para cada conjunto de itens frequente, o algoritmo gera todas as combinações de regras possíveis e calcula o seu valor de confiança. A confiança de cada regra gerada é calculada pela fórmula apresentada na secção anterior. Finalmente o algoritmo devolve todas as regras que satisfaçam o valor de confiança recebido como parâmetro de entrada [8] [10].

3.2.2.2. Frequent Pattern Growth

O algoritmo Frequent Pattern Growth otimiza a geração dos conjuntos de itens frequentes do algoritmo anterior. Constituído por duas fases, a construção da árvore Frequent Pattern Tree (FP-Tree) contendo os conjuntos de itens frequentes, e a extração das regras a partir da FP-Tree. Na primeira fase de construção da FP-Tree, o algoritmo percorre o conjunto de transações duas vezes. Na primeira passagem seleciona os conjuntos de itens frequentes juntamente com o valor de suporte, ou seja, a percentagem de transações que contém esse conjunto de itens, e armazena este conjunto numa lista ordenada por ordem decrescente do seu valor de suporte. Os conjuntos de itens dessa lista, que possuírem valor inferior ao suporte são descartados. De seguida o algoritmo constrói a árvore, com uma segunda passagem no conjunto de transações. Os itens de cada transação são processados pela ordem da lista que possui o conjunto de itens frequentes e mapeados num caminho da árvore. Quando um caminho se sobrepõe a um existente na árvore, a frequência do nó desse caminho é incrementada. Para extrair as regras de associação, o algoritmo percorre todos os caminhos da árvore e devolve todos os nós, cujo seu número de ocorrências satisfaz o valor mínimo de confiança definido [8] [10].

Comparando os dois algoritmos anteriores, o Frequent Pattern Growth é mais rápido que o Apriori devido à melhor abordagem usada na primeira fase, de geração e seleção dos conjuntos de itens frequentes. O Apriori gera estes conjuntos percorrendo o conjunto de transações múltiplas vezes, enquanto o Frequent Pattern Growth utiliza uma FP-Tree necessitando apenas de o percorrer duas vezes [8] [11].

3.2.3. Seleção de atributos

A seleção de atributos (ou de features) num conjunto de dados, consiste em encontrar um subconjunto de atributos relevantes. Estes atributos têm maior capacidade para discriminar instâncias que pertencem a diferentes classes. A remoção de atributos irrelevantes, torna o modelo mais eficaz e eficiente no seu processamento.

Dos vários métodos existentes na literatura, o supervisionado de filter model é importante de estudar neste trabalho. Constituído por duas fases, a primeira para determinar e ordenar a dependência dos atributos em relação à classe-alvo baseado numa medida e a segunda de seleção do subconjunto de atributos com maior importância.

Uma medida popular para o cálculo da importância dos atributos em relação à classe-alvo na primeira fase do método filter model, é o ganho de informação [12]. O ganho de informação para um determinado atributo, é a diferença entre a informação original e a nova informação obtida após a divisão do conjunto de dados pelo atributo [8]. O seu cálculo envolve um conjunto de três etapas:

1. A primeira etapa calcula a informação necessária (ou entropia) para identificar o valor da classe-alvo do conjunto de dados (D), a partir da proporção de instâncias em cada classe-alvo:

$$\text{Info (D)} = -\sum_{i=1}^{m} \text{pi} \,.\log_2(\text{pi})$$

A variável pi representa a probabilidade de uma determinada instância pertencer à classe-alvo, m o número de valores distintos da classe-alvo. Após o cálculo da informação original, o próximo passo é o cálculo da informação necessária para cada atributo.

2. A segunda etapa calcula a informação necessária para classificar uma instância no conjunto de dados, a partir da divisão do atributo (A):

$$\text{Info A (D)} = \sum_{j=1}^{v} \frac{|\text{Dj}|}{|\text{D}|} . \text{Info(Dj)}$$

A divisão de $|\text{Dj}|/|\text{D}|$ representa o peso da partição do conjunto de dados.

3. A última etapa calcula o ganho de informação de cada atributo (A) usando a seguinte fórmula:

$$\text{Ganho (A)} = \text{Info (D)} - \text{Info A (D)}$$

Após o cálculo do ranking dos atributos ordenados pela sua importância entra a segunda fase do método filter model. A segunda fase tem por objetivo a seleção de um subconjunto de atributos com maior importância. A seleção deste subconjunto, pode ser realizado de dois modos distintos. O primeiro através de um threshold que seleciona os atributos a partir de um determinado valor de importância, e o segundo por geração e avaliação de subconjuntos de atributos a partir do ranking calculado a fim de determinar o ponto ótimo. Comparando os dois, a seleção por um threshold é mais eficiente em processamento pelo compromisso da qualidade do modelo. A seleção por teste de combinações favorece a qualidade do modelo. Por outro lado, é menos eficiente em processamento, porque necessita de gerar e avaliar iterativamente vários subconjuntos de atributos a fim de escolher o subconjunto ótimo.

3.2.4. Séries temporais

Em telecomunicações, a previsão com séries temporais é tipicamente aplicada na previsão de consumo e também na deteção de anomalias em tempo real. Os métodos de séries temporais lidam com a sazonalidade (variações que ocorrem em determinados períodos, relacionados com a época do ano) e são dependentes do tempo. O restante desta secção apresenta o conceito de estacionariedade e dois métodos de séries temporais, o Auto-Regressive Integrated Moving Average (ARIMA) [13] e o Prophet [14]. Este último foi lançado pelo Facebook em 2017 para permitir o uso por pessoas com menor conhecimento no domínio.

3.2.4.1. Estacionariedade no ARIMA

Um conceito importante na aplicação do método ARIMA é o de estacionariedade, uma vez que este modelo apenas pode ser construído com séries estacionárias no tempo. Uma série é estacionária, se as suas propriedades estatísticas permanecerem constantes ao longo do tempo. A existência de tendência e de sazonalidade são duas das razões que levam a série a ser não estacionária [13] [16].

Existem dois métodos que permitem verificar se uma série é estacionária ou não. O primeiro método consiste na visualização gráfica da variação das propriedades estatísticas da série, tal como a média móvel (cálculo em cada instante da média dos valores correspondentes ao último período sazonal, tipicamente de doze meses consecutivos) e o desvio padrão móvel ao longo do tempo. Se as propriedades da série não mudarem ao longo do tempo, então a série é estacionária. O segundo método, o teste Dickey-Fuller, assume que a hipótese nula, é a série ser não estacionária. Este teste calcula o valor do teste estatístico e alguns valores críticos para diferentes níveis de confiança. Se o valor do teste estatístico for menor que o valor critico, então a série é estacionária [13].

A diferenciação [13] [16] é uma das técnicas existentes que permite lidar com a sazonalidade e tendência da série temporal, aproximando-a da estacionariedade no tempo. Em cada instante na série, a diferenciação subtrai a observação original, $\mathrm{Y_t}$, a partir do instante anterior, $\mathrm{Y_{t-1}}$, usando a seguinte fórmula:

$$\mathrm{Y'_t} = \mathrm{Y_t} - \mathrm{Y_{t-1}}$$

3.2.4.2. ARIMA

Esta subsecção começa por descrever os modelos Auto-Regressive (AR) e Moving Average (MA), antes de descrever o método Auto-Regressive Integrated Moving Average (ARIMA).

1. O modelo AR [16] extrai a influência dos valores dos períodos anteriores em relação aos do período atual. Este modelo é desenvolvido usando a seguinte equação linear.

$$Y_t = \mathrm{c} + \varphi_1 . Y_{t-1} + \ldots + \varphi_p . Y_{t-p} + e_t$$

O parâmetro p indica a ordem de AR no modelo e representa o período de tempo atrasado da variável dependente. Os restantes parâmetros da equação, φ representa o coeficiente AR, y o valor observado, e o desvio da série no instante atual, c uma constante [16].

2. O modelo MA [16] extrai a influência dos termos de erro do período anterior no período atual. Este modelo é desenvolvido usando a seguinte equação linear:

$$Y_t = \mathrm{c} + e_t + \theta_1 . e_{t-1} - \ldots - \theta_q . e_{t-q}$$

O parâmetro q indica a ordem de MA no modelo e representa os erros de previsão atrasados. Os restantes parâmetros da equação, θ representa o coeficiente MA, y o valor observado, e o desvio da série no instante atual, c uma constante [16].

3. O modelo não sazonal ARIMA [13] [16] é constituído por três componentes, AR, Integrated (I) e MA, cada componente representada por um parâmetro inteiro positivo, p, d e q respetivamente. Estas três componentes são combinadas na seguinte equação linear:

$$Y_t = c + \varphi_1 . Y_{d\ t-1} + \ldots + \varphi_p . Y_{d\ t-p} + e_t + \theta_1 . e_{t-1} - \ldots - \theta_q . e_{t-q}$$

- Componente I [13] [15]: representada pelo parâmetro d do modelo e indica o número de vezes que a série foi diferenciada para aproximar da estacionariedade no tempo. Tipicamente de ordem um, para uma série não estacionária, e de ordem zero para uma série estacionária no tempo. Desenvolvida pela fórmula de diferenciação apresentada na secção anterior.
- Componente AR [13] [15]: representada pelo parâmetro p do modelo, indica a ordem da componente AR e significa o período de tempo atrasado. O seu valor é estimado pela Autocorrelation Function (ACF), função que permite medir a correlação entre a série original e a sua versão atrasada de p períodos de tempo.
- Componente MA [13] [15]: representada pelo parâmetro q do modelo, indica a ordem da componente MA e representa os erros de previsão atrasados. O seu valor é estimado pela Partial Autocorrelation Function (PACF), função que permite medir a correlação parcial entre a série original e sua versão atrasada.

Importa referir que os parâmetros p e q são determinados no momento em que as respetivas funções, ACF e PACF, cruzam o intervalo de confiança superior pela primeira vez. O intervalo de confiança das duas funções é calculado pela fórmula, $\pm 1.96 / \sqrt{n}$, onde a variável n, corresponde ao tamanho dos dados de histórico [13] [16]. Finalmente o modelo sazonal ARIMA [16] estende o modelo anterior, e combina as suas componentes com uma nova de sazonalidade, que indica o período de sazonalidade da série.

3.2.4.3. Prophet

O método de séries temporais Prophet em relação ao ARIMA é mais automatizado na configuração dos seus parâmetros, devido à sua capacidade de encontrar pontos de inflexão nos dados originados por mudanças de tendência. Uma novidade deste método em relação ao anterior é a possibilidade de acomodar a existência de períodos festivos sazonais. O método combina três componentes, a tendência, a sazonalidade e os períodos festivos na seguinte equação [17]:

$$y(t) = g(t) + s(t) + h(t) + \epsilon t$$

Cada uma destas componentes é modelada por uma função. A componente de tendência, $g(t)$, é modelada por uma função logística. A componente de sazonalidade, $s(t)$, por séries de Fourier. Os períodos festivos, $h(t)$, são ajustados por parametrização no modelo. Finalmente o termo de erro, ϵt, representa as mudanças originadas por circunstâncias que não são acomodadas pelo modelo [17]. Mais informação sobre detalhes da formulação de cada uma destas componentes pode ser encontrada em [17].

3.3. Preparação dos dados

A fase de preparação de dados é fundamental no desenvolvimento de requisitos de data mining, porque a qualidade dos dados de entrada no modelo tem um grande impacto no seu resultado [18]. De um modo geral nesta fase tratam-se os valores em falta, removem-se os outliers e transformam-se os valores dos atributos.

Os valores em falta no conjunto de dados de treino tendem a influenciar a precisão dos resultados. Enumeram-se de seguida alguns métodos existentes na literatura que permitem lidar com valores em falta [8] [18]:

- Ignorar o registo que possui pelo menos um valor do atributo por preencher. Método simples, mas reduz o tamanho do conjunto de dados de treino e consequentemente a qualidade do modelo;
- Substituir o valor em falta de um atributo pelo valor médio quando o atributo é numérico ou pelo valor mais frequente quando o atributo é nominal;
- Outra alternativa, substituir o valor em falta usando um valor por omissão definido por uma constante.

Os outliers são valores anómalos que divergem de um padrão dos dados da amostra. A sua existência aumenta a variância dos valores dos atributos afetando a qualidade do modelo [18]. Alguns métodos existentes na literatura que permitem detetar outliers, são os métodos estatísticos e os de clustering. Existem dois métodos estatísticos, o de box-plot por análise visual ou por distribuição normal. Segundo [8], a partir da distribuição normal pode se assumir para um atributo, que o intervalo de valores correspondente ao valor médio ± três vezes o seu desvio padrão, contém quase a percentagem total dos valores considerados normais, e os restantes valores que se encontrarem fora desse intervalo podem ser definidos como outliers [8]. Os métodos de clustering, assumem que os valores normais pertencem a grandes e densos clusters, enquanto os outliers pertencem a pequenos clusters [8]. Após detetados, devem ser tratados de forma semelhante aos métodos de tratamento de valores em falta.

A transformação de variáveis envolve a agregação e também a conversão dos valores dos atributos em formatos apropriados, com recurso a operações de discretização e normalização [18]. A agregação permite obter uma representação reduzida do conjunto de dados. ´Tanto a discretização como a normalização tornam o processamento mais eficiente e facilitam a descoberta e compreensão dos padrões encontrados. A discretização transforma os valores brutos dos atributos numéricos em valores nominais pela divisão dos seus valores em intervalos, por exemplo a categorização dos grupos etários [18]. O Binning [8] divide os valores dos atributos distribuindo-os em partições de igual tamanho ou frequência. A normalização aplica-se aos atributos numéricos num amplo intervalo de valores, dimensionando-os num intervalo menor, com o objetivo de dar a todos os atributos o mesmo peso e também atenuar o efeito dos outliers. O Min-Max [8] é uma técnica de normalização que transforma os valores de um atributo v_i, no intervalo de min_A a max_A, para um novo intervalo de new_max_A a new_min_A, usando a seguinte fórmula:

$$v'_i = \frac{v_i - min_A}{max_A - min_A} (new\_max_A - new\_min_A) + new\_min_A$$

Capítulo 4

Estado da Arte

Este capítulo apresenta a informação recolhida dos sistemas de suporte à decisão reais do mercado concorrente, de forma a contribuírem com funcionalidades na especificação da solução pretendida. De seguida, expõe-se o estudo de abordagens de implementação semelhantes dos requisitos pretendidos neste projeto, numa revisão da literatura de artigos publicados nesta área de investigação. Dada a importância do data mining no desenvolvimento deste projeto, por fim fez-se o levantamento e análise das suas ferramentas.

4.1. Sistemas de suporte à decisão concorrentes

Esta secção apresenta a recolha de informação dos sistemas de suporte à decisão concorrentes de três empresas, IBM, AsiaInfo e Oracle. Tendo em conta a informação limitada disponibilizada por estas empresas dos seus sistemas, apenas foi possível obter as funcionalidades que cada um fornece.

A IBM possui um sistema de suporte à decisão chamado IBM Telecom Analytics [20], destinado ao mercado das telecomunicações. A solução da IBM a partir do grande volume de dados proveniente dos sistemas operacionais do operador, permite extrair conhecimento útil ao processo de tomada de decisão e exploração de novas oportunidades na área de marketing. O conhecimento extraído é apresentado ao analista num dashboard. Do conjunto de funcionalidades fornecidas por este sistema, destaca-se a gestão de clientes prováveis de churn, estimação do lucro do cliente ao longo do seu tempo de vida provável no operador, análise da eficácia das campanhas de marketing. Outra funcionalidade fornecida, é a que analisa e agrupa os clientes pelos seus interesses e preferências, de modo a sugerir-lhes serviços complementares ou de maior valor acrescentado. A Figura 3 apresenta um dos dashboards deste sistema com os fatores de churn mais frequentes ao longo do tempo [21].

Figura 3 - Dashboard do sistema IBM Telecom Analytics

A AsiaInfo no seu portefólio de produtos possui um sistema de suporte à decisão chamado, Telecom Business Intelligence [22], desenvolvido especificamente para a indústria de telecomunicações. Esta solução faz a recolha e tratamento de dados estruturados e não estruturados com o objetivo de fornecer indicadores, que permitem acompanhar o desempenho do negócio. Adicionalmente fornece funcionalidades inteligentes, capazes de obter conhecimento sobre o comportamento do cliente e sobre o desempenho dos serviços fornecidos. Destaca-se a previsão de receita e análise do comportamento de consumo do cliente, a segmentação de clientes para estudo de preferências dos clientes, a fim de melhorar a eficácia das campanhas de marketing. Na área dos serviços permite estimar o seu consumo, classificar o seu desempenho com base em indicadores de desempenho, obter uma visão de margem de lucro, a fim de poder ajustar o seu preço de venda. Outra funcionalidade importante, é a análise do desempenho dos canais de venda, para atividades de recarga, emissão de cartões e adição de novos assinantes de serviços. A Figura 4 mostra um extrato da interface deste sistema com os seus indicadores de desempenho [22].

Figura 4 - Dashboard do sistema Telecom Business Intelligence

A Oracle fornece uma solução de suporte à decisão para as telecomunicações chamado TBIDS [23] com relatórios e análises Online Analytical Processing (OLAP) para análise do negócio, do que está a acontecer no presente e no futuro através de análises preditivas. Das análises preditivas, evidenciam-se a identificação de clientes possíveis de deixar um serviço, deteção de uma possível fraude de subscrição por parte dos clientes, previsão da utilização dos canais de comunicação. Além destas existem ainda, a análise de rentabilidade do cliente, a sugestão de serviços adicionais a clientes (cross-selling) e de serviços de maior valor (up-selling), configuração de campanhas de acordo com as necessidades dos clientes.

A Tabela 7 resume a informação recolhida destes sistemas, apresentando a comparação das funcionalidades que cada um possui.

Funcionalidade	IBM	AsiaInfo	Oracle
Análise OLAP	Sim	Sim	Sim
Análise de churn de clientes	Sim	Não	Sim
Deteção de fraude de subscrição	Não	Não	Sim
Estimação da rentabilidade do cliente	Sim	Sim	Sim
Análise de eficácia das campanhas	Sim	Sim	Não
Estudo de preferências e sugestão de serviços a clientes	Sim	Sim	Sim
Previsão do consumo de serviços/ utilização dos canais de comunicação	Não	Sim	Sim

Tabela 7 - Comparação das funcionalidades dos sistemas concorrentes

A Tabela 6 mostra a existência de uma certa semelhança no conjunto de funcionalidades fornecido por cada sistema. Destes sistemas importa referir as funcionalidades que podem ser relevantes e incluídas de forma adaptada neste projeto, que vai ser desenvolvido à medida dos produtos atualmente existentes na empresa. Destas funcionalidades destacam-se, o estudo de preferências de serviços pelos clientes, previsão de consumo dos serviços, e análise de eficácia das campanhas. O capítulo 5 do relatório, detalha a especificação de requisitos deste projeto com estas funcionalidades e com novas funcionalidades criadas à medida das necessidades e produtos existentes na empresa.

4.2. Abordagens analíticas em telecomunicações

Nesta secção revêm-se trabalhos relacionados sobre os requisitos funcionais deste estágio, no contexto de telecomunicações.

Em 2007, o autor Jansen S. M. H propôs uma abordagem para o problema da segmentação de clientes empresariais por perfil de uso de chamadas para a Vodafone, com o objetivo de encontrar os clientes mais rentáveis e desenvolver estratégias de marketing eficazes e direcionadas ao público-alvo [24]. A abordagem seguida neste estudo foi dividida em três fases. A primeira fase consistiu em selecionar os dados relevantes para a análise, divididos nas categorias de dados pessoais e de chamadas. A segunda fase consistiu em agrupar um subconjunto de clientes representativos correspondente a 3% do total de clientes do operador pelos seus hábitos de uso de chamadas, recorrendo à técnica de clustering. A decisão neste estudo de escolher este pequeno subconjunto de clientes, foi tomada para evitar problemas de desempenho. Os modelos de clustering que obtiveram melhores resultados, foram o Gath-Geva com quatro grupos e o Gustafson-Kessel com seis grupos. Os clusters alcançados neste estudo apresentaram diferenças nos hábitos de consumo de SMS, chamadas de voz, duração das chamadas, chamadas internacionais, diurnas e noturnas. Depois de obtido o conjunto de clusters, teve início a terceira e última fase com recurso ao classificador Support Vector Machines (SVM). O SVM permitiu classificar o perfil de uso de chamadas de um novo cliente, utilizando os seus dados pessoais com uma precisão de 80.3% para quatro grupos e 78.5% para seis grupos.

Em 2008, S. T., & Sampaio, R. J. B propuseram um modelo para prever a curto prazo o consumo de um serviço de telecomunicações [25]. Dado que o uso do serviço apresenta um comportamento não linear originado pela existência de tendência e sazonalidade, neste estudo foram utilizados dois algoritmos de redes neuronais, o Multilayer Perceptron e o Radial Basis Function (RBF). A escolha destes dois algoritmos teve em conta a sua capacidade de lidar com

uma relação não linear entre a variável dependente e independente. No estudo de diferentes modelos, o conjunto de dados agregado ao mês foi dividido em dois conjuntos, o de treino constituído por um histórico de três, quatro e seis meses, e o conjunto de teste constituído por apenas um mês à frente. A métrica utilizada na avaliação dos resultados foi o Mean Squared Error (MSE). O modelo Multilayer Perceptron apresentou o seu melhor resultado de previsão para um histórico de seis meses, sendo o seu MSE de 3.91e+5. Enquanto o modelo RBF apresentou o seu melhor resultado para um histórico de três meses com um MSE de 8.79e+5. Das experiências realizadas para diferentes períodos de histórico, o modelo Multilayer Perceptron apresentou uma melhor qualidade de previsão, embora com um pior desempenho computacional.

Em 2015, os autores Wang, M., Wang, Y., Wang, X., & Wei, Z propuseram um modelo baseado no Auto-Regressive Integrated Moving Average (ARIMA), com o objetivo de prever o rendimento de um operador de telecomunicações [26]. Neste estudo, foram utilizados dados agregados ao mês no período de dois anos e meio. Sendo os dados dos primeiros dois anos para análise e treino do modelo, e os restantes seis meses para validação do modelo. O método usado para verificar a estacionariedade da série temporal foi a análise das propriedades estatísticas, como a média, variância e coeficiente de correlação, verificando se estas permanecem constantes ao longo do tempo. Neste estudo verificou-se que a série era não estacionária. Para a tornar estacionaria e ser possível usar o método ARIMA, foi aplicada uma dupla transformação diferencial à série original. De seguida com recurso às funções, Autocorrelation Function (ACF) e Partial Autocorrelation Function (PACF), foram identificados os possíveis valores dos parâmetros p e q. Após a identificação dos possíveis valores dos parâmetros do modelo, o próximo passo foi avaliar um conjunto de modelos variando a sua parametrização. Dos modelos avaliados foi o modelo ARIMA com a parametrização ($p = 1, d = 2, q = 1$) que apresentou um melhor resultado. O modelo obteve um erro médio de 1%, sendo o menor erro de 0.4% no quinto mês, e o maior de 1.8% no terceiro mês.

Em 2017, Hideaki Hayashi comparou o desempenho de dois métodos de séries temporais num contexto diferente, na previsão do número de voos realizados nos EUA [27]. Os dois métodos utilizados foram o Prophet e o ARIMA, este último com os parâmetros configurados manualmente e automaticamente com os valores por default. Neste estudo, foi usado um conjunto de dados agregados mensalmente num período de três anos. A partir deste conjunto, os primeiros dois anos foram usados para treinar o modelo, e o terceiro para testar o modelo. A comparação dos resultados mostra que o Prophet foi inferior ao ARIMA com os parâmetros configurados manualmente, e superior ao ARIMA com os parâmetros configurados automaticamente com os valores por default. O autor concluiu que o método ARIMA, ao contrário do Prophet, exige configuração manual dos parâmetros do modelo para ter bons resultados. Isto significa que o método ARIMA requer bastante conhecimento no domínio para ser configurado manualmente.

Em 2016, os autores Rokhmatul & Hira apresentaram uma abordagem de duas fases para efetuar uma análise de market basket nas telecomunicações [28]. A abordagem apresentada no artigo, permite gerar relações entre serviços (internet, telefone, televisão, fixo) para atividades de marketing, como a criação de pacotes ou sugestão de serviços de valor acrescentado a clientes. A primeira fase, agrupou em clusters os clientes semelhantes com base na sua demografia e uso dos serviços, com recurso ao método de clustering K-Means. Dos clusters alcançados foi possível classificar os clientes de acordo com o uso dos serviços, identificar os mais lucrativos e os prováveis de deixar um determinado serviço. Na segunda fase, foi aplicada uma análise de market basket nos serviços para o cluster de clientes rentáveis para gerar um conjunto de relações entre os serviços. Foi aplicado o algoritmo Apriori da técnica de regras de associação, configurado com um valor de suporte de 40% e um valor de confiança de 50%.

Para um conjunto de seis serviços, o algoritmo foi capaz de gerar vinte relações entre os serviços, com um valor de confiança entre 50.8% e 99.9%.

Em 2017, os autores Iglesias, J. A., Ledezma, A., Sanchis, A., & Angelov, P. propuseram uma abordagem para analisar os registos de chamadas, com o objetivo de extrair e agrupar diferentes comportamentos de chamadas e detetar anomalias em tempo real [29]. Neste estudo foram utilizados os dados correspondentes aos registos de chamadas dos clientes. Para cada cliente foi criado um perfil diário, que resume o seu comportamento de chamadas nesse dia, contendo o momento da primeira chamada, a duração e o número de chamadas agregadas por quatro períodos ao dia (madrugada, manhã, tarde e noite), ou seja, das 0h às 6h, das 6h às 12h, e daí em diante. O estudo propõe um método não supervisionado com recurso à técnica de clustering para compreender o comportamento de chamadas, e detetar as chamadas que podem ser consideradas outliers. Pela comparação do perfil de chamadas recente de um utilizador com o seu histórico, é possível examinar a sua consistência. Cada observação representa o comportamento ou perfil de chamadas de um utilizador durante um determinado dia. Uma observação é considerada um outlier, se pertencer a um cluster de baixa densidade e com uma distância de similaridade superior a dois ou três vezes o valor do desvio padrão em relação à média do cluster. Esta medida permite determinar se uma observação é suficientemente diferente das anteriores.

Em 2016, os autores Yu, Q., Jibin, L., & Jiang, L propuseram uma abordagem baseada no método de séries temporais ARIMA, para detetar anomalias no tráfego de redes de sensores sem fios [30]. A abordagem apresentada, começa por construir um modelo com um período de histórico fixo, de quinze leituras do tráfego. Após a construção do modelo, a previsão é realizada com um passo de cinco seguido do cálculo da sua média ponderada. O uso da média ponderada permite detetar anomalias no tráfego num período maior para não detetar mudanças súbitas de tráfego causadas pelo equipamento. Em cada iteração, o modelo é atualizado com o período de histórico mais recente disponível. Para distinguir o tráfego normal do anormal foi definido um threshold com um desvio até 15%. Nas experiências efetuadas foi alcançada uma precisão de 96% na deteção de anomalias.

Os trabalhos [24] e [29] sugerem que a técnica de clustering adequa-se à segmentação de clientes em telecomunicações, a fim de extrair insights dos vários perfis de consumo. Os trabalhos [26] e [27] mostram que o método ARIMA pode ser a melhor escolha para previsão de séries temporais em telecomunicações, e também em outros contextos, mas têm um grande inconveniente, requerem configuração manual dos seus parâmetros. Além disso, neste trabalho foi importante avaliar dois métodos, o ARIMA e o Prophet, com dados reais de consumo em telecomunicações, e perceber como o modelo pode ser integrado numa aplicação. Quanto aos restantes trabalhos, o [29] sugere o algoritmo Apriori, para análise de relações entre serviços subscritos pelos clientes. Finalmente o trabalho [30] sugere também o método de séries temporais ARIMA, para detetar anomalias de tráfego de consumo em telecomunicações.

4.3. Ferramentas de data mining

Atualmente, diversas ferramentas de data mining estão disponíveis no mercado, cada uma com os seus benefícios e suas limitações. Para se efetuar uma análise das ferramentas que implementam algoritmos nas diversas tarefas de data mining, selecionaram-se quatro ferramentas. O Waikato Environment for Knowledge Analysis (Weka) [31], o H_2O [32], o Python [33] [34] [35], e ainda uma ferramenta líder no quadrante mágico da Gartner de 2017 [36], o Rapid Miner [37]. De maneira a fazer uma comparação entre estas ferramentas definiu-se o seguinte conjunto de critérios:

- Open source, restrição imposta no desenvolvimento deste projeto;
- Suporte, incluí documentação, ampla comunidade e pequenos tutoriais de iniciação;
- Fornecer funcionalidades para análise exploratória de dados;
- Fornecer todas as técnicas de data mining (algoritmos de classificação, regressão, clustering, regras de associação, seleção de atributos);
- Fornecer métodos de séries temporais;
- Integração com fontes de dados, nomeadamente ficheiros e base de dados;
- Capacidade de lidar com grande quantidade de dados;
- Fornecer API para integração com linguagens de programação.

A Tabela 8 resume a comparação efetuada às ferramentas selecionadas usando os critérios definidos.

Critério	Weka	H_2O	Python	Rapid Miner
Open source	Sim	Sim	Sim	Sim
Suporte	Não	Sim	Sim	Sim
Análise exploratória	Sim	Não	Sim	Sim
Todas as técnicas de data mining	Sim	Não	Sim	Sim
Métodos de séries temporais	Sim	Não	Sim	Sim
Integração com fontes de dados	Sim	Sim	Sim	Sim
Lida com grande quantidade de dados	Não	Sim	Sim	Não
API para integração com linguagens programação	Sim	Sim	Sim	Sim

Tabela 8 - Comparação de ferramentas de data mining

Após a análise das ferramentas e tendo em consideração o conjunto de critérios de escolha apresentados acima, conclui-se que o Weka e o H_2O não são escolhas adequadas. O Weka é uma ferramenta pouco flexível na criação de aplicações. O H_2O é uma ferramenta recente, disponível desde 2014, que está em desenvolvimento e neste momento ainda não possui todas as técnicas de data mining nem métodos de séries temporais. Entre as restantes ferramentas, o Python possui uma ampla comunidade com diversas bibliotecas para visualização gráfica de dados e maior flexibilidade na integração e otimização de um modelo em uma aplicação em produção. Além disso o Python permite lidar com grande quantidade de dados com recurso à nova framework chamada Dask [38]. Esta framework fornece desempenho e escalabilidade horizontal no processamento, construção e previsão de modelos em grandes conjuntos de dados.

Capítulo 5

Especificação de Requisitos

A plataforma desenvolvida com modelos preditivos e descritivos, consiste num back-end disponibilizado na forma de uma API REST. A plataforma recolhe e analisa dados provenientes dos produtos existentes na empresa, com o objetivo de extrair conhecimento. Pretende-se que a aplicação tenha a capacidade de ser integrada e fornecer dados num formato universal e reconhecível pelos produtos existentes. O utilizador final vai ser um analista, com funções diferenciadas nas áreas de marketing, gestão de produtos e operações. Este capítulo descreve os requisitos funcionais e não funcionais, restrições técnicas e de negócio da plataforma.

5.1. Requisitos funcionais

Os requisitos funcionais foram identificados e definidos a partir do plano de desenvolvimento dos produtos da empresa (ou seja, já estavam planeados para este estágio) e dos sistemas de suporte à decisão estudados e apresentados no Estado da Arte. Estes foram descritos nesta secção utilizando a seguinte estrutura: título, descrição da interação entre a plataforma e o exterior, e importância para o operador. Os requisitos levantados foram priorizados utilizando o método MoSCoW [39]. Este método prioriza os requisitos numa das seguintes quatro categorias:

- Must: o requisito deve estar implementado na plataforma;
- Should: o requisito se possível deve estar implementado na plataforma;
- Could: requisito desejável, mas não necessário nesta fase;
- Won't: requisito não precisa de ser implementado nesta fase, mas importante no futuro.

ID: RF1 - Perfil de clientes
Título: determinar o perfil dos clientes que aderiram e não a uma campanha
Descrição: a API REST deve receber os seguintes parâmetros de entrada:
- Uma campanha existente
- O número de grupos (ou de perfis) de clientes no intervalo de três a cinco
Após receber e validar o pedido, deve determinar os perfis dos clientes e devolver um conjunto de insights e métricas para cada grupo obtido. Os insights dos perfis dos clientes devem ser representados pelos seus atributos (ou features) relevantes, de consumo e comportamento no momento do incentivo à campanha. Relativamente às métricas, para cada grupo, deve devolver a percentagem de clientes aderentes e o valor resultante do cálculo do lift over grupo de controlo
Importância para o operador: fornecer insights durante a campanha para permitir ao operador ajustar e identificar com maior precisão o seu público-alvo
Prioridade: Must

ID: RF2 - Previsão de consumo
Título: previsão do consumo de recargas, SMS, e dados de internet
Descrição: a API REST deve receber como parâmetro de entrada o tipo de consumo, de entre três opções possíveis (número de recargas usadas nos serviços de telecomunicações,

número de eventos de SMS ou volume de dados de internet). Após receber e validar o pedido, deve prever os três meses futuros a partir do histórico e devolver os valores previstos
Importância para o operador: permite otimizar ou eventualmente reforçar os recursos de rede e adequar os canais de recarga à sua taxa de utilização
Prioridade: Must

ID: RF3 - Anomalias por grupo
Título: deteção de anomalias (desvios súbitos e acentuados no nível de tráfego) no comportamento de consumo de um grupo de clientes
Descrição: a API REST deve receber como parâmetro de entrada, o tipo de consumo. Após receber e validar o pedido, deve ser capaz de detetar anomalias de consumo no grupo de clientes do serviço Pré-Pago. Finalmente deve devolver a informação do consumo real e do previsto. Este último com recurso a um intervalo de confiança para permitir distinguir o consumo anormal
Importância para o operador: permite automatizar a deteção de anomalias de consumo, aumentando a rapidez e eficácia da sua deteção
Prioridade: Should

ID: RF4 - Relações entre serviços subscritos
Título: descoberta de relações nos registos de subscrições de serviços baseada numa análise de market basket
Descrição: a API REST deve receber os seguintes parâmetros de entrada:
- A data com o ano e o mês
- O valor mínimo de suporte no intervalo de 0.6 a 1
- O valor mínimo de confiança no intervalo de 0.6 a 1

Após receber e validar o pedido, deve ser capaz de determinar e devolver um conjunto de relações entre os serviços representadas por regras do tipo $A \rightarrow B$, juntamente com o valor de suporte e confiança
Importância para o operador: permite estudar preferências e criar pacotes de serviços subscritos em conjunto para potenciar o negócio
Prioridade: Should

ID: RF5 - Autenticação
Título: mecanismo de autenticação
Descrição: a API REST deve receber o login e a password de administrador previamente definidos para devolver um token de acesso às funcionalidades da API por um determinado período de tempo
Prioridade: Must

5.2. Requisitos não funcionais

Os requisitos não funcionais definem-se como as características que o sistema deve possuir em adição à sua funcionalidade [40]. Foram identificados e considerados importantes para esta plataforma os seguintes requisitos não funcionais:

- **Interoperabilidade**: traduz a capacidade de dois sistemas trocarem informações entre si [40]. A plataforma deve estar preparada para comunicar e fornecer dados num formato interoperável e universal a qualquer outra aplicação.
- **Desempenho**: é a capacidade do sistema de executar o seu trabalho no tempo especificado [40]. A plataforma deve ser capaz de responder aos pedidos num intervalo de tempo que se considere aceitável.
- **Robustez**: refere-se à capacidade de o sistema continuar em funcionamento na presença de eventos que estão além do seu normal domínio de funcionamento [41]. A plataforma deve fornecer uma API capaz de funcionar corretamente na presença de parâmetros inválidos de entrada e de falhas no acesso à fonte de dados.

As Tabelas 9, 10, 11 e 12 apresentam cenários que ajudam a descrever os requisitos não funcionais descritos acima.

Cenário de interoperabilidade	
Fonte do estímulo	Exterior à plataforma
Estímulo	Um pedido para troca de informação à API
Ambiente	Funcionamento normal
Artefacto	A plataforma
Resposta	A plataforma responde com sucesso ao pedido para troca de informação
Mensuração da resposta	Percentagem de trocas de informação processadas com sucesso

Tabela 9 - Cenário de interoperabilidade

Cenário de desempenho	
Fonte do estímulo	Exterior à plataforma
Estímulo	Chegada de um pedido
Ambiente	Funcionamento normal
Artefacto	A plataforma
Resposta	Pedido processado dentro do tempo esperado
Mensuração da resposta	Pedido respondido com um tempo inferior a sete segundos

Tabela 10 - Cenário de desempenho

Cenário de robustez 1	
Fonte do estímulo	Exterior à plataforma
Estímulo	Chegada de um pedido com pelo menos um parâmetro inválido de entrada
Ambiente	Funcionamento normal
Artefacto	A plataforma
Resposta	A plataforma verifica os parâmetros de entrada e devolve uma mensagem de erro
Mensuração da resposta	Percentagem de pedidos negados pela plataforma

Tabela 11 - Cenário de robustez 1

Cenário de robustez 2	
Fonte do estímulo	Exterior à plataforma
Estímulo	Falha de ligação à fonte de dados
Ambiente	Plataforma em funcionamento normal e fonte de dados indisponível
Artefacto	A plataforma
Resposta	A plataforma deteta a falha, contínua em funcionamento e devolve uma mensagem de erro
Mensuração da resposta	A plataforma não falha

Tabela 12 - Cenário de robustez 2

5.3. Restrições técnicas e de negócio

As restrições técnicas e de negócio são decisões impostas que devem ser atendidas no desenho da arquitetura.

A plataforma deve satisfazer as seguintes restrições técnicas:

- Tecnologias open source: o uso de tecnologias sem custos foi uma restrição imposta pela empresa;
- Estilo arquitetural Representational State Transfer (REST): a arquitetura da plataforma deve seguir esse estilo, de modo a permitir integração e comunicação com outras aplicações.

O desenvolvimento da plataforma deve ter em conta as seguintes restrições de negócio:

- Data limite de desenvolvimento: a versão base da plataforma deve estar terminada a 31 de julho de 2018, data que termina o estágio;
- Confidencialidade dos dados: no início do estágio foi assinado um acordo de confidencialidade sobre o uso de dados reais dos operadores de telecomunicações. O acesso a estes dados, apenas pode ser feito dentro da rede interna da empresa;
- Dados disponibilizados pelos operadores de telecomunicações: os dados utilizados neste trabalho estão sujeitos à aprovação por parte do operador.

Capítulo 6

Arquitetura

A arquitetura de um software define-se como o conjunto de estruturas necessárias ao seu funcionamento, os seus componentes e as relações entre eles. Este capítulo apresenta a arquitetura proposta dividindo-se em duas partes, a primeira com o desenho de alto nível e a segunda com a especificação que descreve detalhadamente os módulos da aplicação.

6.1. Desenho de alto nível

O desenho da arquitetura tem diversas vantagens tais como, mostrar e comunicar o que está desenvolvido aos stakeholders e gerir as suas modificações futuras [40]. Com base no C4 model [42], esta secção apresenta o desenho de alto nível da plataforma, utilizando um conjunto de vistas relevantes que a dividem em múltiplas representações e demostram como a plataforma suporta os requisitos especificados no capítulo anterior. Inclui a escolha de tecnologias e frameworks a usar na implementação.

6.1.1. Diagrama de contexto

A Figura 5 mostra onde a plataforma desenvolvida neste projeto se situa e os produtos existentes com os quais interage.

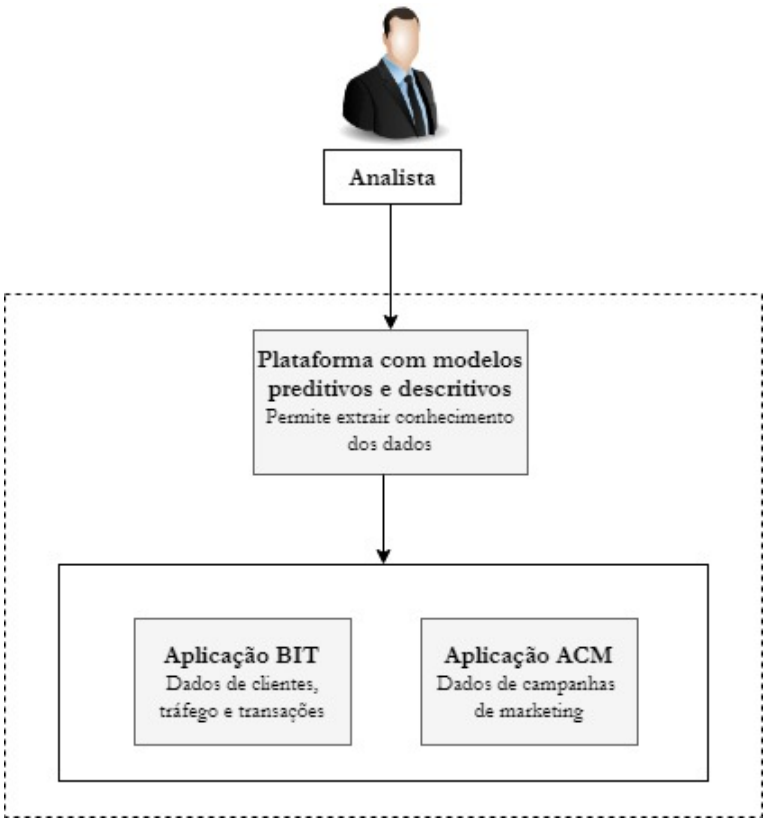


Figura 5 - Diagrama de contexto

A plataforma desenvolvida neste projeto interage com os produtos existentes atualmente na empresa, são eles o Business Intelligence Tools (BIT) e o Active Campaign Manager (ACM). Além destas aplicações, também vai fornecer informação de suporte à decisão ao utilizador final definido como analista.

6.1.2. Vista de componentes

Uma vez que já se sabe onde a plataforma se encaixa no ecossistema da empresa, esta secção mostra uma visão de alto nível dos componentes que a compõe, e como estes comunicam entre si. A Figura 6 apresenta o diagrama de vista de componentes.

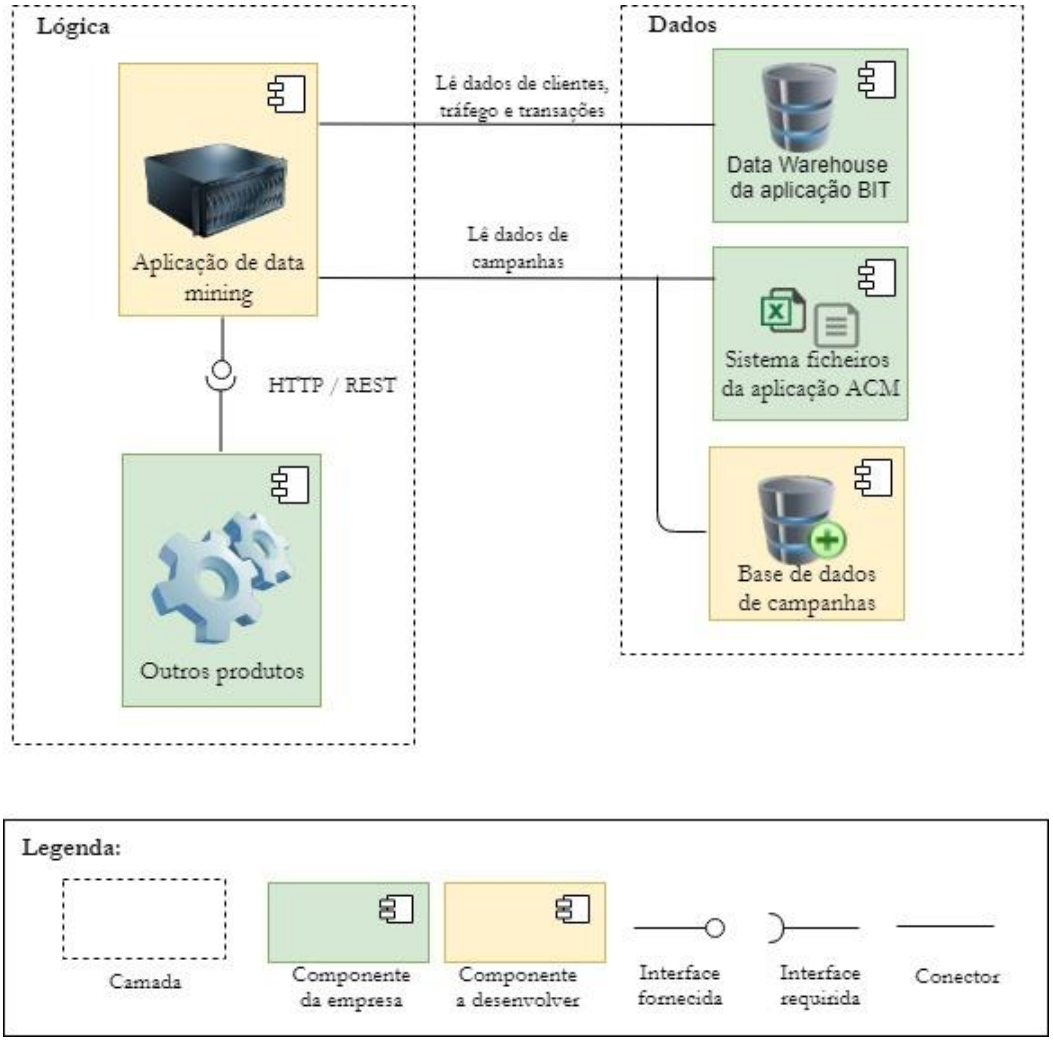


Figura 6 - Vista de componentes

A **camada de dados** é responsável por fornecer dados à aplicação de data mining e por auxiliar no processo de preparação dos dados provenientes dos ficheiros. Constituída pelas fontes de dados existentes, o Data Warehouse da aplicação BIT, os ficheiros de dados exportados da aplicação ACM, e pela base de dados de campanhas que vai conter os dados dos ficheiros.
O Data Warehouse possui dados limpos e transformados, sobre os quais é aplicado o processo ETL (Extract, Transform, Load) nos dados brutos de um operador de média dimensão. Constituído por três data marts, que possuem indicadores operacionais e de cada cliente, referente ao período de até quatro anos. O data mart de clientes, contém informação sobre os clientes, nomeadamente o saldo e a associação de clientes aos seus serviços subscritos. O data

mart de transações, contém informação sobre as subscrições e recargas. Na tabela de subscrições, cada registo representa um cliente, contendo a data de registo da subscrição, ativação, desativação, mudanças de estado. As tabelas de recarga, contêm agregados à hora, dia e mês por cliente e por grupo de clientes, contendo o número, o montante de recargas e o saldo antes da recarga. O data mart de tráfego, contém a informação agregada à hora, dia e mês por cliente e por grupo de clientes. As tabelas de tráfego, contêm o registo de eventos de chamadas de voz, SMS e dados de internet, nomeadamente o número de chamadas, duração, valor tarifado.
Os ficheiros exportados do ACM possuem dados de campanhas de um operador de média a grande dimensão. A informação dos clientes encontra-se agregada ao mês no momento do incentivo a uma campanha, referente aos seus dados de cartão, comportamento, consumo e sua rentabilidade para o operador. Também, contém informação das transições de estado dos clientes ao longo do ciclo de vida de cada campanha, para que seja possível identificar os clientes que aderiram e não a uma campanha.
A criação da base de dados de campanhas foi criada apenas para estudo e desenvolvimento do primeiro requisito funcional deste projeto com dados estáticos. Além disso, justifica-se pela necessidade de integrar uma grande quantidade de dados provenientes dos ficheiros exportados do ACM, para que estes possam ser selecionados, limpos e preparados para a aplicação de técnicas de data mining. A secção 6.2.1 do relatório descreve o modelo de dados deste componente.

A **camada lógica** é onde se situa a aplicação de data mining. Esta aplicação é responsável por recolher, preparar e analisar os dados provenientes das fontes de dados existentes, a fim de criar modelos que permitam extrair conhecimento. Além disso, recebe e responde aos pedidos provenientes do exterior.

Os componentes apresentados acima comunicam da seguinte maneira:

- A comunicação entre a aplicação de data mining e o exterior, vai ser feita por API REST usando o protocolo Hypertext Transfer Protocol (HTTP). A escolha da tecnologia REST permite satisfazer o requisito não funcional de interoperabilidade e a restrição técnica referente ao uso do estilo arquitetural REST.
- A ligação entre a aplicação de data mining e as bases de dados é feita por conector.

6.1.3. Vista lógica da aplicação

Nesta secção fez-se zoom-in ao componente da aplicação de data mining. A Figura 7 decompõe este componente num conjunto de módulos que o constituí. De seguida para cada módulo descreve-se as suas responsabilidades e como interagem entre si.

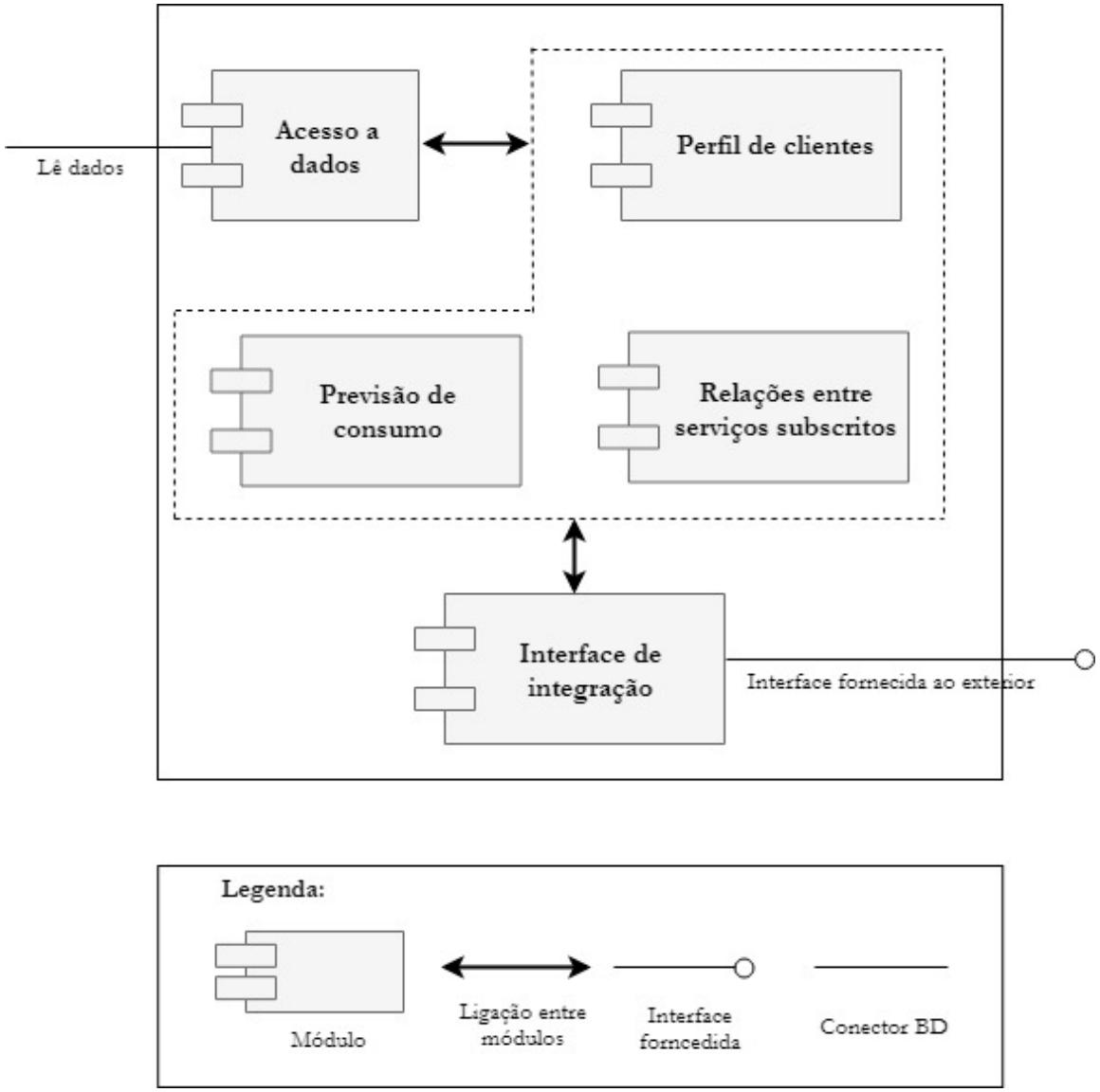


Figura 7 - Vista lógica da aplicação

O módulo de **acesso a dados**, é responsável por conectar e recolher os dados das fontes de dados existentes. Também tem a função de integrar e carregar os dados dos ficheiros exportados do ACM na base de dados de campanhas. Os módulos de **perfil de clientes**, **previsão de consumo** e **relações entre serviços subscritos** correspondem aos requisitos funcionais. Cada um destes módulos é responsável por um requisito funcional, com o seu processo de preparação dos dados e com as suas técnicas de data mining especificas a utilizar. Os modelos dos requisitos são criados em tempo real para responder aos pedidos recebidos, e de seguida eliminados. O módulo da **interface de integração**, fornece um conjunto de endpoints com parâmetros configuráveis através de uma API REST. A integração por API com outras aplicações, minimiza o acoplamento e a dependência entre componentes, necessitando apenas que estes mantenham uma certa consistência entre si [44]. Além de fornecer a API, recebe pedidos do exterior, valida e deteta parâmetros inválidos de entrada.

Comunica com os módulos dos requisitos funcionais com o objetivo de obter a informação de acordo com o pedido recebido e em seguida responde ao pedido. Cada um destes módulos encontra-se detalhado na secção 6.2 do relatório.

6.1.4. Escolha de ferramentas e tecnologias

No desenvolvimento do componente de aplicação de data mining optou-se pelo Python (versão 3.6). O Python é uma ferramenta flexível que disponibiliza um conjunto de bibliotecas para o estudo e análise comprativa dos modelos dos requisitos, e permite a implementação de mecanismos de parametrização dos modelos para integração numa aplicação de suporte à decisão. As principais bibliotecas necessárias ao desenvolvimento desta plataforma foram as seguintes: sklearn [33] com métodos de clustering e de seleção de atributos, mlxtend [34] com regras de associação, e statsmodels [35] com métodos de séries temporais. O módulo de interface de integração, implementou-se com recurso ao Flask [43], framework web usada na implementação de uma API REST com um conjunto de endpoints disponibilizados por HTTP. O JavaScript Object Notation (JSON) é o formato de dados utilizado para troca de informação com o exterior. É um formato de intercâmbio de dados, universal, interoperável e lightweight, constituído por uma coleção de pares chave-valor [44]. No componente de base de dados de campanhas, optou-se por um motor de base de dados PostgreSQL [45]. Este permite auxiliar no processo de integração e preparação dos dados exportados do ACM.

6.2. Especificação detalhada

Esta secção descreve o modelo de dados da base de dados de campanhas e os módulos do componente de aplicação.

6.2.1. Modelo de dados da base de dados de campanhas

Os dados das campanhas exportados da aplicação Active Campaign Manager (ACM) foram integrados e inseridos na base de dados de campanhas, pelo módulo de acesso a dados para que estes possam ser processados. Importa referir que estes dados são estáticos, cujo seu processo de ETL (Extract, Transform, Load) está em desenvolvimento e não faz parte do estágio. A Figura 8 apresenta o modelo de dados deste componente constituído por três tabelas, Cliente, Transição da campanha, e Mapeamento entre a tabela de cliente e transição de estado de campanha.

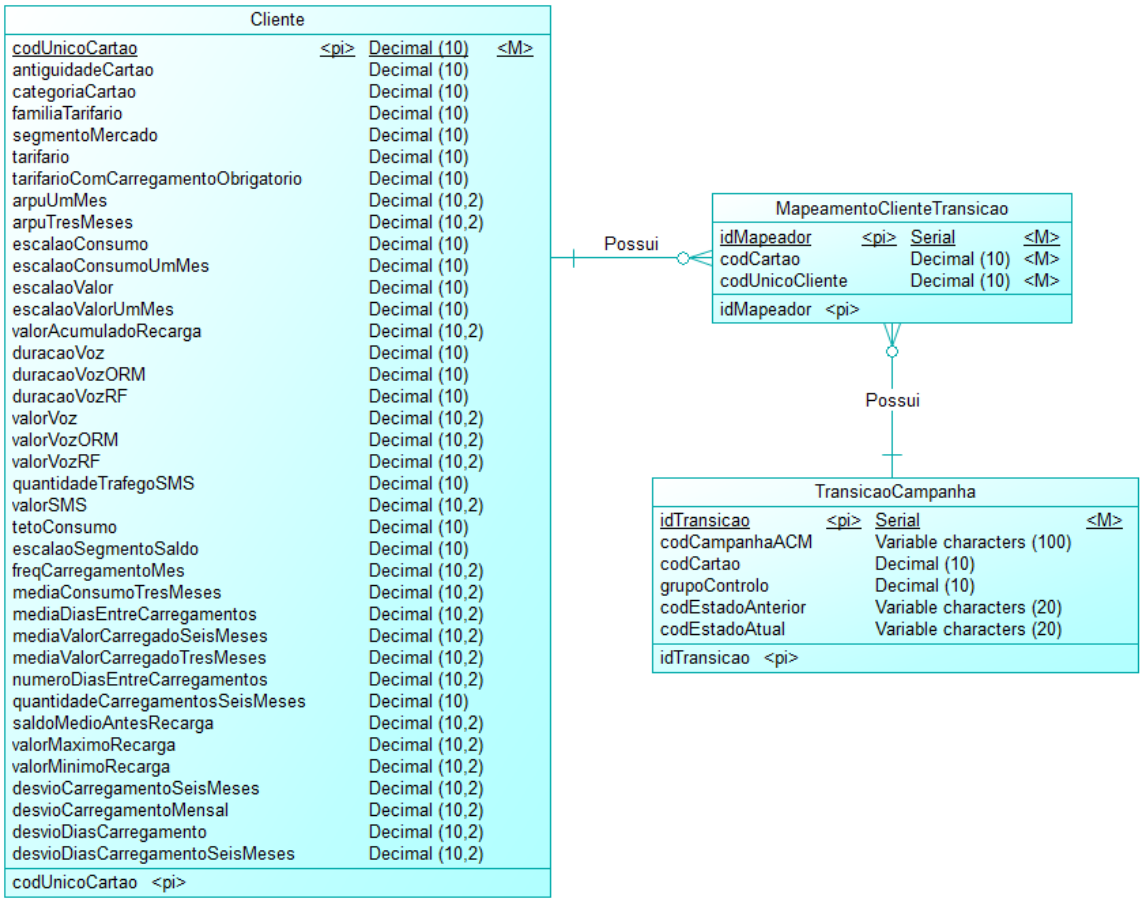


Figura 8 - Modelo de dados da base de dados de campanhas

De seguida descreve-se a informação que cada uma das tabelas do modelo acima contém:

- Cliente: contém um código de identificação e um conjunto de atributos que definem o perfil de cada cliente nas categorias de dados de cartão, consumo, rentabilidade e valor para o operador. Os valores dos atributos dos clientes já se encontram agregados a um, a três ou a seis meses, dependendo do seu tipo.
- TransiçãoCampanha: representa as transições dos clientes no ciclo de vida da campanha, permitindo identificar os clientes que aderiram e não.
- MapeamentoClienteTransição: contém um conjunto de pares de identificadores que permitem ligar os clientes às suas transições de estado na campanha. O atributo

nomeado grupo de controlo, indica se o cliente foi selecionado para pertencer a esse grupo de clientes que não recebe notificação do incentivo.

6.2.2. Módulo de perfil de clientes

Este módulo é responsável pelo primeiro requisito funcional, de determinar o perfil de clientes. Recorreu-se à técnica de clustering, usada no mesmo tipo de problema numa das abordagens analíticas estudadas e apresentadas no capítulo de Estado da Arte [24]. As experiências realizadas com o objetivo de escolher e configurar o modelo deste requisito, encontram-se descritas na secção 7.1 do relatório.
A implementação deste requisito dividiu-se em duas fases. A primeira fase, de segmentação de clientes com o objetivo de extrair insights dos perfis de clientes que aderiram e não a uma campanha. A segunda fase, de cálculo do lift over grupo controlo (GC) em cada um dos grupos (ou clusters) obtidos. Esta métrica relaciona os clientes pertencentes ao público-alvo (PA), incentivados, com os do grupo de controlo (GC), não incentivados, a fim de medir e avaliar o sucesso que a campanha está a ter em cada perfil obtido. A Figura 9 apresenta as etapas de cada uma destas duas fases.

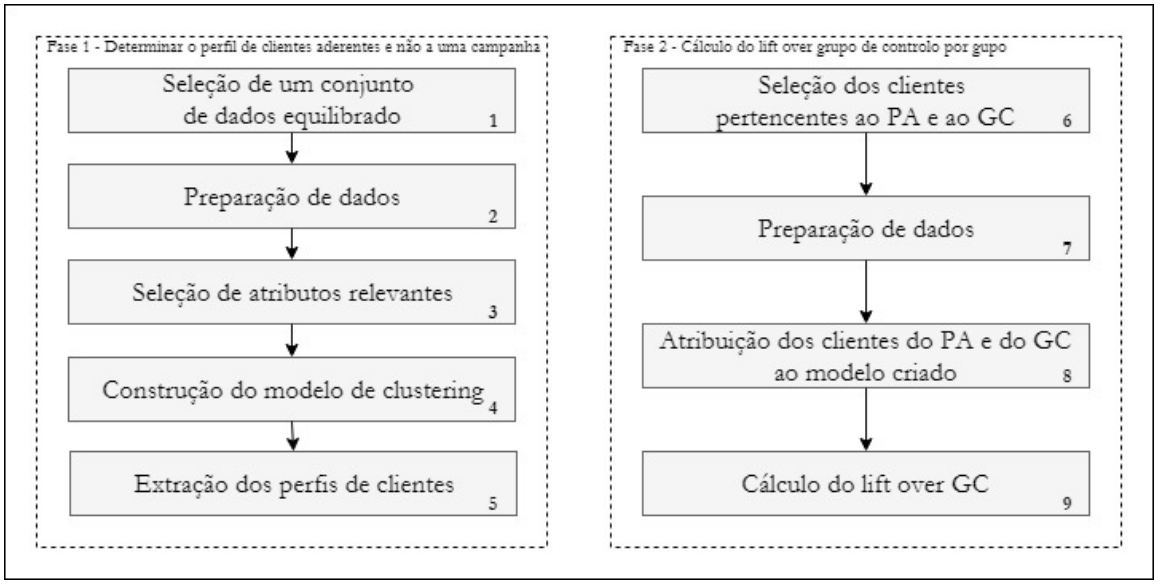


Figura 9 - Diagrama de etapas para determinar o perfil de clientes

A primeira fase determina o perfil de clientes que adere e não a uma campanha, com o objetivo de separar tanto quanto possível os clientes aderentes e não aderentes em grupos diferentes. Constituída pelas seguintes cinco etapas:

1. Seleção de um conjunto de dados equilibrado: começa por selecionar os clientes que aderiram a uma campanha num dado mês na base de dados de campanhas. Dado que o número de clientes aderentes a uma campanha é significativamente inferior ao de não aderentes, aplicou-se a técnica de undersampling que permite melhorar os resultados em problemas com classes desequilibradas segundo [8]. Consiste na seleção aleatória de um número igual de clientes não aderentes para permitir a criação de um conjunto equilibrado.
2. Preparação de dados: nesta etapa são automatizadas as seguintes tarefas:
    - Os valores numéricos dos atributos em falta, são identificados e preenchidos pelo valor médio do atributo. Quanto aos valores nominais, são preenchidos pelo valor mais frequente do atributo;
    - Os atributos nominais como respostas "sim/não", são convertidos em valores numéricos, requisito exigido pelos métodos de clustering;

- Os atributos numéricos com um número de casas decimais superior a dois, são arredondados a duas casas decimais;
- Os valores dos atributos são normalizados pelo método Min-Max [8];
- Finalmente cria a estrutura de dados de entrada no modelo.

A Tabela 13 ilustra a estrutura de dados utilizada nas tarefas de data mining. Cada cliente está associado a um conjunto de atributos numéricos que caracteriza o seu perfil, juntamente com a classe-alvo que identifica se aderiu ou não à campanha.

	Valores dos atributos que definem o perfil do cliente	**Classe**
Cliente 1	7.16, 105, 1.49, 10	Aderiu
…	…	…
Cliente N	21.88, 330, 18.79, 22	Não aderiu

Tabela 13 - Estrutura de dados de entrada no modelo dos perfis

3. Seleção de atributos relevantes de forma dinâmica para cada campanha, divide-se no seguinte conjunto de tarefas:
- Cálculo do ranking da importância dos atributos pelo método filter model, chamado SelectFromModel na biblioteca do Python [19];
- Geração de subconjuntos de atributos de dois em dois atributos (os dois, os quatro, seis primeiros até ao total de atributos);
- Avaliação da precisão (métrica descrita na secção 7.1.2) de vários modelos de clustering K-Means com diferentes subconjuntos de atributos gerados;
- Escolha do número de atributos ótimo no modelo, ou seja, que maximiza a precisão;
- Remoção dos atributos irrelevantes.

4. Construção do modelo de clustering para um determinado número de grupos (ou de perfis) recebido como parâmetro de entrada a partir da interface de integração.
5. Extração dos perfis de clientes: consiste no cálculo dos valores do centro de cada grupo, que resumem e fornecem insights do perfil dos clientes.

Após a primeira fase inicia-se a segunda fase, do cálculo do lift over GC por grupo, usando a seguinte fórmula fornecida pela empresa:

$$\text{Lift over GC (\%)} = \frac{\text{Taxa de adesão do PA} - \text{Taxa de adesão do GC}}{\text{Taxa de adesão do GC}}$$

A métrica de lift over GC relaciona a taxa de adesão dos clientes do público-alvo (PA), incentivados, com a taxa de adesão dos clientes do grupo de controlo (GC), não incentivados que poderiam aderir, a fim de medir e avaliar o sucesso da campanha. Constituída pelas seguintes etapas:

6. Seleção dos clientes pertencentes ao PA e ao GC: nesta etapa seleciona o conjunto todo de clientes pertencentes ao PA e também dos pertencentes ao GC.
7. Preparação de dados: aplica as mesmas transformações descritas na segunda etapa e remove os atributos não selecionados na terceira etapa.
8. Atribuição dos clientes do PA e do GC ao modelo criado: consiste na atribuição ao modelo construído na quarta etapa de todos os clientes pertencentes ao PA, e dos pertencentes ao GC, ambos de forma independente.
9. Cálculo do lift over GC: começa por calcular a taxa de adesão por grupo para os clientes pertencentes PA e para os pertencentes ao GC. Após o cálculo da taxa de adesão destes dois conjuntos de clientes por grupo, o próximo passo é o cálculo do lift over GC usando a fórmula apresentada acima, para cada grupo obtido e no conjunto total de clientes incentivados.

No final este módulo devolve insights do perfil de clientes que adere e não a uma campanha, representados pelo valor médio dos atributos de todos os elementos pertencentes a um grupo (ou cluster). Devolve também o valor percentual de clientes aderentes e o valor do lift over GC ambos por grupo, e o lift over GC no conjunto total de clientes que foram incentivados.

6.2.3. Módulo de previsão de consumo

O presente módulo é responsável por fazer uma previsão aproximada do número de recargas usadas nos serviços de telecomunicações, eventos de SMS ou volume de dados de internet. Na previsão utilizou-se o método de séries temporais Auto-Regressive Integrated Moving Average (ARIMA) sazonal. Esta escolha teve em conta dois motivos. O primeiro, as experiências realizadas e apresentadas na secção 7.2 do relatório, deram vantagem a este método. Em segundo, o ARIMA é um método popular e usado em várias abordagens de previsão em telecomunicações [26] [30]. O diagrama da Figura 10 apresenta as etapas de implementação deste requisito por este módulo.

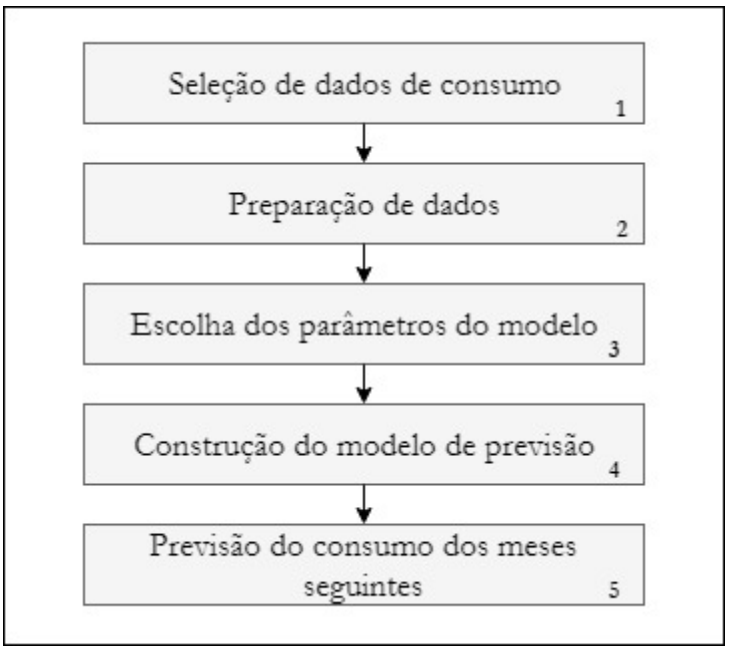


Figura 10 - Diagrama de etapas para previsão de consumo

1. Seleção de dados, extração e agregação por mês de um período de histórico de quatro anos do Data Warehouse da aplicação Business Intelligence Tools (BIT), relativo aos dados que se pretende prever.
2. Preparação de dados, subdivide-se em três tarefas. A primeira converte o formato da data original no formato exigido pelo modelo. A segunda constrói a estrutura de dados de entrada no modelo de previsão com duas colunas, como ilustra a Tabela 14.

Data	Valor do tipo de consumo
2015 - 01 - 01	1000
…	…
2017 - 12 - 01	2000

Tabela 14 - Estrutura de dados de entrada no modelo de previsão

A terceira aplica uma transformação logarítmica exigida pelo método para atenuar a tendência da série temporal [13].
3. Escolha dos parâmetros do modelo: testa e avalia diversos modelos a fim de escolher o modelo com o menor erro de previsão nos dados de histórico.

Nesta etapa, todas as combinações dos parâmetros p, q e d são geradas. Os primeiros dois podem ser zero, um ou dois e o parâmetro d pode ser zero ou um. O intervalo de valores definido teve em conta os testes de parametrização manual, apresentados e discutidos na secção 7.2 do relatório e também uma certa flexibilidade para adaptar a dados futuros. O parâmetro de sazonalidade está definido com o valor de doze, correspondente ao período sazonal. Após a geração das combinações de parâmetros, cada modelo instanciado é testado com valores históricos. A estacionariedade da série temporal é um requisito para a aplicação do método ARIMA. Assim sendo, quando o modelo é construído com uma série não estacionária, a combinação de parâmetros testada nesta iteração é descartada, e o teste avança para a combinação de parâmetros seguinte. A abordagem apresentada nesta etapa, permite a integração do modelo na aplicação, ajustando automaticamente os parâmetros a mudanças no comportamento da série, sem a necessidade de configuração manual.
4. Construção do modelo de previsão: após a escolha dos parâmetros, o modelo de previsão é construído com um período de histórico de três anos recente (ou seja, do mês passado ou anterior até três anos atrás) e com os parâmetros identificados na etapa anterior.
5. Previsão do consumo dos meses seguintes: inicia-se após a construção do modelo escolhido, e consiste em prever os três meses seguintes. Os valores previstos são convertidos de volta à escala dos valores originais, com a realização de uma transformação exponencial (operação inversa da transformação logarítmica). Finalmente os valores previstos são devolvidos ao módulo da API.

6.2.4. Módulo de relações entre serviços subscritos

Responsável por extrair regras de associação nos serviços que estão subscritos pelos clientes numa data selecionada. Recorreu-se ao algoritmo de associação Apriori, usado numa abordagem analítica apresentada no capítulo de Estado da Arte [28]. A Figura 11 apresenta o conjunto de etapas deste requisito.

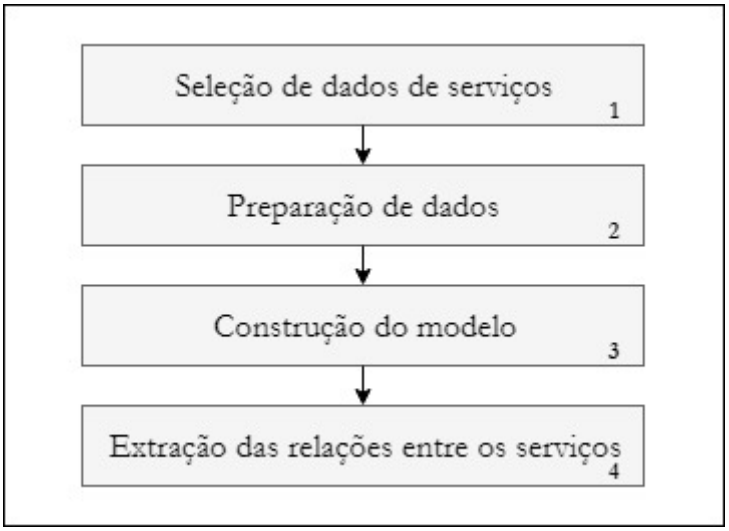


Figura 11 - Diagrama de etapas para extrair relações entre serviços subscritos

1. Seleção de dados de serviços: consiste em extrair aleatoriamente dez mil clientes do tipo Pré-Pago, seguido dos seus serviços subscritos de telemóvel, de internet no estado ativo e também serviços complementares no Data Warehouse da aplicação Business Intelligence Tools (BIT).
2. Preparação de dados: os indentificadores dos serviços de cada cliente selecionados na base de dados, são convertidos para o nome do serviço por cruzamento de chaves primárias. Após este passo é criada a estrutura de dados de entrada no modelo como ilustra a Tabela 15, cada linha contém a lista de serviços subscritos por um cliente.

	Serviços subscritos
Cliente 1	Serviço A, Serviço B
…	…
Cliente N	Serviço A, Serviço C

Tabela 15 - Estrutura de dados de entrada no modelo

3. Construção do modelo: aplica o algoritmo Apriori de associação parametrizado com o valor de suporte e confiança recebido por parâmetro de entrada. O valor do lift está definido para ser maior ou igual a um segundo a referência [8].
4. Extração das relações entre os serviços: após a etapa anterior, as regras de relacionamento entre os serviços são geradas e devolvidas as que possuírem um valor de confiança maior ou igual ao recebido por parâmetro.

6.2.5. Módulo de interface de integração

Responsável por fornecer uma API REST com quatro endpoints, um para autenticação e os restantes para as funcionalidades.

6.2.5.1. Endpoint de autenticação

Com o objetivo de proteger os recursos da API de acessos não autorizados, recorreu-se ao protocolo de autenticação OAuth 2.0 utilizando o JSON web token do Python [46]. Este mecanismo permite dar acesso aos recursos fornecidos pela API a outra aplicação externa, através da atribuição de um token de acesso (código de hash que associa o login e password do utilizador ao timestamp).
A API possui o seguinte endpoint de autenticação:

- Endpoint: /login
- Método HTTP: POST

Recebe como parâmetro de entrada o login e a password de administrador previamente definidos e devolve o token de acesso às funcionalidades. O token deve ser usado pela aplicação externa que pretende aceder a recursos, no cabeçalho HTTP de cada pedido. Sempre que a aplicação recebe um pedido a recursos protegidos, o token é validado. No caso de este ser válido o pedido é processado, caso contrário uma mensagem de erro com o código 401 é enviada.

6.2.5.2. Endpoints das funcionalidades

A API devolve uma mensagem constituída por objetos JSON com duas secções:

- Meta: contém o código HTTP de três dígitos. O código enviado é 200 se tudo funcionou corretamente, 400 se receber um pedido inválido (por exemplos argumentos inválidos), 401 se receber um pedido de acesso a recursos não autorizado, 500 se ocorrer um erro interno (por exemplo falha no acesso à base de dados).
- Response: contém um conjunto de objetos contendo os resultados específicos de cada pedido. Os objetos que constituem a response, podem ser do tipo number (número), string (cadeia de carateres), list (lista) ou dict (dicionário).

O endpoint de perfil de clientes, devolve informação do perfil de clientes que aderiu e não aderiu a uma campanha

- Endpoint: /perfil/{campanha}/{nGrupos}
- Método HTTP: GET
- Autenticação: requerida com OAuth 2.0

A Tabela 16 apresenta os parâmetros de entrada recebidos e de saída devolvidos pelo método.

Parâmetros	Nome	Tipo	Descrição
Entrada	campanha	String	O nome da campanha
	nGrupos	String	O número de grupos (ou de perfis) de clientes. O valor deve estar entre 3 e 5
	tokenAcesso	String	A chave OAuth do consumidor
Resposta	perfis	Dict	Contém uma lista por grupo, contendo os valores médios dos respetivos atributos
	percentagemClientesIncentivadosAderentesGrupo	List	Contém a percentagem de clientes aderentes por grupo
	liftGrupo	List	Contém o cálculo do lift over grupo de controlo por grupo
	liftGlobal	Number	O cálculo do lift over grupo de controlo no conjunto total

Tabela 16 - Parâmetros do endpoint de perfil de clientes

A Figura 12 apresenta um exemplo da mensagem em JSON devolvida pelo método.


```
{
    'meta': {
        'status': 200,
        'msg': 'Ok'
    },
    'response': {
        'perfis': {
            'perfil_1': {
                'arpuTresMeses': 5.75,
                'mediaDiasEntreCarregamentos': 34.32,
                'quantidadeCarregamentosSeisMeses': 4.24,
                'saldoMedioAntesRecarga': 1.68
            },
            'perfil_2': {...},
            'perfil_3': {...},
            'perfil_4': {...}
        },
        'percentagemClientesIncentivadosAderentesGrupo': [
            11.2,
            4.8,
            4.37,
            2.93
        ],
        'liftGrupo': [
            12.23,
            104.26,
            124.21,
            -46.73
        ],
        'liftGlobal': 45.82
    }
}
```


Figura 12 - Exemplo do formato JSON devolvido pelo primeiro método

O endpoint de previsão de consumo, devolve os valores previstos a três meses de acordo com o tipo de consumo recebido no pedido.

- Endpoint: /previsao/{tipoConsumo}
- Método HTTP: GET
- Autenticação: requerida com OAuth 2.0

A Tabela 17 apresenta os parâmetros de entrada recebidos e de saída devolvidos pelo método.

Parâmetros	Nome	Tipo	Descrição
Entrada	tipoConsumo	String	Indica o tipo de consumo que se pretende prever. Pode ser o número de recargas, SMS ou volume de dados de internet
	tokenAcesso	String	A chave OAuth do consumidor
Resposta	datasHistorico	List	As datas do histórico utilizado
	valoresHistoricos	List	Os valores do histórico utilizado
	datasPrevistas	List	As datas dos três meses previstos
	valoresPrevistos	List	Os valores previstos para os próximos três meses
	erroPrevisao	String	O menor erro de previsão obtido nos testes de escolha dos parâmetros do modelo com dados históricos

Tabela 17 - Parâmetros do endpoint de previsão de consumo

A Figura 13 apresenta um exemplo da mensagem em JSON devolvida pelo método.


```
{
   'meta': {
      'status': 200,
      'msg': 'Ok'
   },
   'response': {
      'datasHistorico': [
         '2015-01-01',
         '2015-02-01',
         ...,
         '2017-11-01',
         '2017-12-01'
      ],
      'valoresHistorico': [
         1219291,
         1095724,
         ...,
         715025,
         809964
      ],
      'datasPrevistas': [
         '2018-01-01',
         '2018-02-01',
         '2018-03-01'
      ],
      'valoresPrevistos': [
         730300,
         680607,
         745642
      ],
      'erroPrevisao': 'RMSE = 36942.77 , MAPE = 3.71'
   }
}
```


Figura 13 - Exemplo do formato JSON devolvido pelo segundo método

O endpoint de relações entre serviços subscritos, devolve um conjunto de regras de acordo com o grau de confiança recebido por parâmetro.

- Endpoint: /relacoesServicos/{ano}/{mês}/{suporte}/{confiança}
- Método HTTP: GET
- Autenticação: requerida com OAuth 2.0

A Tabela 18 apresenta os parâmetros de entrada recebidos e de saída devolvidos pelo método.

Parâmetros	Nome	Tipo	Descrição
Entrada	ano	String	O ano em valor numérico
	mês	String	O mês em valor numérico
	suporte (de A → B)	String	Indica a percentagem de clientes que possui os serviços A e B. O seu valor deve estar entre 0.6 a 1.0
	confiança (de A → B)	String	Indica a percentagem de clientes que subscreveu o serviço A, que também subscreveu o serviço B. O seu valor deve estar entre 0.6 a 1.0
	tokenAcesso	String	A chave OAuth do consumidor
Resposta	regras (do tipo A → B)	Dict	Conjunto de regras, cada uma com quatro atributos, o antecedente, o consequente, o valor de suporte e de confiança

Tabela 18 - Parâmetros do endpoint de relações entre serviços subscritos

A Figura 14 apresenta um exemplo da mensagem JSON devolvida pelo método.


```
{
    'meta': {
        'status': 200,
        'msg': 'ok'
    },
    'response': {
        'regra_1': {
            'antecedente': [
                'Serviço A'
            ],
            'consequente': [
                'Serviço B'
            ],
            'suporte': 0.94,
            'confianca': 0.79
        },
        'regra_2': {
            'antecedente': [
                'Serviço C'
            ],
            'consequente': [
                'Serviço B'
            ],
            'suporte': 0.93,
            'confianca': 0.81
        },
        'regra_3': {
            'antecedente': [
                'Serviço B',
                'Serviço C'
            ],
            'consequente': [
                'Serviço A'
            ],
            'suporte': 0.79,
            'confianca': 0.95
        },
    }
}
```


Figura 14 - Exemplo do formato JSON devolvido pelo terceiro método

Capítulo 7

Experimentação e Desenvolvimento de Modelos

Este capítulo apresenta o estudo e análise comparativa de modelos dos requisitos funcionais em diferentes configurações e conjuntos de dados, com o objetivo de encontrar o modelo que melhor se adapta ao problema. No estudo de modelos seguiu-se a metodologia Cross-Industry Standard Process for Data Mining (CRISP-DM) [3], e teve-se como ponto de partida as abordagens estudadas e apresentadas no capítulo de Estado da Arte.

7.1. Perfil de clientes

Pretende-se construir um modelo que determine o perfil dos clientes que aderiu e não a uma campanha para um número de grupos (ou clusters) fixo de três a cinco. Um grupo (ou cluster) é constituído por clientes com atributos semelhantes dentro do mesmo agrupamento. No desenvolvimento de modelos avaliou-se a seleção de atributos com maior importância, a precisão de dois métodos de clustering e verificou-se se o modelo é generalizável utilizando quatro campanhas.

7.1.1. Abordagem

Como referido anteriormente, recorreu-se à técnica de clustering para determinar o perfil dos clientes que aderiu e não a uma campanha. Esta técnica foi usada num problema semelhante, numa das abordagens analíticas estudadas e apresentadas no capítulo de Estado da Arte [24]. As várias experiências aos modelos deste requisito, pretendem avaliar qual a configuração que melhor separa os clientes aderentes e não a uma campanha em diferentes grupos.

7.1.2. Métricas de avaliação de resultados

As métricas usadas na avaliação dos modelos foram a precisão e o coeficiente de Silhouette. A precisão indica a capacidade de o modelo conseguir separar os clientes que aderiram e não à campanha em diferentes grupos. O cálculo desta métrica ilustra-se no seguinte exemplo com os dados da Tabela 19.

Grupo	Nº de clientes	Nº de clientes aderentes	Nº de clientes não aderentes	Classificação do grupo
Grupo 1	40	30	10	Aderente
Grupo 2	30	20	10	Aderente
Grupo 3	20	5	15	Não aderente
Grupo 4	10	0	10	Não aderente

Tabela 19 - Dados de exemplo para ilustrar o cálculo da métrica de precisão

Precisão = (precisão nos grupos aderentes * % de clientes nos grupos de aderentes) +
(precisão nos grupos não aderentes * % de clientes nos grupos de não aderentes)
= ((30 + 20) / (40 + 30) * 0.7) + ((15 + 10) / (20 + 10) * 0.3)
= 0.75 * 100%
= 75%

O coeficiente de Silhouette [8] foi usado para medir a qualidade dos grupos (ou clusters) alcançados, após a realização de experiências que permitiram otimizar a métrica de precisão.

Este coeficiente mede a distância de separação entre os grupos, indicando a proximidade de cada elemento atribuído num dos grupos, em relação aos seus grupos vizinhos. O seu valor varia no intervalo de [-1, 1], em que o coeficiente de 1 indica que o elemento está longe dos grupos vizinhos, o coeficiente de 0 indica que o elemento está no limite entre dois grupos, e o coeficiente de -1 indica que o elemento está perto de elementos em outro grupo vizinho, podendo ter sido atribuído ao grupo errado.

7.1.3. Setup experimental

A Tabela 20 mostra o número de clientes existente nos dados de cada uma das quatro campanhas usadas nas experiências. O universo de teste corresponde aos dados equilibrados, ou seja, metade de clientes que aderiram e a outra metade por clientes que não aderiam. O universo total corresponde aos dados com todos os clientes incentivados à campanha. Os atributos dos clientes estão especificados no modelo de dados da secção 6.2.1.

Campanha	Universo de teste (equilibrado)	Universo total (incentivados)
15	14.058	122.028
87	2.514	123.340
28	1.336	72.986
75	1.246	13.106

Tabela 20 - Número de clientes no conjunto de dados de cada campanha

O processo de seleção, preparação de dados, escolha de atributos está automatizado pelo módulo de determinar o perfil de clientes, descrito na secção 6.2.2 deste relatório.

Uma vez que os clientes da classe maioritária são escolhidos aleatoriamente (universo de teste, com classes equilibradas), em cada experiência fez-se uma série de dez testes, e calculou-se a média e o desvio padrão das medições.

7.1.4. Resultado

A título de exemplo a Figura 15 apresenta graficamente os perfis dos clientes representados pelos valores médios dos atributos relevantes normalizados (para facilitar a sua visualização), para quatro grupos na campanha 15. Cada perfil está representado por uma linha de cor diferente. Na legenda do lado direito do gráfico indica para cada grupo, o valor percentual da taxa de clientes aderentes e o lift over GC (%).

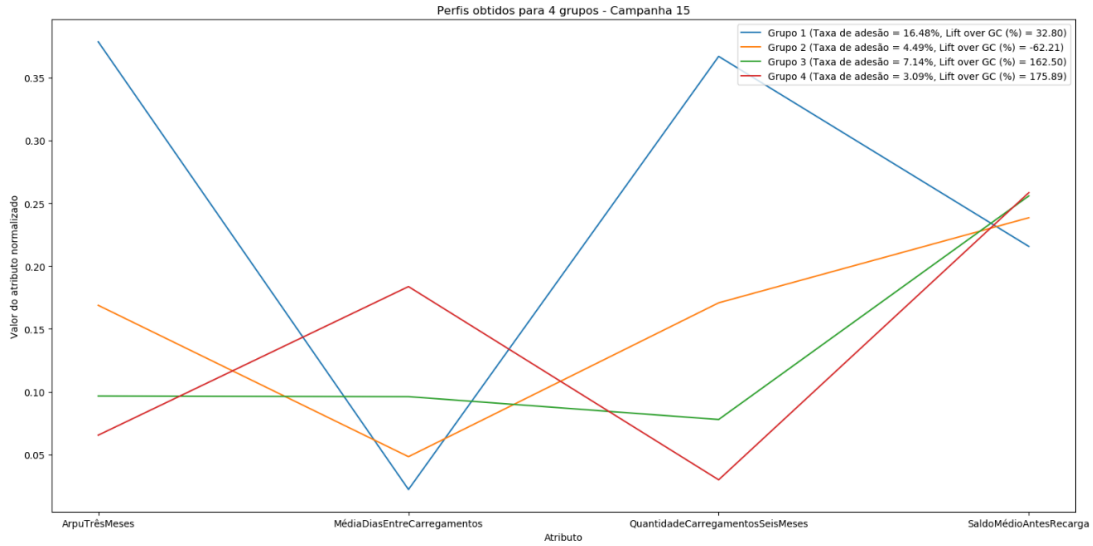


Figura 15 - Gráfico de perfis de clientes obtidos na campanha 15 para quatro grupos

Grupo	1	2	3	4
Atributos de perfil de cliente				
ARPU a três meses	62.58	24.51	11.41	5.76
Média de dias entre carregamentos	5.02	9.74	18.39	34.25
Quantidade de carregamentos a seis meses	41.0	19.61	9.49	4.26
Saldo médio antes da recarga	1.68	2.35	3.01	3.08
Métricas da campanha				
Taxa de adesão (universo de teste)	88.08%	75.14%	51.66%	35.26%
Taxa de adesão (universo total)	16.48%	4.49%	7.14%	3.09%
Lift over GC no universo total (global) = 45.82%				
Lift over GC	32.8%	-62.21%	162.5%	175.89%

Tabela 21 - Perfis de clientes e métricas para a campanha 15 para quatro grupos

A partir dos grupos descritos no gráfico da Figura 19 e Tabela 21 consegue-se extrair a seguinte informação:

- Os perfis obtidos ficaram bem separados entre si, variando na sua forma que sugere que os clientes têm um diferente comportamento de consumo.
- O grupo 1 representa o perfil de clientes mais aderente à campanha com uma taxa de adesão de 16.48%. Neste perfil os clientes possuem um elevado valor de Average Revenue Per User (ARPU), de 62.58 euros, e também efetuam uma quantidade elevada de carregamentos, 41 a cada seis meses.
- O grupo 2 representa os clientes com valores de consumo relativamente altos.
- O grupo 3 representa o perfil com uma taxa de adesão média à campanha. Estes clientes têm valores medianos nos atributos de ARPU, média de dias entre carregamentos, quantidade de carregamentos nos últimos seis meses e saldo médio antes da recarga.
- O grupo 4 representa o perfil de clientes menos aderente à campanha com uma taxa de adesão de 3.09%. Neste perfil os clientes possuem um baixo valor de ARPU, de 5.76 euros, e efetuam uma baixa quantidade de carregamentos, cerca de 4 a cada seis meses. Além disso estes clientes possuem o maior saldo no seu cartão antes de efetuarem um carregamento.
- Dado que o valor do lift over GC global (universo total) é de 45.82%, a partir dos grupos obtidos verifica-se que os grupos 3 e 4 obtiveram um lift over GC bastante superior ao global, de 162.5% e 175.89% respetivamente. Isto significa que estes dois grupos representam os clientes que mudaram mais o seu comportamento fase ao grupo de controlo, ou seja, os clientes dependentes do incentivo. Os restantes grupos representam os clientes independentes do incentivo, ou seja, que poderiam aderir à campanha sem a necessidade de gastar recursos no seu incentivo.

Os resultados das restantes campanhas encontram-se no anexo B.1.5.

7.1.5. Experiências e avaliação de resultados

Nas experiências realizadas comparou-se a precisão do modelo para três, quatro e cinco grupos variando o subconjunto de atributos selecionados, os métodos de clustering e a função de distância que calcula a similaridade entre elementos. Começou-se por comparar a precisão

do modelo com todos os atributos e com subconjuntos de atributos relevantes (ou de maior importância). Os atributos relevantes foram selecionados de duas formas. A primeira com um threshold definido pelo método. A segunda por teste de subconjuntos de atributos, ou seja, os dois, quatro, seis primeiros do ranking ordenado pela sua importância. Após a seleção dos atributos, experimentaram-se dois métodos de clustering e duas funções de distância. Finalmente fez-se a análise de Silhouette para verificação da separação dos grupos obtidos entre si. As secções seguintes apresentam e analisam os resultados das experiências realizadas.

7.1.5.1. Experimentação com o algoritmo de seleção de atributos relevantes

Nesta experiência usou-se o método supervisionado de filter model, chamado SelectFromModel na biblioteca do Python [19] como candidato à seleção e cálculo do ranking dos atributos de maior importância em relação à classe-alvo (aderiu ou não aderiu). A Figura 16 apresenta graficamente a importância de cada atributo em relação à classe-alvo, nas quatro campanhas selecionadas.

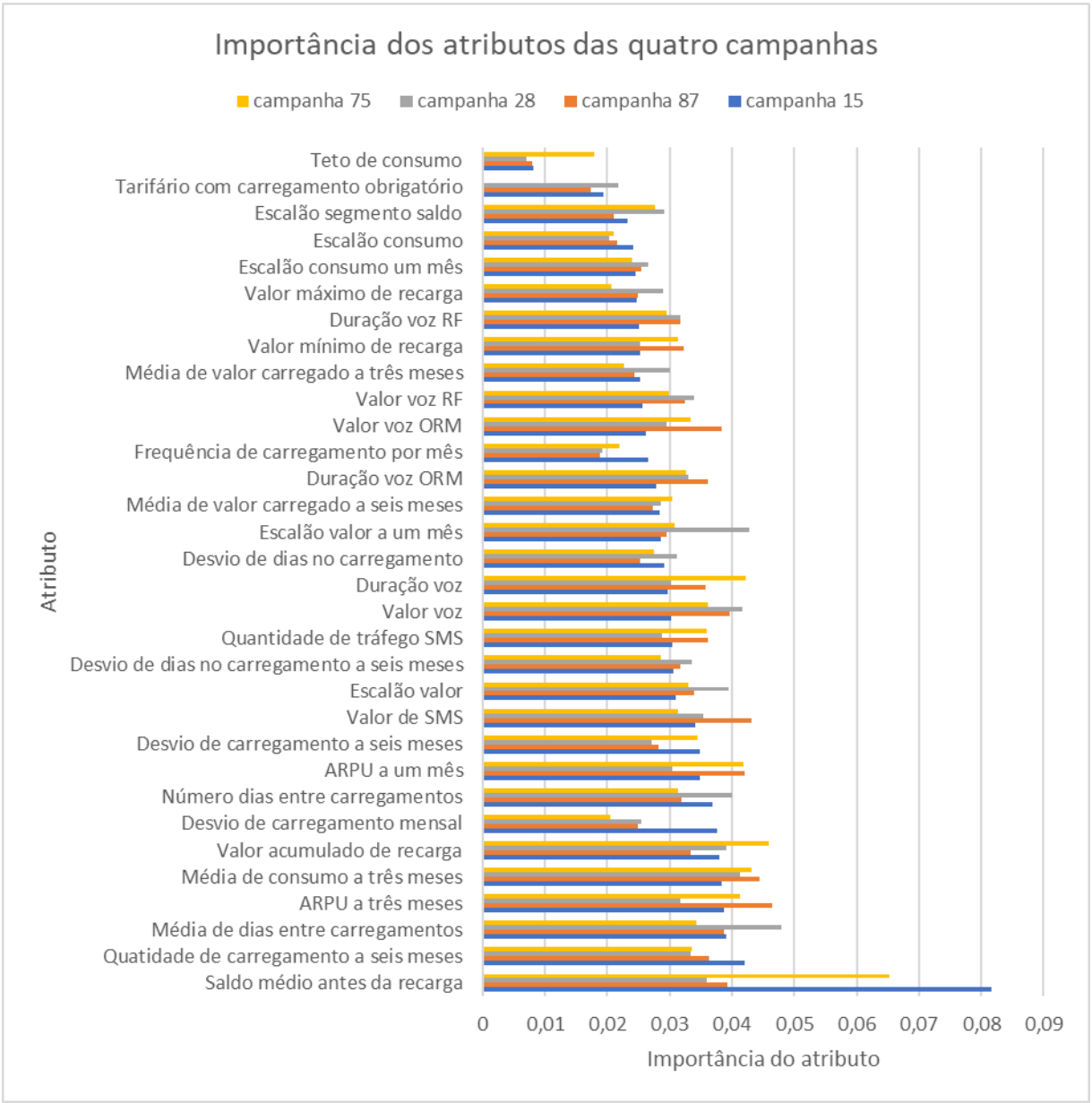


Figura 16 - Importância dos atributos em relação à classe-alvo de cada campanha

Os resultados mostram que a importância dos atributos varia de campanha para campanha. Deste modo a seleção de atributos foi feita de forma dinâmica para cada campanha de dois modos distintos, com o objetivo de avaliar qual deles consegue um melhor resultado.
O primeiro modo, através de um threshold definido internamente no método que define o ponto de corte no ranking dos atributos. Nesta configuração, de um total de 32 atributos, o método selecionou onze atributos para a campanha 15, dezanove atributos para a campanha 87, quinze atributos para a campanha 28, e dezassete atributos para a campanha 75. A lista de atributos selecionados em cada campanha pelo threshold do método encontra-se na Tabela 34 do anexo B.1.1.
O segundo modo baseia-se em procura exaustiva. Após o cálculo do ranking dos atributos ordenado pela sua importância pelo método, são realizados testes de subconjuntos de atributos. Os testes de subconjuntos, foram realizados iterativamente de dois em dois atributos, ou seja, com os dois, quatro, seis primeiros do ranking até ao número total de atributos. Para cada subconjunto é criado um modelo de clustering K-Means e avaliada a sua precisão. O modelo com maior precisão e o seu subconjunto de atributos, é então escolhido em cada campanha. Nesta configuração o número de atributos selecionados foi inferior à configuração anterior. Nas experiências realizadas, os atributos selecionados variaram entre quatro e dezasseis dependendo da campanha e do número de grupos, sendo a combinação com os quatro primeiros atributos a mais frequente. A lista de atributos selecionados em cada campanha nesta configuração encontra-se na Tabela 35 do anexo B.1.1.

7.1.5.2. Validação da importância da seleção de atributos relevantes

Nesta segunda experiência compara-se a precisão do modelo quando se utiliza todos os atributos e cada uma das alternativas que selecionam o subconjunto de atributos relevantes (ou maior importância) por threshold e procura exaustiva. A Figura 17 apresenta a precisão do modelo com cada uma destas três alternativas. Mais detalhe dos resultados das medições encontra-se no anexo B.1.2. Na realização desta experiência utilizou-se o método K-Means com as configurações pré-definidas na sua biblioteca [47].

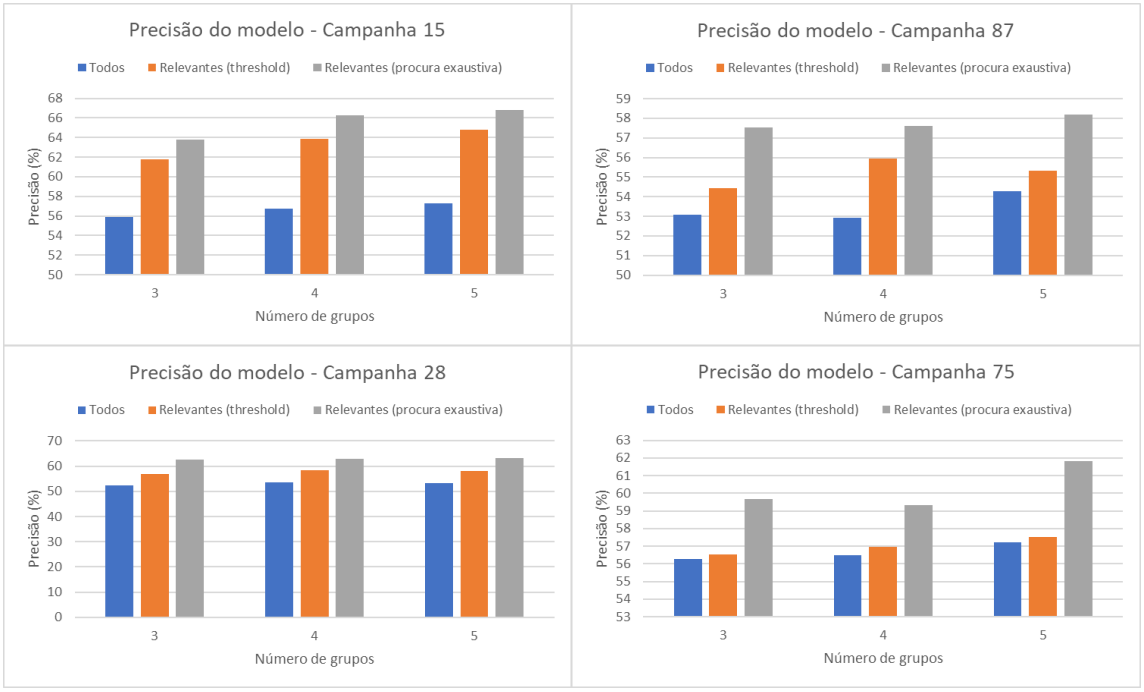


Figura 17 - Precisão dos modelos com os atributos todos e com os relevantes

Os resultados obtidos nas campanhas testadas mostram que os modelos constituídos por todos os atributos são os que apresentam menores valores de precisão, ou seja, os grupos são menos discriminatórios em relação à classe-alvo. Comparando os modelos com os atributos selecionados pelo threshold definido no método e por procura exaustiva, a segunda configuração apresentou-se superior em todos os testes realizados. Após esta experiência optou-se pelos modelos com os atributos escolhidos por procura exaustiva do subconjunto de atributos com maior importância.

7.1.5.3. Comparação da precisão de diferentes métodos de clustering

Na terceira experiência avaliou-se a precisão do modelo nas quatro campanhas selecionadas, por dois métodos de clustering com o subconjunto de atributos relevantes selecionado por procura exaustiva. Os métodos disponibilizados pelo Python e avaliados foram o K-Means e o hierárquico aglomerativo. O método hierárquico aglomerativo foi experimentado com duas funções de distância, a Euclidean e Manhattan, de maneira a verificar qual delas se adapta melhor aos dados. No método K-Means foi utilizada a função de distância Euclidean (única disponibilizada neste método) [48]. Os restantes parâmetros foram os pré-definidos nas respetivas bibliotecas destes dois métodos [47] [49]. A Figura 18 compara a precisão dos modelos para cada campanha com os dois métodos selecionados e as duas funções de distância. Mais detalhe das medições encontra-se no anexo B.1.3.

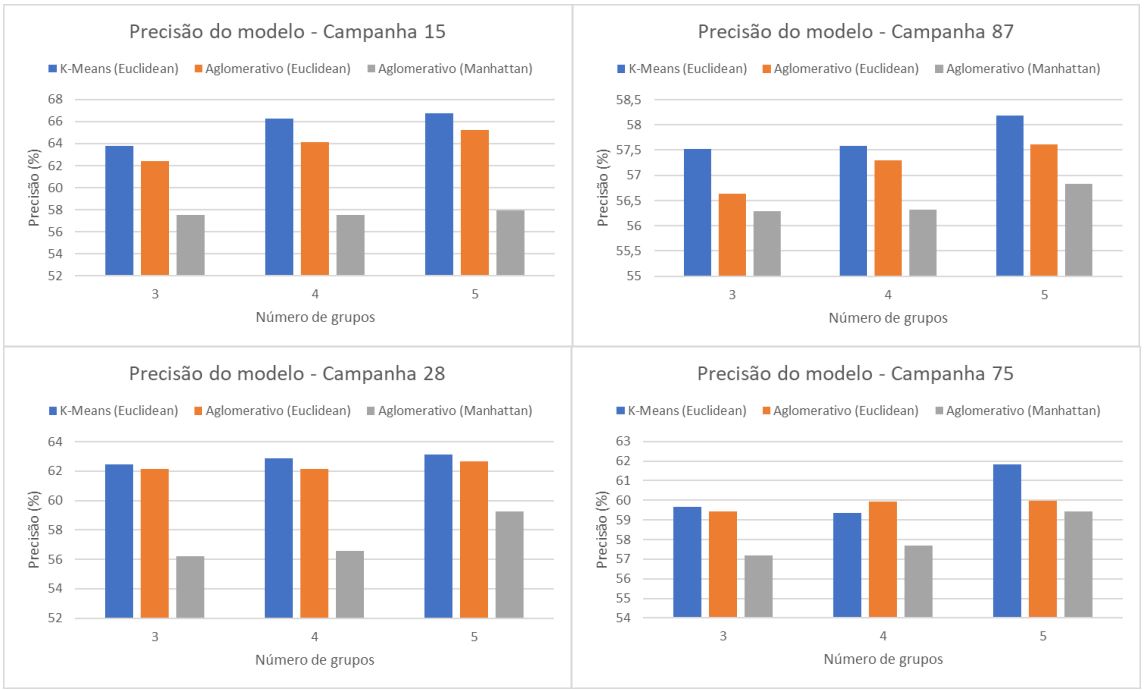


Figura 18 - Precisão dos modelos em diferentes métodos

Os resultados apresentados na Figura 18 mostram de um modo geral que o modelo do K-Means obtém uma maior precisão nas quatro campanhas. Do mesmo modo, verifica-se que no intervalo de grupos testados, a precisão do modelo aumenta com o número de grupos. Uma das possíveis razões para o método hierárquico apresentar piores resultados, pode estar relacionado com as fusões dos elementos realizadas nos passos intermédios não poderem ser desfeitas ou melhoradas. Por outro lado, o K-Means iterativamente minimiza a distância de todos os elementos aos centros dos grupos. Analisando os resultados do método hierárquico aglomerativo com as duas funções de distância, verifica-se que a função Euclidean foi superior à de Manhattan. Isto pode ser explicado pelo número de atributos distribuídos numa escala de valores ser muito maior que o número de atributos discretos num conjunto fixo de possíveis valores. Após a realização desta experiência optou-se pelos modelos K-Means.

7.1.5.4. Análise de Silhouette variando o número de grupos

Os gráficos da Figura 19 apresentam a análise de Silhouette para a campanha 15 para três, quatro e cinco grupos com o método K-Means e com os atributos selecionados por procura exaustiva. Os gráficos das restantes campanhas encontram-se no anexo B.1.4.

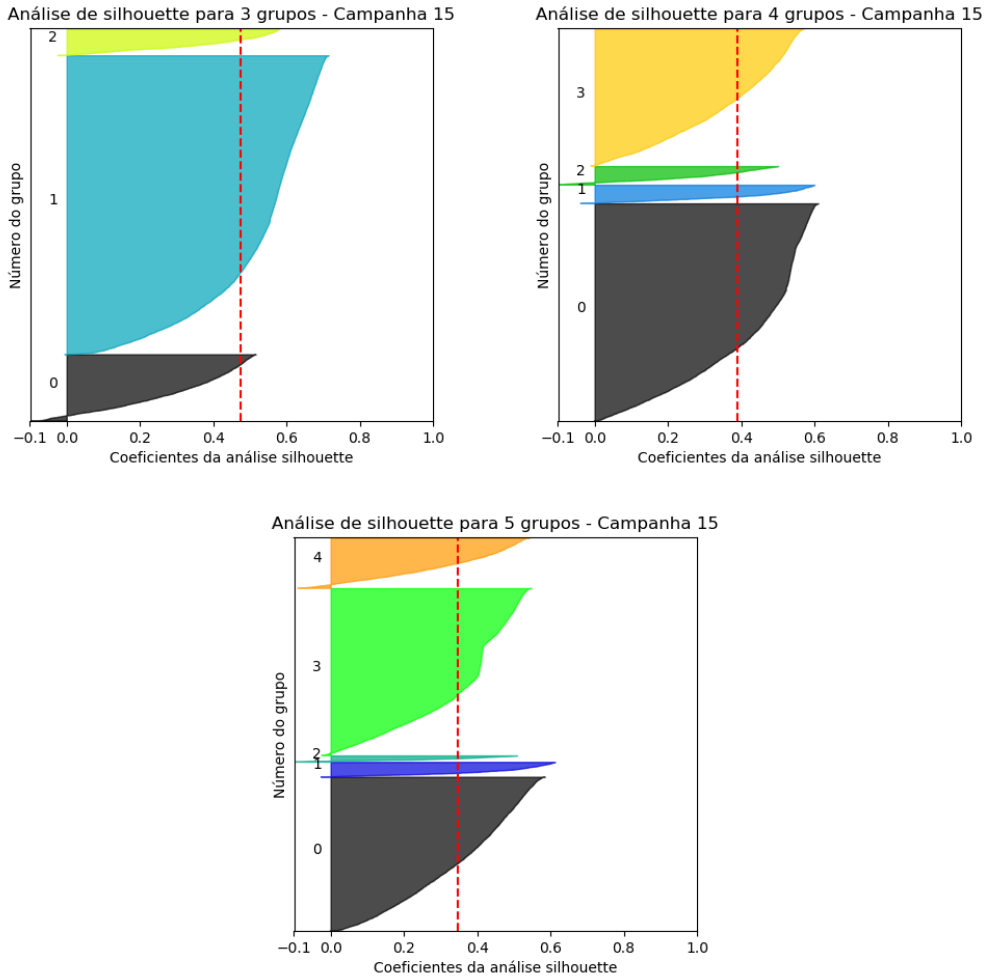


Figura 19 - Análise de Silhouette da campanha 15

No intervalo de grupos testados os resultados mostram que, quanto maior o número de grupos, menor o coeficiente de Silhouette médio de todos os elementos no conjunto de dados. Ou seja, maior o número de elementos, que podem ter sido atribuídos ao grupo errado e menor a distância média de separação entre os grupos.

7.1.6. Sumário

As experiências realizadas permitiram verificar que o modelo fica mais eficaz, quando construído com os atributos relevantes. Numa primeira fase é calculada a importância de cada atributo pelo método de filter model, chamado SelectFromModel. De seguida por procura exaustiva são realizados diversos testes a subconjuntos de atributos. O subconjunto selecionado é o que tiver maior valor de precisão no modelo. Dos dois métodos experimentados, verificou-se que os modelos do K-Means foram superiores ao hierárquico aglomerativo. Então, escolheu-se este modelo que teve melhores resultados e foi capaz de generalizar nas experiências realizadas nas quatro campanhas.

7.2. Previsão de consumo

Neste requisito pretende-se desenvolver modelos capazes de fazer uma previsão aproximada a três meses dos vários tipos de consumo. No seu desenvolvimento, avaliou-se o erro de previsão com dois métodos de séries temporais, o sazonal Auto-Regressive Integrated Moving Average (ARIMA) e o Prophet. Importante também avaliar a capacidade de generalização do modelo nas suas previsões variando o trimestre do ano.

7.2.1. Abordagem

Das duas abordagens a este requisito apresentadas no capítulo de Estado da Arte, uma baseada em modelos de redes neuronais [25] e a outra baseada no método ARIMA de séries temporais [26] [27], seguiu-se esta última. Esta escolha teve em conta a capacidade dos métodos de séries temporais serem dependentes do tempo e conseguirem lidar com a sazonalidade e tendência existente nos dados.

7.2.2. Métricas de avaliação de resultados

A precisão dos modelos criados foi avaliada usando duas métricas, o Root Mean Squared Error (RMSE) e a Mean Absolute Percentage Error (MAPE). O RMSE [50] permite medir a magnitude média do erro, atribuindo um maior peso aos erros maiores e um menor peso aos erros menores. Calcula-se usando a seguinte formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2}$$

Na fórmula, a variável y_i representa o valor observado, x_i o valor previsto, n o número de valores previstos.

A métrica alternativa MAPE [51] é mais intuitiva e fácil de interpretar por utilizadores não técnicos, e permite comparar resultados entre diferentes dados. O MAPE calcula o erro médio em percentagem usando a seguinte fórmula:

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{I=1}^{n}\frac{|\,y_i - x_i\,|}{y_i}$$

As variáveis nesta fórmula têm o mesmo significado da fórmula do RMSE.

7.2.3. Visualização da variação dos consumos

As Figuras 20 e 21 apresentam graficamente a variação do consumo mensal de recargas (usado nos serviços de telecomunicações), SMS e dados de internet ao longo dos anos de 2014 a 2017, para verificar a existência de tendência e de sazonalidade nos dados. Importa referir que os gráficos apresentam os valores de consumo normalizados para facilitar a sua visualização gráfica.

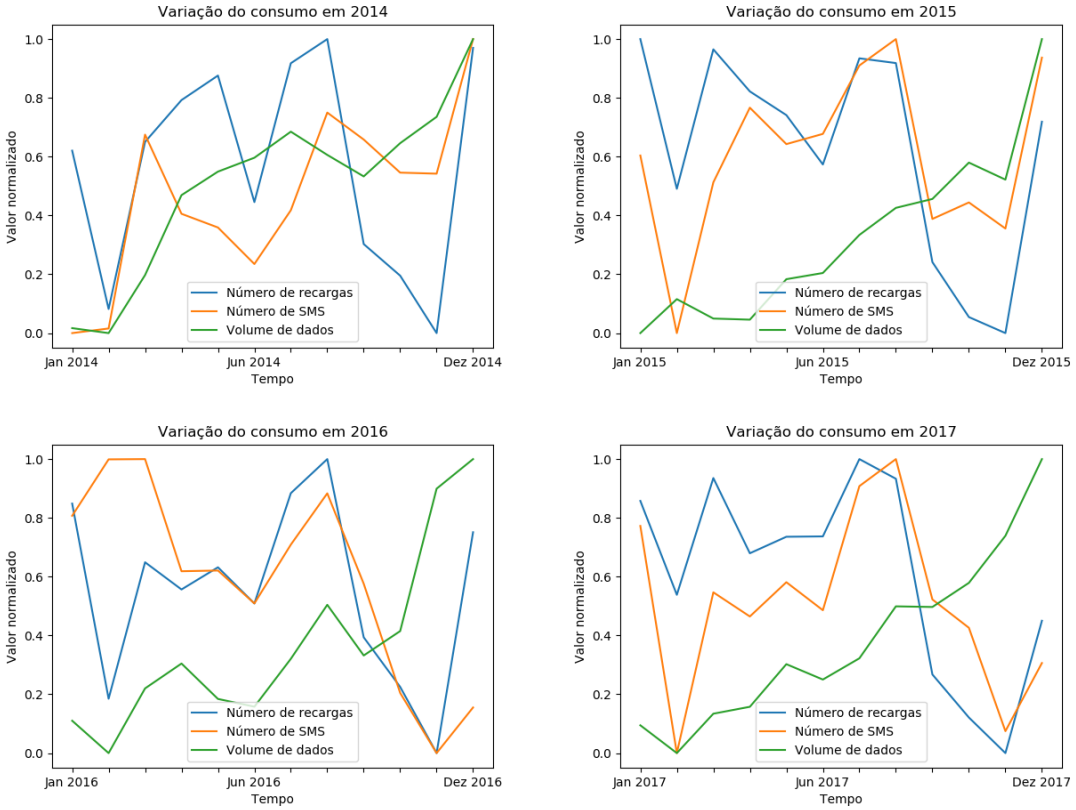


Figura 20 - Variação de consumo de cada um dos quatro anos

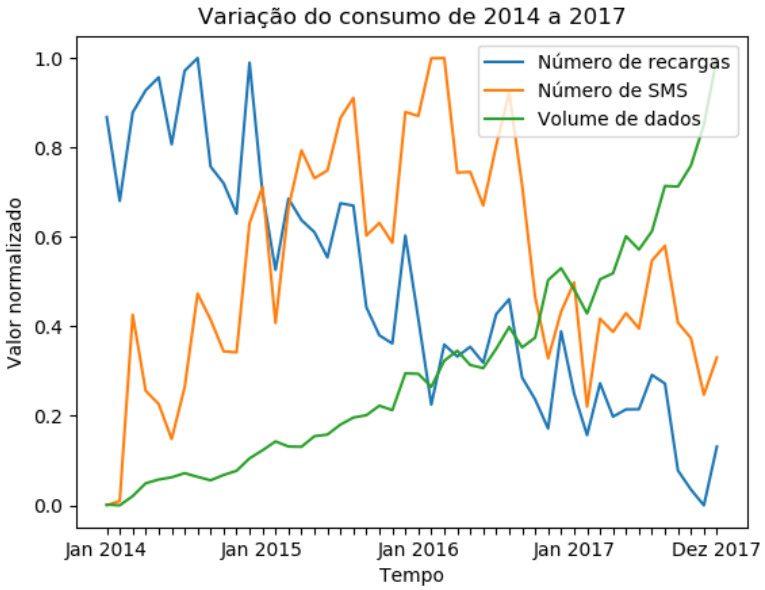


Figura 21 - Variação de consumo ao longo dos quatro anos

Pela análise dos gráficos anteriores observa-se ao longo do tempo, uma tendência decrescente no consumo de recargas, crescente no consumo de dados de internet e ambas no consumo de SMS. A sazonalidade verifica-se pelos temporários aumentos e decréscimos no consumo em certos meses do ano. Na série de recargas, existe um baixo consumo nos meses de fevereiro, outubro e novembro, e um consumo mais elevado nos meses de janeiro, agosto e dezembro. A sazonalidade verificada na série de SMS, é muito semelhante à de recargas. Finalmente a série de dados de internet, apresenta um consumo mais elevado e também crescimento no mês de dezembro de cada ano.

7.2.4. Setup experimental

Cada conjunto de dados foi dividido em conjunto de treino e de teste. Devido à existência de sazonalidade e tendência, utilizou-se um histórico para treino do modelo de três anos (trinta e seis meses), e os três meses seguintes para teste. Utilizou-se os quatro conjuntos de dados apresentados na Tabela 22 para teste dos modelos, sendo o lado esquerdo da tabela o período de treino e o lado direito o período de teste.

Treino	Teste / Previsão
Janeiro de 2014 - dezembro de 2016	Janeiro de 2017 - março de 2017
Abril de 2014 - março de 2017	Abril de 2017 - junho de 2017
Julho de 2014 - junho de 2017	Julho de 2017 - setembro de 2017
Outubro de 2014 - setembro de 2017	Outubro de 2017 - dezembro de 2017

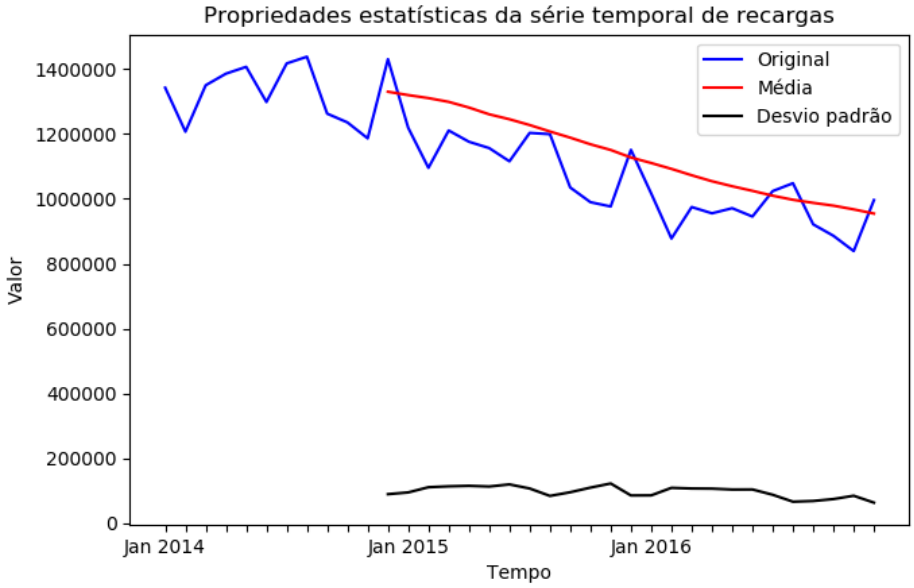
Tabela 22 - Conjuntos de dados utilizados nas experiências

O processo de seleção, preparação de dados está automatizado pelo módulo de previsão de consumo, descrito na secção 6.2.3 deste relatório.

7.2.5. Análise das séries temporais e parametrização dos modelos

O desenvolvimento do modelo ARIMA divide-se em duas etapas, uma de análise da série temporal e uma segunda de desenvolvimento de modelos variando a sua parametrização. Dado que o método de séries temporais ARIMA lida apenas com séries estacionárias no tempo, a primeira etapa consiste em verificar se a série é estacionária ou não com recurso a dois métodos. O primeiro de visualização gráfica das propriedades estatísticas da série temporal, a média móvel (cálculo em cada momento da média dos valores correspondentes ao último período sazonal) e o desvio padrão móvel. O segundo método com recurso ao teste estatístico de Dickey-Fuller. Caso a série seja não estacionária, é necessário torná-la estacionária com a realização de uma transformação de diferenciação. Em cada instante da série temporal subtrai a observação original com a do instante anterior [13]. Após a transformação realiza-se novo teste de estacionariedade, e este processo repete-se até a série se aproximar da estacionariedade. O parâmetro d do modelo corresponde ao número de diferenciações necessárias até estacionar a série. Segue-se a segunda etapa, de estimação dos parâmetros p e q com recurso aos gráficos Autocorrelation Function (ACF) e Partial Autocorrelation Function (PACF) respetivamente. O parâmetro p indica a ordem da componente AR do modelo e representa o período de tempo atrasado. O parâmetro q indica a ordem da componente MA do modelo e representa os erros de previsão atrasados. Em seguida são testadas e avaliadas várias combinações destes dois parâmetros do modelo. Nesta secção apresenta-se a série temporal de consumo do número de recargas usadas nos serviços de telecomunicações. Dado que as séries de SMS e de dados de internet seguem o mesmo processo, estas são apresentadas no anexo B.2.1 e B.2.2 respetivamente.

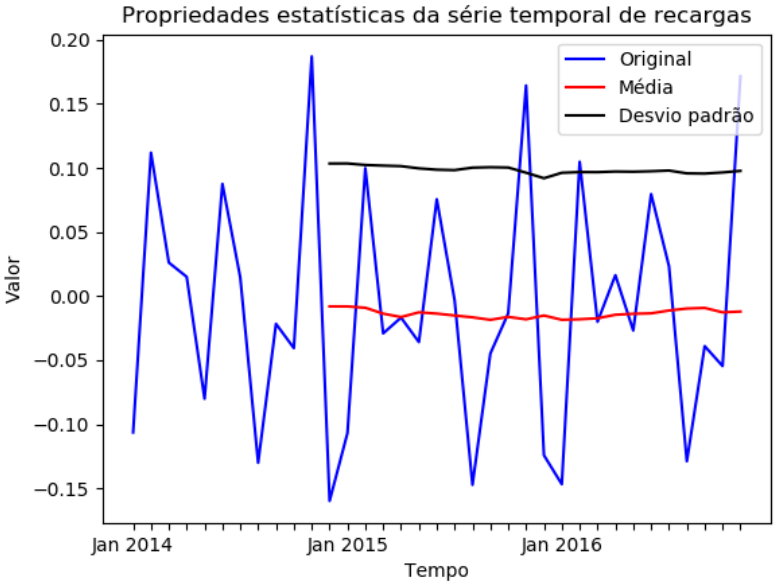
A Figura 22 apresenta o teste de estacionariedade da série temporal com dois métodos. A figura de cima apresenta a visualização gráfica das propriedades estatísticas da série ao longo do tempo, média móvel e desvio padrão móvel. Ambos foram configurados com o valor de doze [13], correspondente ao período ou ciclo sazonal. A figura de baixo mostra o teste estatístico de Dickey-Fuller.




```
Results of Dickey-Fuller Test:
Test Statistic                  -1.149969
p-value                          0.694740
#lags Used                       6.000000
Number of Observations Used     29.000000
Critical Value (1%)             -3.679060
Critical Value (5%)             -2.967882
Critical Value (10%)            -2.623158
```

Figura 22 - Teste de estacionariedade da série temporal de recargas original

Pela análise do gráfico anterior verifica-se um claro decréscimo da média móvel ao longo do tempo. Esta variação ao longo do tempo, indica que a série é não estacionária no tempo. O resultado do teste estatístico apresenta um valor maior que o valor crítico com um nível de confiança de 95% (-1.149969 > -2.967882), indicando também que a série é não estacionária no tempo. De seguida aplicou-se uma transformação de diferenciação de primeira ordem para tornar a série temporal estacionária no tempo. A Figura 23 apresenta o teste de estacionariedade da série após a transformação.




```
Results of Dickey-Fuller Test:
Test Statistic                  -4.815318
p-value                          0.000051
#lags Used                      10.000000
Number of Observations Used     24.000000
Critical Value (1%)             -3.737709
Critical Value (5%)             -2.992216
Critical Value (10%)            -2.635747
```

Figura 23 - Teste de estacionariedade da série temporal de recargas após a transformação

Pela análise do gráfico anterior verifica-se que as propriedades estatísticas da série temporal tornaram-se aproximadamente constantes ao longo do tempo. O resultado do teste estatístico é menor que o valor crítico para um nível de confiança de 95% (-4.815318 < -2.992216), indicando também que a série temporal se aproximou da estacionariedade. Então o parâmetro d do modelo é igual a um, ou seja, o número de diferenciações que permitiram aproximar a série da estacionariedade.

O próximo passo é determinar os valores dos parâmetros p (ordem da componente AR do modelo) e q (ordem da componente MA do modelo) do ARIMA. Para estimar estes dois parâmetros recorreu-se às funções de Autocorrelation Function (ACF) e Partial Autocorrelation Function (PACF). A Figura 24 apresenta os gráficos destas duas funções.

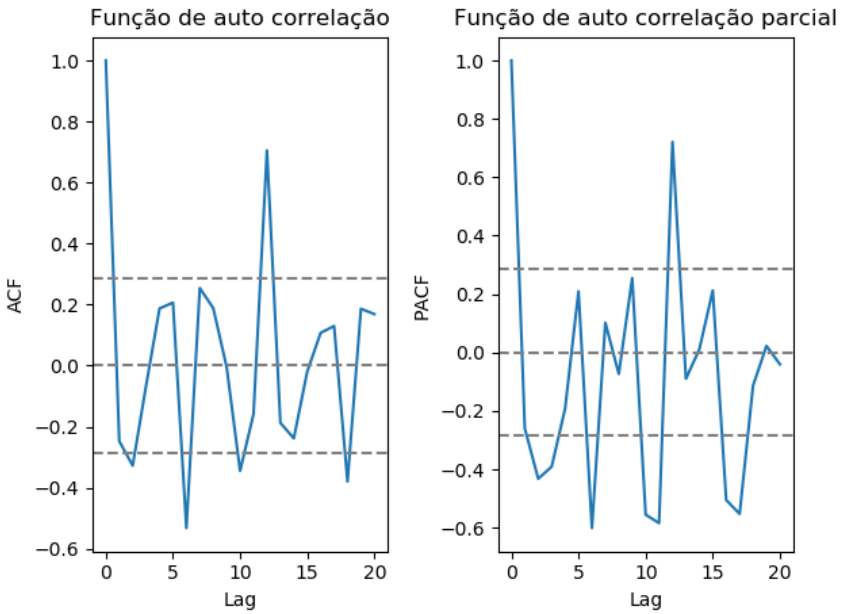


Figura 24 - Gráficos das funções ACF e PACF da série temporal de recargas

O gráfico do lado esquerdo da ACF permite estimar o valor do parâmetro p, que corresponde ao momento em que a função cruza o intervalo de confiança pela primeira vez, ou seja no gráfico acima isso acontece entre p = 0 e p = 1. O outro gráfico do lado direito da PACF permite estimar o valor do parâmetro q, pelo cruzamento da função com o intervalo de confiança, ou seja no gráfico acima isso acontece entre q = 0 e q = 1. De seguida fez-se os testes com as quatro combinações de parâmetros no modelo, (p = 0, d = 1, q = 0), (p = 1, d = 1, q = 0), (p = 0, d = 1, q = 1) e (p = 1, d = 1, q = 1). O parâmetro de sazonalidade foi configurado com o valor de doze, correspondente ao período sazonal. Os testes ao modelo com estas combinações de valores dos parâmetros encontram-se na Tabela 43 do anexo B.2.3.

O modelo Prophet foi mais simples de desenvolver que o ARIMA devido à sua capacidade de encontrar automaticamente pontos de inflexão nos dados, ou seja, onde existe mudanças de tendência [20]. Na parametrização deste modelo configurou-se o período de sazonalidade da série temporal com o valor de doze.

7.2.6. Experiências e avaliação de resultados

As Tabelas 23, 24 e 25 mostram a comparação do erro de previsão (RMSE e o MAPE) usando o ARIMA e Prophet, escolhendo o melhor resultado do modelo ARIMA. Quatro previsões foram usadas, correspondendo aos quatro trimestres do ano. As tabelas mostram o valor médio, desvio padrão e valor máximo dos erros dos quatro testes realizados.

Consumo de recargas				
Método	**RMSE**		**MAPE**	
	Média	**Máximo**	**Média**	**Máximo**
ARIMA	3.69e+4± 1.31e+4	4.97e+4	3.71 ± 1.25	5.01
Prophet	7.63e+4 ± 5.93e+4	1.05e+5	7.82 ± 2.54	11.08

Tabela 23 - Resultado dos modelos de previsão do número de recargas

Consumo de SMS				
Método	**RMSE**		**MAPE**	
	Média	**Máximo**	**Média**	**Máximo**
ARIMA	7.95e+5 ± 5.5e+5	1.66e+6	4.67 ± 3.09	9.43
Prophet	2.7e+6 ± 2.82e+6	4.54e+6	14.25 ± 8.76	27.4

Tabela 24 - Resultado dos modelos de previsão do número de SMS

Consumo de dados				
Método	**RMSE**		**MAPE**	
	Média	**Máximo**	**Média**	**Máximo**
ARIMA	2.59e+13 ±1.48e+13	5.12e+13	6.99 ± 5.79	17.0
Prophet	4.67e+13± 4.18e+13	6.35e+13	8.11 ± 3.39	13.26

Tabela 25 - Resultado dos modelos de previsão do volume de dados

Os resultados mostram que os modelos ARIMA foram sempre melhores que os Prophet (menor erro médio, quase metade) em todas as séries. Importa referir que, enquanto para as séries de recargas e de SMS o erro máximo foi sempre menor para o ARIMA, no caso da série de consumo de dados um dos trimestres teve um erro mais elevado para o ARIMA e o erro médio foi pouco melhor que o Prophet. No entanto, isto foi uma exceção, porque os resultados do ARIMA foram consistentemente superiores em todas as séries. Uma das possíveis razões poderá estar relacionada com a natureza dos dados, em que o método ARIMA se adaptou melhor aos dados de telecomunicações. A Figura 25 mostra graficamente os resultados de previsão do modelo ARIMA, comparando os valores reais e previstos, nos quatro testes de previsão. Os resultados de forma gráfica das séries de SMS e de dados de internet, são apresentadas no anexo B.2.1 e B.2.2 respetivamente.

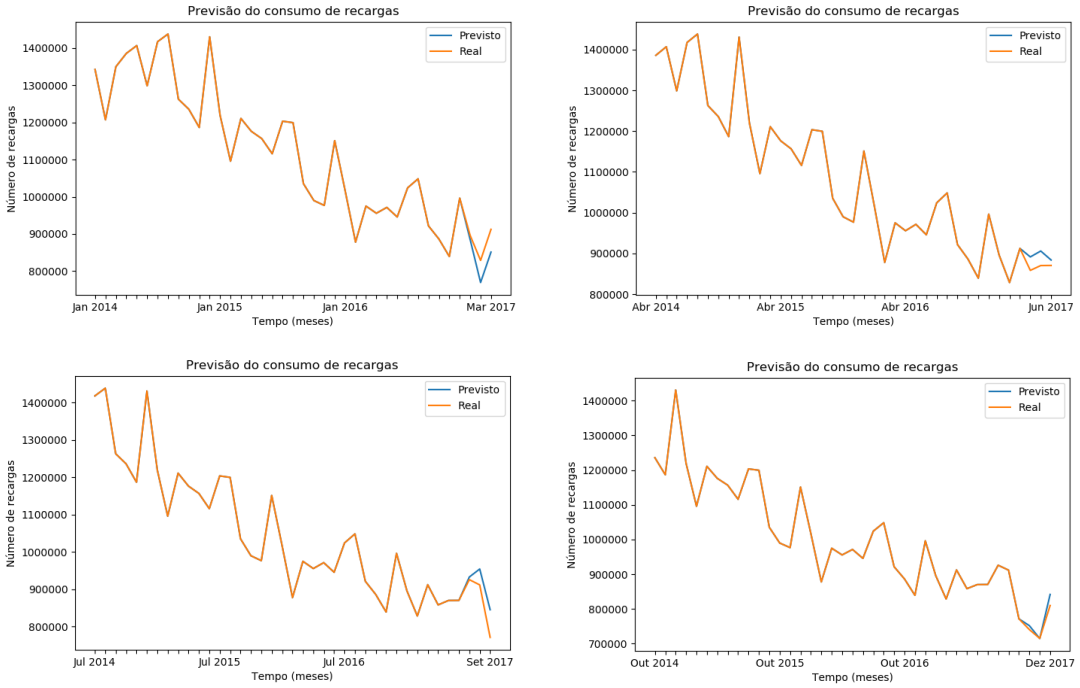


Figura 25 - Previsão de recargas a três meses

Os gráficos das previsões a três meses, mostram que conseguiram uma previsão aproximada boa em comparação com os valores reais. Outro teste realizado com o mesmo modelo ARIMA de previsão a três meses foi a previsão a doze meses a partir do histórico. A Figura 26 apresenta de forma gráfica o resultado deste teste.

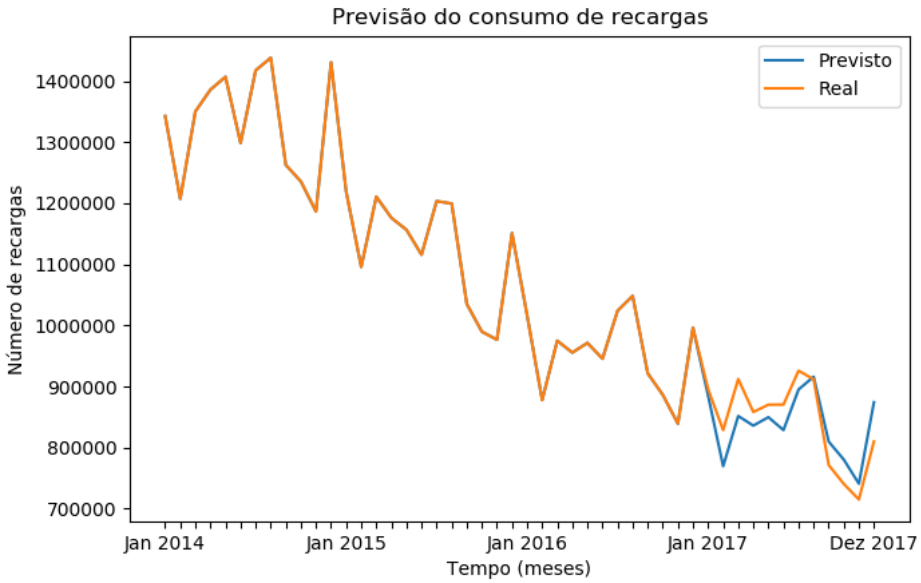


Figura 26 - Previsão de recargas a doze meses

O gráfico da previsão a doze meses, mostra que o modelo é capaz de lidar bastante bem com a tendência e a sazonalidade dos dados. Neste teste conseguiu-se um valor de RMSE de 3.99e+4 e um MAPE de 4.19%.

7.2.7. Sumário

Nas experiências realizadas verificou-se que os modelos ARIMA apresentaram melhores resultados sobre o Prophet nas três séries temporais de consumo, e capazes de lidar com a sazonalidade e tendência dos dados. Importa referir que as três séries de consumo, necessitaram de apenas uma diferenciação de primeira ordem para se aproximarem da estacionariedade no tempo. No entanto a parametrização dos modelos das três séries apresentaram variação nos parâmetros p e q.

7.3. Relações entre serviços subscritos

Esta secção apresenta um exemplo do resultado das regras de associação geradas pelo módulo de relações entre serviços subscritos descrito na secção 6.2.4 deste relatório. A Tabela 26 apresenta as regras geradas na fotografia de subscrições no primeiro mês de 2017 para um valor mínimo de suporte e confiança de 0.65. Importa referir que os nomes reais dos serviços foram substituídos por um nome genérico para manter o nome do operador confidencial e ocultar os detalhes do negócio.

Antecedente	Consequente	Suporte	Confiança
Serviço de carregamento em qualquer lugar	Serviço de dados móveis	0.68	0.7
Serviço de carregamento em qualquer lugar	Serviço de chamadas	0.83	0.85
Serviço de saldo para chamadas urgentes	Serviço de chamadas	0.83	0.84
Serviço de notificação de saldo, Serviço de carregamento em qualquer lugar	Serviço de dados móveis	0.68	0.7
Serviço de notificação de saldo, Serviço de carregamento em qualquer lugar	Serviço de chamadas	0.82	0.85
Serviço de carregamento em qualquer lugar, serviço de saldo para chamadas urgentes	Serviço de dados móveis	0.67	0.7
Serviço de carregamento em qualquer lugar, serviço de saldo para chamadas urgentes	Serviço de chamadas	0.83	0.86
Serviço de carregamento em qualquer lugar, serviço de saldo para chamadas urgentes	Serviço de chamadas	0.82	0.85
Serviço de carregamento em qualquer lugar	Serviço de dados móveis	0.68	0.96
Serviço de notificação de saldo	Serviço de dados móveis	0.67	0.96
Serviço de saldo para chamadas urgentes	Serviço de dados móveis	0.66	0.96

Tabela 26 - Regras de associação geradas

As regras geradas permitem descobrir o grau de confiança de relacionamento entre os diversos serviços do operador. Para um conjunto de cinco serviços do Pré-Pago, foram geradas onze relações entre os serviços com um valor máximo de confiança de 96%. O resultado mostra que alguns dos serviços estão frequentemente subscritos ao mesmo tempo pelos clientes. Esta informação vai ser útil na criação de pacotes com serviços agregados.

Capítulo 8

Testes

Este capítulo apresenta o plano e resultados de execução dos testes de aceitação, requisitos funcionais e não funcionais à plataforma desenvolvida. Os testes aos requisitos funcionais não incluem os testes à qualidade dos modelos, dado que estes foram apresentados e analisados no capítulo anterior.

8.1. Aceitação

Os testes de aceitação verificam se o projeto satisfaz as necessidades do cliente. Os requisitos funcionais foram apresentados aos Orientadores da empresa à medida que foram implementados e experimentados para validar se estes cumprem os objetivos definidos. A Tabela 27 mostra o resultado dos testes de aceitação.

Requisito funcional	Implementado	Resultado
Perfil de clientes	Sim	Aceite
Previsão de consumo - Recargas	Sim	Aceite
Previsão de consumo - SMS	Sim	Aceite
Previsão de consumo - Dados	Sim	Aceite
Anomalias de consumo por grupo	Não	Não aplicável
Relações entre serviços subscritos	Sim	Aceite
Autenticação	Sim	Aceite

Tabela 27 - Testes de aceitação

A tabela anterior mostra que os requisitos implementados foram de encontro aos objetivos pretendidos pela empresa. Todos os requisitos com grau de prioridade máximo, segundo as necessidades da empresa foram desenvolvidos. De notar que o requisito de Anomalias de consumo por grupo de utilizadores não foi implementado por falta de tempo e prioridade a outras tarefas, no entanto este não era dos mais prioritários.

8.2. Requisitos funcionais

O plano de testes aos requisitos funcionais é constituído por testes unitários e manuais. Os testes unitários de black-box abrangem os testes à API juntamente com os módulos da aplicação. Uma vez que este sistema não é crítico, os testes não abrangem os de white-box ao código fonte [41]. Os testes manuais são realizados ao mecanismo de parametrização automático do modelo de previsão de consumo, de modo a verificar se este encontra o modelo com menor erro de previsão.

8.2.1. Unitários

Os testes unitários têm o objetivo de garantir que a API está protegida contra parâmetros inválidos de entrada, acessos não autorizados a recursos e validar o resultado do pedido. Na realização de testes unitários definiram-se casos de teste para cada requisito funcional com os seus valores de entrada e o resultado esperado. Os casos de teste definidos tiveram em conta a fronteira entre os valores válidos e inválidos em função do domínio de valores aceite por cada variável de entrada. A comparação entre o resultado esperado e o obtido é realizado com

asserções. De maneira a automatizar a sua execução recorreu-se à Framework unittest do Python [52].

A Tabela 28 apresenta o planeamento e resultados dos casos de teste realizados ao primeiro requisito de perfil de clientes. De notar que na coluna chamada Saída, o código 200 significa que tudo funcionou corretamente, 400 recebeu um pedido inválido, 401 recebeu um pedido não autorizado. Os testes aos pedidos realizados com sucesso (código 200), validaram-se os dados devolvidos pelo método com recurso a asserções:

- Perfis: verifica se o número de perfis corresponde ao número de grupos recebido por parâmetro.
- Percentagem de clientes incentivados aderentes por grupo: verifica se esta lista está preenchida e contém o número de elementos igual ao número de grupos recebido por parâmetro.
- Lift por grupo: verifica se esta lista está preenchida e contém o número de elementos igual ao número de grupos recebido por parâmetro.
- Lift global: verifica se este valor está preenchido e é diferente de zero.

Foi ainda verificado por observação a coerência entre os valores de lift por grupo e o valor do lift global em reunião com os orientadores.

Requisito 1 - Perfil de clientes							
Teste	**Parâmetros de entrada**			**Observações**	**Saída (código)**		**Resultado**
	Campanha [15,87,28,75]	**nGrupos [3, 5]**	**Token**		**Esperado**	**Obtido**	
1	15	2	válido	nGrupos inválido	400	400	Passou
2	15	3	válido	nGrupos válido	200	200	Passou
3	15	4	válido	nGrupos válido	200	200	Passou
4	15	5	válido	nGrupos válido	200	200	Passou
5	15	6	válido	nGrupos inválido	400	400	Passou
6	“	“	válido	campanha e nGrupos vazios	400	400	Passou
7	“	4	válido	campanha vazia	400	400	Passou
8	15	“	válido	nGrupos vazio	400	400	Passou
9	87	4	válido	campanha válida	200	200	Passou
10	75	4	válido	campanha válida	200	200	Passou
11	100	4	válido	campanha inválida	400	400	Passou
12	15	4	inválido	token inválido	401	401	Passou

Tabela 28 - Testes unitários ao requisito de perfil de clientes

A Tabela 29 apresenta o planeamento e resultados dos casos de teste realizados ao segundo requisito de previsão de consumo. Os casos de teste realizados com sucesso (código 200), validaram-se os dados devolvidos pelo método com recurso a asserções:

- Datas de histórico: verifica se esta lista está preenchida e contém um número de datas igual a trinta e seis (período de histórico utilizado).
- Valores histórico: verifica se esta lista está preenchida e contém um número de valores igual a trinta e seis do tipo de consumo recebido por parâmetro.
- Datas previstas: verifica se esta lista está preenchida e contém um número de datas que foram previstas igual a três (período de previsão).
- Valores previstos: verifica se esta lista está preenchida e contém um número de valore igual a três do tipo de consumo previsto.
- Erro de previsão: verifica se este campo está preenchido.

Requisito 2 - Previsão de consumo						
Teste	Parâmetros de entrada		Observações	Saída (código)		Resultado
	tipoConsumo [recargas, SMS, dados]	Token		Esperado	Obtido	
1	recargas	válido	tipoConsumo válido	200	200	Passou
2	SMS	válido	tipoConsumo válido	200	200	Passou
3	dados	válido	tipoConsumo válido	200	200	Passou
4	chamadas	válido	tipoConsumo inválido	400	400	Passou
5	“	válido	tipoConsumo vazio	400	400	Passou
6	SMS	inválido	token inválido	401	401	Passou

Tabela 29 - Testes unitários ao requisito de previsão de consumo

A Tabela 30 apresenta o planeamento e resultados dos casos de teste realizados ao terceiro requisito de determinar relações entre serviços subscritos. Em cada caso de teste, validaram-se os quatro atributos de cada uma das regras de associação devolvidas pelo método, foram preenchidos corretamente com recurso a asserções.

Requisito 3 - Relações entre serviços subscritos									
Teste	Parâmetros de entrada					Observações	Saída (código)		Resultado
	ano [2017]	mês [1,12]	sup. [0.6,1.0]	conf. [0.6,1.0]	Token		Esperado	Obtido	
1	2017	0	0.75	0.75	válido	mês inválido	400	400	Passou
2	2017	1	0.75	0.75	válido	tudo válido	200	200	Passou
3	2017	2	0.75	0.75	válido	tudo válido	200	200	Passou
4	2017	3	0.75	0.75	válido	tudo válido	200	200	Passou
5	2017	12	0.75	0.75	válido	tudo válido	200	200	Passou
6	2017	13	0.75	0.75	válido	mês inválido	400	400	Passou
7	2017	1	0.59	0.75	válido	suporte inválido	400	400	Passou
8	2017	1	0.60	0.75	válido	tudo válido	200	200	Passou
9	2017	1	1.0	0.75	válido	tudo válido	200	200	Passou
10	2017	1	1.01	0.75	válido	suporte inválido	400	400	Passou
11	2017	1	0.75	0.59	válido	confiança inválido	400	400	Passou
12	2017	1	0.75	0.6	válido	tudo válido	200	200	Passou
13	2017	1	0.75	1.0	válido	tudo válido	200	200	Passou
14	2017	1	0.75	1.01	válido	confiança inválido	400	400	Passou
15	“	“	“	“	válido	parâmetros inválidos	400	400	Passou
16	2017	13	0.5	0.75	inválido	token inválido	401	401	Passou

Tabela 30 - Testes unitários ao requisito de relações entre serviços subscritos

Importa referir que os testes realizados não passaram todos na primeira tentativa, mas após várias iterações com correções efetuadas conseguiu-se passar todos os casos de teste. Quanto à validação dos dados lidos do sistema fonte do Data Warehouse, utilizados no segundo e terceiro requisito, decidiu-se não os fazer, uma vez que sobre estes é aplicado o processo de ETL (Extract, Transform, Load). O ETL é da responsabilidade de outros projetos da empresa consistindo na realização das tarefas de limpeza e tratamento de valores em falta ou sem sentido.

8.2.2. Validação ao mecanismo de parametrização automático do ARIMA

Esta seção apresenta os resultados de validação do mecanismo de parametrização automática do modelo ARIMA descrito na seção 6.2.3. A Tabela 31 mostra o erro de cada combinação testada pela abordagem de pesquisa exaustiva na previsão de três meses, para as três séries de consumo testadas.

A abordagem de configuração de parâmetros por pesquisa exaustiva avalia os erros de previsão para todos os casos e, em seguida escolhe aquele com o menor erro, mostrado em negrito. Em todos os testes verifica-se que o mecanismo implementado foi capaz de encontrar automaticamente a combinação de parâmetros que minimiza o erro de previsão nos dados históricos.

Os parâmetros escolhidos pela abordagem automática para as três séries (recargas, SMS e dados de internet) foram sempre consistentes com os encontrados nas experiências de configuração manual (discutido e exemplificado na secção 7.2). De notar que a abordagem manual envolve inspeção humana iterativa, executando o teste Dickey-Fuller ou inspeção visual, depois diferenciando e testando sucessivamente com os dois testes anteriores até que os limites de decisão sejam atingidos. A pesquisa exaustiva simplesmente substitui esse processo manual pela versão automatizada, com bons resultados para estes conjuntos de dados e objetivos de previsão.

ARIMA	**Recargas**		**SMS**		**Dados**	
(p, d, q)	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**
(0, 0, 0)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(0, 0, 1)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(0, 0, 2)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(0, 1, 0)	3.87e+4	3.99	**7.95e+5**	4.67	3.98+e13	9.05
(0, 1, 1)	**3.69e+4**	3.71	8.82e+5	5.19	4.02+e13	9.08
(0, 1, 2)	3.75e+4	3.83	1.02e+6	6.63	3.84+e13	8.52
(1, 0, 0)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(1, 0, 1)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(1, 0, 2)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(1, 1, 0)	3.86e+4	3.96	8.34e+5	4.89	4.18e+13	9.65
(1, 1, 1)	3.77e+4	3.84	1.71e+6	9.24	**2.59e+13**	6.99
(1, 1, 2)	3.75e+4	3.83	1.04e+6	5.88	2.84e+13	7.17
(2, 0, 0)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(2, 0, 1)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(2, 0, 2)	Discarded	Discarded	Discarded	Discarded	Discarded	Discarded
(2, 1, 0)	3.76e+4	3.85	9.66e+5	5.30	3.94e+13	8.94
(2, 1, 1)	3.88e+4	4.05	1.08e+6	6.02	3.55e+13	8.94
(2, 1, 2)	6.58e+4	7.15	1.35e+6	8.01	3.96e+13	10.02

Tabela 31 – Valores dos parâmetros encontrados por procura exaustiva

Embora o mecanismo funcione corretamente para os conjuntos de dados testados, a sua execução leva cerca de seis segundos como mostram os resultados apresentados na secção 8.3.1. Então é importante desenvolver uma abordagem mais eficiente para parametrização do modelo. Portando, identificou-se como trabalho futuro a possibilidade de melhorar o mecanismo de parametrização do modelo com a aplicação de heurísticas para reduzir o espaço de procura.

8.3. Requisitos não funcionais

Esta secção apresenta os testes de avaliação e validação aos requisitos não funcionais de desempenho, robustez e interoperabilidade descritos no capítulo 5 de especificação de requisitos.

8.3.1. Desempenho

O desempenho da plataforma desenvolvida foi avaliado pelo tempo de execução com recurso à framework unittest do Python [52]. Neste teste, mediu-se o tempo de resposta no processamento de cada pedido recebido, desde o momento que o pedido é recebido até receber a resposta. Relembrando o cenário de desempenho descrito na secção 5.2 do relatório, espera-se um tempo máximo de sete segundos. A Tabela 32 apresenta o tempo médio e máximo que leva a responder a cada requisito em dez execuções. De notar que no teste ao requisito de previsão, comparou-se o mecanismo de parametrização automático com um único teste (modelo estudado na secção 7.2 usado para referência de comparação). O teste foi realizado em uma máquina de desenvolvimento com as seguintes características:

- Sistema operativo: Windows 8 de 64 bits
- Processador: i5 de 2.50 GHz
- Memória RAM: 8.00 Gb
- Disco HDD: 297 Gb

Requisito funcional	Tempo (s)		Resultado
	Médio	Máximo	
Perfil de clientes - campanha 15	3.66 ± 0.19	3.86	Passou
Perfil de clientes - campanha 87	3.46 ± 0.18	3.79	Passou
Perfil de clientes - campanha 75	1.31 ± 0.36	1.98	Passou
Previsão de consumo recargas (automático)	6.38 ± 0.11	6.57	Passou
Previsão de consumo SMS (automático)	6.32 ± 0.31	6.94	Passou
Previsão de consumo dados (automático)	6.21 ± 0.39	6.92	Passou
Previsão de consumo recargas (único)	0.30 ± 0.16	0.55	Passou
Previsão de consumo SMS (único)	0.14 ± 0.04	0.19	Passou
Previsão de consumo dados (único)	0.61 ± 0.03	0.66	Passou
Relações entre serviços subscritos	3.86 ± 0.24	3.94	Passou

Tabela 32 - Testes de desempenho

O resultado dos testes de desempenho mostra que os requisitos atingiram tempos de execução aceitáveis dentro do objetivo definido. Analisando o tempo de execução dos requisitos de previsão usando o mecanismo de procura exaustiva, estes levam cerca de seis segundos ficando próximo do limite, devido ao número significativo de alternativas necessárias a serem testadas e avaliadas. Apresentou-se o tempo de execução deste requisito em reunião, e foi aceite pelo Orientador do projeto. No entanto para trabalho futuro, propõe-se uma possibilidade de melhoria.

8.3.2. Robustez

Os testes de robustez foram realizados ao nível funcional e não funcional, tendo em conta os dois cenários descritos na secção 5.2 de requisitos não funcionais. Ao nível funcional foi avaliado se a plataforma está protegida contra parâmetros inválidos de entrada. Estes testes foram realizados juntamente com os testes unitários descritos na secção 8.2.1. Ao nível não funcional foi avaliado o tratamento de exceções da plataforma na presença de dois tipos de

falhas, de rede e de ligação à base de dados. Para simular a primeira falha, desligou-se o cabo de ligação à rede. Para simular a segunda falha, alteraram-se os dados de acesso à base dados no ficheiro de configuração. Importa referir que na presença destas falhas, a plataforma tenta ligar-se de novo à fonte de dados e na ocorrência de uma segunda falha desiste. Em ambos os cenários foi avaliada a capacidade da plataforma continuar em funcionamento e devolver uma resposta adequada. A Tabela 33 apresenta os resultados dos testes destes dois cenários de falha.

Requisito funcional	Falha simulada	Saída (código)	Resultado
Previsão de consumo	Rede	500	Passou
	Ligação à base de dados	500	Passou
Relações entre serviços subscritos	Rede	500	Passou
	Ligação à base de dados	500	Passou

Tabela 33 - Testes de robustez

Os resultados mostram que a plataforma está protegida contra os parâmetros de entrada inválidos, falhas de rede e de ligação à base de dados, sendo capaz de as detetar, continuar em funcionamento e devolver uma resposta adequada.

8.3.3. Interoperabilidade

A interoperabilidade da plataforma desenvolvida é suportada pela tecnologia REST, projetada para este requisito específico [41]. Neste projeto, foi criada e documentada (na secção 6.2.6) uma interface de integração REST com um conjunto de endpoints disponibilizados por HTTP. A interface utiliza o formato de dados JavaScript Object Notation (JSON), interoperável e universal da indústria para devolver informação.

Capítulo 9

Conclusão

O trabalho desenvolvido ao longo do estágio resultou numa prova de conceito que será a base das funcionalidades futuras dos produtos da empresa na área de Inteligência no Negócio.

9.1. Trabalho realizado

O relatório apresentou todas as fases do desenvolvimento do trabalho ao longo do estágio. O primeiro semestre permitiu estudar a aplicação de técnicas de data mining e métodos de séries temporais aos dados reais de telecomunicações, com o objetivo de integrar numa aplicação de suporte à decisão. Com base no plano de desenvolvimento dos produtos da empresa e na informação recolhida dos sistemas de suporte à decisão do mercado concorrente, fez-se a especificação de requisitos e o desenho da plataforma. Relativamente ao segundo semestre, desenvolveram-se os módulos da plataforma em conjunto com o estudo e experimentação de modelos dos requisitos funcionais com dados reais de telecomunicações. Após a fase de implementação realizou-se o planeamento e execução dos testes de validação à plataforma.

Deste trabalho resultou uma plataforma desenhada para a interoperabilidade com as funcionalidades especificadas e com mecanismos automáticos. Os mecanismos desenvolvidos, permitem ajustar automaticamente os modelos e o uso das funcionalidades numa aplicação de suporte à decisão sem a necessidade de ajuste manual. Do ponto de vista científico, concluiu-se a vantagem de usar alguns métodos em relação a outros no contexto das telecomunicações. Os resultados das experiências realizadas deram vantagem ao método de clustering K-Means na segmentação de clientes e ao método de séries temporais ARIMA na previsão de consumo. Na perspetiva da empresa a plataforma desenvolvida permitirá integrar com a ferramenta de Business Intelligence (BIT) do NGIN PCC para, a partir dos dados que a própria ferramenta disponibiliza, passar a fornecer previsão de consumo. No Active Campaign Manager (ACM), a partir dos dados de clientes e campanhas permitirá apresentar sugestões de melhoria do público-alvo a uma campanha com o objetivo de melhorar a assertividade e poupar recursos no incentivo dos clientes.

9.2. Trabalho futuro

Como trabalho futuro é necessário integrar os requisitos implementados com as ferramentas de visualização de resultados usadas nos produtos da empresa. Para os requisitos implementados, existe sempre a possibilidade de os seus resultados serem melhorados com recurso a novas abordagens e métodos. Em particular no requisito de previsão de consumo, pretende-se desenvolver um mecanismo de parametrização automática do modelo mais eficiente que o desenvolvido neste trabalho, aplicando heurísticas que reduzam o espaço de procura. Finalmente seria importante desenvolver o requisito de deteção de anomalias de consumo de um grupo de utilizadores, que atualmente é realizado de forma manual tendo como consequência a ocorrência de erros e um elevado custo para o operador.

9.3. Balanço

A realização do estágio integrado num contexto empresarial foi muito gratificante tanto ao nível pessoal como académico. As competências pessoais foram potenciadas ao nível do trabalho em grupo, autonomia e gestão do tempo. No aspeto académico permitiu usar muitos conhecimentos e competências adquiridas num projeto e para resolver problemas concretos dos operadores de telecomunicações. Este objetivo foi conseguido com a aplicação de técnicas de data mining na análise de dados reais de telecomunicações. A sua aplicação demostrou enorme potencial na área de telecomunicações por duas razões. A primeira pela enorme quantidade e qualidade de dados disponíveis, a segunda pela inovação e valor acrescentado que fornece às ferramentas de suporte à decisão usadas pelos operadores.

A complexidade e abrangência desta área de análise de dados num contexto real, a limitação temporal, a curva de aprendizagem ingreme e alguma inexperiência do estagiário não permitiram a implementação de todas as funcionalidades. No entanto conseguiram-se as mais importantes e prioritárias que permitem dar resposta a alguns problemas dos clientes da Altice Labs. O principal desafio encontrado no desenvolvimento das funcionalidades foi torná-las generalizáveis, de modo a adaptarem-se a novos dados e poderem ser usadas numa aplicação de suporte à decisão sem a necessidade de ajuste manual.

Referências

[1] Bahjat El-Darwiche, 2017 Telecommunications Trends, disponível em https://www.strategyand.pwc.com/trend/2017-telecommunications-industry-trends, consultado em 2017-12-06.

[2] Slides das aulas de Gestão de Projetos de Marco Vieira, consultado em 2017-09-18.

[3] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide, consultado em 2017-09-28.

[4] Cross-industry standard process for data mining, disponível em https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining, consultado em 2017-09-27.

[5] GanttProject: free desktop project management app, disponível em http://www.ganttproject.biz/, consultado em 2017-09-18.

[6] Risk Management Plan Template, disponível em https://www2a.cdc.gov/cdcup/library/templates/CDC_UP_Risk_Management_Plan_Template.doc, consultado em 2017-09-20.

[7] Ćamilović, D. (2008). Data mining and CRM in telecommunications. Serbian Journal of Management, 3(1), 61-72, consultado em 2017-10-26.

[8] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier, consultado em 2017-10-27.

[9] Steinbach, M., Karypis, G., & Kumar, V. (2000, August). A comparison of document clustering techniques. In KDD workshop on text mining (Vol. 400, No. 1, pp. 525-526), consultado em 2017-09-29.

[10] Kumbhare, T. A., & Chobe, S. V. (2014). An Overview of Association Rule Mining Algorithms. International Journal of Computer Science and Information Technologies, 5(1), 927-930, consultado em 2017-10-09.

[11] Kavitha, M., & Selvi, S. T. Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Pros and Cons. International Journal of Computer Science Trends and Technology (I JCS T)–Volume, 4, consultado em 2017-10-10.

[12] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. Data Classification: Algorithms and Applications, 37, consultado em 2018-03-12.

[13] A comprehensive beginner's guide to create a Time Series Forecast, disponível em https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/, consultado em 2018-02-15.

[14] Prophet Forecasting at scale, disponível em https://facebook.github.io/prophet/, consultado em 2018-02-16.

[15] ARIMA - Time Series Analysis of Tractor Sales, disponível em http://ucanalytics.com/blogs/wp-content/uploads/2017/08/ARIMA-TimeSeries-Analysis-of-Tractor-Sales.html, consultado em 2018-02-20.

[16] MATH6011: Forecasting, disponível em https://www.southampton.ac.uk/~abz1e14/papers/Forecasting.pdf, consultado em 2018-03-22.

[17] Taylor, S. J., & Letham, B. (2017). Forecasting at scale. The American Statistician, (just-accepted), consultado em 2018-04-16.

[18] Analytics Vidhya, A Comprehensive Guide to Data Exploration, disponível em https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/, consultado em 2017-10-11.

[19] Feature selection — scikit-learn 0.19.1 documentation, disponível em http://scikit-learn.org/stable/modules/feature_selection.html, consultado em 2018-02-20.

[20] Telecom Analytics Solutions V1.0.4 brings advanced, predictive analytic insights to communication service providers, disponível em https://www-01.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/2/897/ENUS216-032/index.html&lang=en&request_locale=en, consultado em 2017-10-19.

[21] IBM Analytic Solutions | Telecom, disponível em https://www.ibm.com/analytics/us/en/industry/telecom-customer-experience-management/index.html, consultado em 2017-10-18.

[22] Telecoms-specific Business Intelligence, designed to provide operational and strategic insights through Big Data analytics, disponível em https://www.asiainfo.com/Portals/0/New_Branded_Collateral/Datasheets/BI%20datasheet.pdf, consultado em 2017-10-20.

[23] TBIDS on Oracle: Comprehensive BI Solution for the Telecommunications Industry, disponível em http://www.oracle.com/ocom/groups/public/@opnpublic/documents/webcontent/021728.pdf, consultado em 2017-10-21.

[24] Jansen, S. M. H. (2007). Customer segmentation and customer profiling for a mobile telecommunications company based on usage behavior. A Vodafone Case Study, consultado em 2017-10-29.

[25] BRANCO, S. T., & DE SAMPAIO, R. J. B. A New Artificial Neural Networks Forecast Model in Telecommunications, consultado em 2017-10-28.

[26] Wang, M., Wang, Y., Wang, X., & Wei, Z. (2015). Forecast and Analyze the Telecom Income based on ARIMA Model. The Open Cybernetics & Systemics Journal, 9(1), consultado em 2018-03-15.

[27] Is Prophet Really Better than ARIMA for Forecasting Time Series Data, disponível em https://blog.exploratory.io/is-prophet-better-than-arima-for-forecasting-time-series-fa9ae08a5851, consultado em 2018-05-16.

[28] Insani, R., & Soemitro, H. L. (2016, May). Data mining for marketing in telecommunication industry. In Region 10 Symposium (TENSYMP), 2016 IEEE (pp. 179-183). IEEE, consultado em 2017-10-30.

[29] Iglesias, J. A., Ledezma, A., Sanchis, A., & Angelov, P. (2017). Real-Time Recognition of Calling Pattern and Behaviour of Mobile Phone Users through Anomaly Detection and Dynamically-Evolving Clustering. Applied Sciences, 7(8), 798., consultado em 2017-10-26.

[30] Yu, Q., Jibin, L., & Jiang, L. (2016). An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks. International Journal of Distributed Sensor Networks, 12(1), 9653230, consultado em 2018-04-16.

[31] Weka, disponível em https://www.cs.waikato.ac.nz/ml/weka/, consultado em 2017-11-03.

[32] H2O.ai, disponível em https://www.h2o.ai/, consultado em 2017-11-05.

[33] Scikit-learn: machine learning in Python — scikit-learn 0.19.1 documentation, disponível em http://scikit-learn.org/stable/, consultado em 2017-11-04.

[34] Mlxtend 0.10.0: Python Package Index, disponível em https://pypi.python.org/pypi/mlxtend, consultado em 2017-11-04.

[35] StatsModels Statistics in Python, disponível em http://www.statsmodels.org/devel/generated/statsmodels.tsa.arima_model.ARIMA.html, consultado em 2018-03-20.

[36] Gartner Magic Quadrant for Data Science Platforms, disponível em https://rapidminer.com/resource/gartner-magic-quadrant-data-science-platforms/, consultado em 2017-11-03.

[37] Data Science Platform | RapidMiner, disponível em https://rapidminer.com/, consultado em 2017-11-04.

[38] Dask natively scales Python, disponível em https://dask.pydata.org/en/latest/, consultado em 2018-03-22.

[39] MoSCoW Method, disponível em https://www.projectsmart.co.uk/moscow-method.php, consultado em 2017-10-11.

[40] Slides das aulas de Arquitetura de Software de Bruno Cabral, consultado em 2017-11-14.

[41] Slides das aulas de Qualidade e Confiabilidade de Software de Raul Barbosa e Henrique Madeira, consultado em 2018-02-04.

[42] The C4 model for software architecture, disponível em https://c4model.com/, consultado em 2017-11-27.

[43] Flask (A Python Microframework), disponível em http://flask.pocoo.org/, consultado em 2017-12-14.

[44] Slides das aulas de Integração de Sistemas de Filipe Araújo, consultado em 2017-12-14.

[45] PostgreSQL, disponível em https://www.postgresql.org/, consultado em 2018-01-18.

[46] Flask-JWT-Entended's Documentation, disponível em http://flask-jwt-extended.readthedocs.io/en/latest/, consultado em 2018-04-16.

[47] Sklearn.cluster.KMeans, disponível em http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html, consultado em 2018-02-29.

[48] GitHub k_means_.py, disponível em https://github.com/scikit-learn/scikit-learn/blob/a24c8b464d094d2c468a16ea9f8bf8d42d949f84/sklearn/cluster/k_means_.py, consultado em 2018-02-28.

[49] Sklearn.cluster.AgglomerativeClustering, disponível em http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html, consultado em 2018-02-29.

[50] MAE and RMSE – Which Metric is Better, disponível em https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d, consultado em 2018-03-20.

[51] Mean absolute percentage error, disponível em https://en.wikipedia.org/wiki/Mean_absolute_percentage_error, consultado em 2018-03-20.

[52] Unit testing Framework, disponível em: https://docs.python.org/3/library/unittest.html, consultado em 2018-04-09.

Anexo A

Gestão do Projeto

As Figuras 27 e 28 deste anexo apresentam os diagramas de Gantt do planeamento do primeiro e segundo semestre respetivamente, descrito no capítulo 2 do relatório. A Figura 29 apresenta o diagrama de Gantt final do segundo semestre.

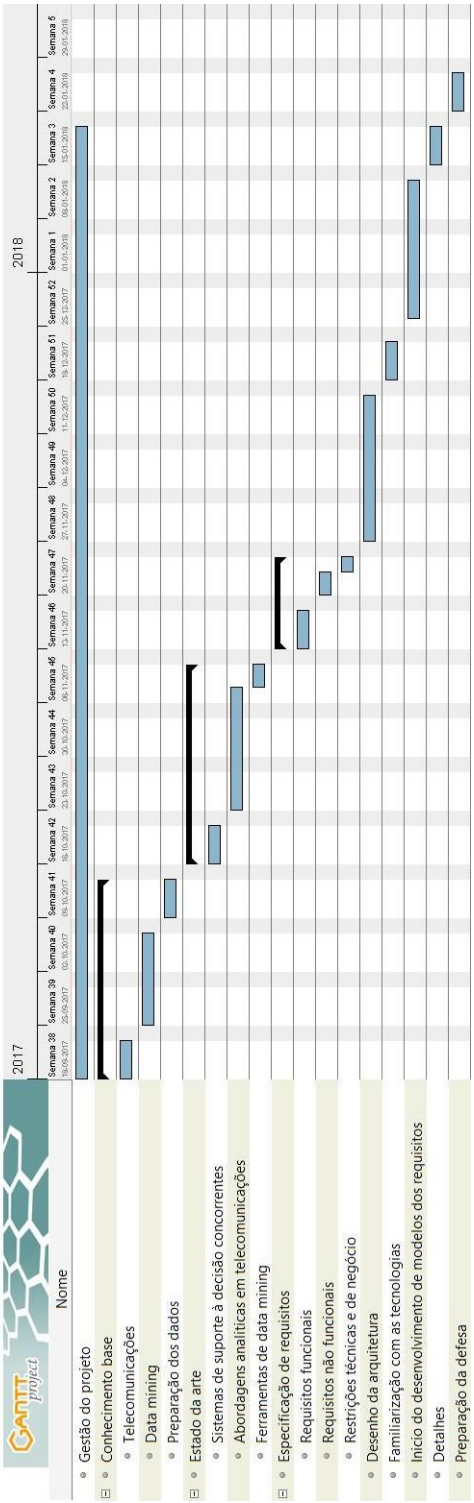


Figura 27 - Diagrama de Gantt do primeiro semestre

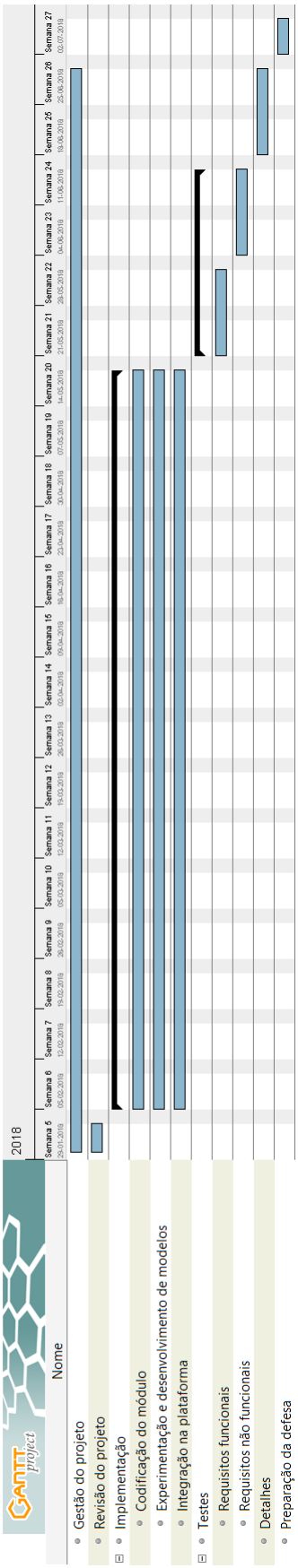


Figura 28 - Diagrama de Gantt do segundo semestre

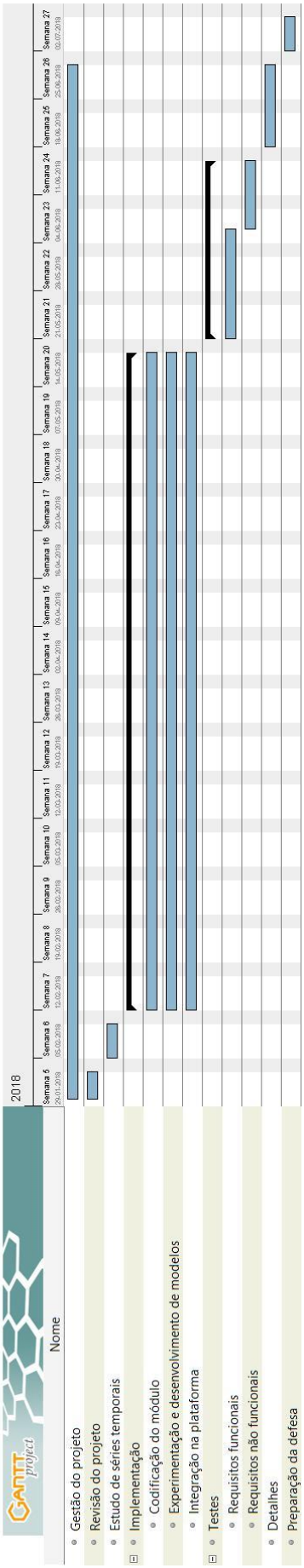


Figura 29 - Diagrama de Gantt real do segundo semestre

Anexo B

Experimentação e Desenvolvimento de Modelos

Este anexo apresenta os resultados detalhados das experiências dos requisitos descritos no capítulo 7 do relatório, com o valor médio e desvio padrão das dez execuções em cada teste.

B.1. Perfil de clientes

Esta secção contém os resultados das experiências do requisito de, determinar o perfil de clientes, apresentadas na secção 7.1 do relatório.

B.1.1. Seleção de atributos

A Tabela 34 apresenta a lista de atributos selecionados pelo threshold do método que calcula a importância dos atributos em relação à classe-alvo.

Campanha	Nº atributos	Atributos
15	11	Saldo médio antes da recarga, quantidade de carregamento a seis meses, média de dias entre carregamentos, ARPU a três meses, média de consumo a três meses, valor acumulado de recarga, desvio de carregamento mensal, número dias entre carregamentos, ARPU a um mês, desvio de carregamento a seis meses, valor de SMS
87	19	ARPU a três meses, média de consumo a três meses, valor de SMS, ARPU a um mês, valor voz, saldo médio antes da recarga, média de dias entre carregamentos, valor voz ORM, quantidade de carregamento a seis meses, quantidade de tráfego SMS, duração voz ORM, duração voz, escalão valor, valor acumulado de recarga, valor voz RF, valor mínimo de recarga, número dias entre carregamentos, duração voz RF, desvio de dias no carregamento a seis meses
28	15	Média de dias entre carregamentos, escalão valor a um mês, valor voz, média de consumo a três meses, número dias entre carregamentos, escalão valor, valor acumulado de recarga, saldo médio antes da recarga, valor de SMS, valor voz RF, desvio de dias no carregamento a seis meses, quantidade de carregamentos a seis meses, duração voz ORM, ARPU a três meses, duração voz RF
75	17	Saldo médio antes da recarga, valor acumulado de recarga, média de consumo a três meses, duração voz, ARPU a um mês, ARPU a três meses, valor voz, quantidade de tráfego SMS, desvio de carregamento a seis meses, média de dias entre carregamentos, quantidade de carregamento a seis meses, valor voz ORM, escalão valor, duração voz ORM, valor mínimo de recarga, número de dias entre carregamentos, valor SMS

Tabela 34 - Atributos selecionados pelo threshold do método

A Tabela 35 apresenta a lista de atributos selecionados por procura exaustiva, depois de obter o ranking de atributos ordenados pela sua importância e de testar diversos subconjuntos.

Campanha	Grupos	Nº atributos	Atributos
15	3	4	Saldo médio antes da recarga, quantidade de carregamento a seis meses, média de dias entre carregamentos, ARPU a três meses,
	4	4	
	5	4	
87	4	8	ARPU a três meses, média de consumo a três meses, valor de SMS, ARPU a um mês, valor voz, saldo médio antes da recarga, média de dias entre carregamentos, valor voz ORM
	3	16	ARPU a três meses, média de consumo a três meses, valor de SMS, ARPU a um mês, valor voz, saldo médio antes da recarga, média de dias entre carregamentos, valor voz ORM, quantidade de carregamento a seis meses, quantidade de tráfego SMS, duração voz ORM, duração voz, escalão valor, valor acumulado de recarga, valor voz RF, valor mínimo de recarga, número dias entre carregamentos, duração voz RF
	5	16	
28	3	4	Média de dias entre carregamentos, escalão valor a um mês, valor voz, média de consumo a três meses
	4	4	
	5	4	
75	3	4	Saldo médio antes da recarga, valor acumulado de recarga, média de consumo a três meses, duração voz
	4	4	
	5	4	

Tabela 35 - Atributos selecionados por procura exaustiva

B.1.2. Precisão dos modelos com os atributos todos e com os relevantes

A Tabela 36 compara a precisão do modelo K-Means com todos os atributos e com apenas os relevantes, para as quatro campanhas. Dentro dos atributos relevantes existem duas versões, a primeira com seleção dos atributos pelo threshold definido pelo método, o segundo com teste de combinações de atributos para encontrar o ponto ótimo.

Campanha	Grupos	Precisão (%)		
		Todos	Relevantes (threshold)	Relevantes (ponto ótimo)
15	3	55.92 ± 0.44	61.79 ± 0.99	63.82 ± 1.01
	4	56.71 ± 1.68	63.87 ± 0.32	66.26 ± 2.51
	5	57.32 ± 2.06	64.80 ± 0.44	66.77 ± 0.73
87	3	53.10 ± 0.93	54.43 ± 2.04	57.53 ± 0.33
	4	52.91 ± 0.84	55.93 ± 0.98	57.59 ± 0.23
	5	54.30 ± 0.41	55.33 ± 0.94	58.19 ± 0.54
28	3	52.20 ± 1.10	56.98 ± 0.62	62.48 ± 0.71
	4	53.66 ± 0.97	58.33 ± 0.62	62.85 ± 1.01
	5	53.30 ± 1.01	58.15 ± 0.45	63.15 ± 0.68
75	3	56.29 ± 1.61	56.54 ± 0.21	59.66 ± 1.52
	4	56.50 ± 1.04	56.95 ± 0.62	59.34 ± 1.63
	5	57.24 ± 1.15	57.51 ± 0.38	61.82 ± 1.32

Tabela 36 - Precisão dos modelos com os atributos todos e com os relevantes

B.1.3. Precisão dos modelos em diferentes métodos

As Tabelas 37, 38, 39 e 40 comparam a precisão do modelo K-Means com a função de distância Euclidean, e do modelo hierárquico aglomerativo com a função Euclidean e Manhattan.

Campanha 15		
Método	**Grupos**	**Precisão (%)**
K-Means	3	63.82 ± 1.01
	4	66.26 ± 2.51
	5	66.77 ± 0.73
Hierárquico aglomerativo (Euclidean)	3	62.38 ± 1.23
	4	64.15 ± 1.97
	5	65.25 ± 2.13
Hierárquico aglomerativo (Manhattan)	3	57.53 ± 2.31
	4	57.53 ± 2.31
	5	57.92 ± 1.34

Tabela 37 - Precisão dos modelos da campanha 15 variando o método

Campanha 87		
Método	**Grupos**	**Precisão (%)**
K-Means	3	57.53 ± 0.33
	4	57.59 ± 0.23
	5	58.19 ± 0.54
Hierárquico aglomerativo (Euclidean)	3	56.64 ± 0.07
	4	57.29 ± 0.47
	5	57.61 ± 0.19
Hierárquico aglomerativo (Manhattan)	3	56.29 ± 0.38
	4	56.32 ± 0.39
	5	56.83 ± 0.59

Tabela 38 - Precisão dos modelos da campanha 87 variando o método

Campanha 28		
Método	**Grupos**	**Precisão (%)**
K-Means	3	62.48 ± 0.71
	4	62.85 ± 1.01
	5	63.15 ± 0.68
Hierárquico aglomerativo (Euclidean)	3	62.15 ± 1.10
	4	62.15 ± 1.11
	5	62.68 ± 1.09
Hierárquico aglomerativo (Manhattan)	3	56.19 ± 2.51
	4	56.56 ± 2.26
	5	59.28 ± 1.18

Tabela 39 - Precisão dos modelos da campanha 28 variando o método

Campanha 75		
Método	**Grupos**	**Precisão (%)**
K-Means	3	59.66 ± 1.52
	4	59.34 ± 1.63
	5	61.82 ± 1.32
	3	59.44 ± 1.58

Hierárquico aglomerativo (Euclidean)	4	59.93 ± 1.90
	5	59.98 ± 1.88
Hierárquico aglomerativo (Manhattan)	3	57.19 ± 0.57
	4	57.70 ± 0.80
	5	59.44 ± 1.84

Tabela 40 - Precisão dos modelos da campanha 75 variando o método

B.1.4. Análise de Silhouette variando o número de grupos

Os gráficos da Figura 30, 31 e 32 apresentam a análise de Silhouette para as restantes campanhas com o método K-Means e com os atributos selecionados por procura exaustiva, para três, quatro e cinco grupos.

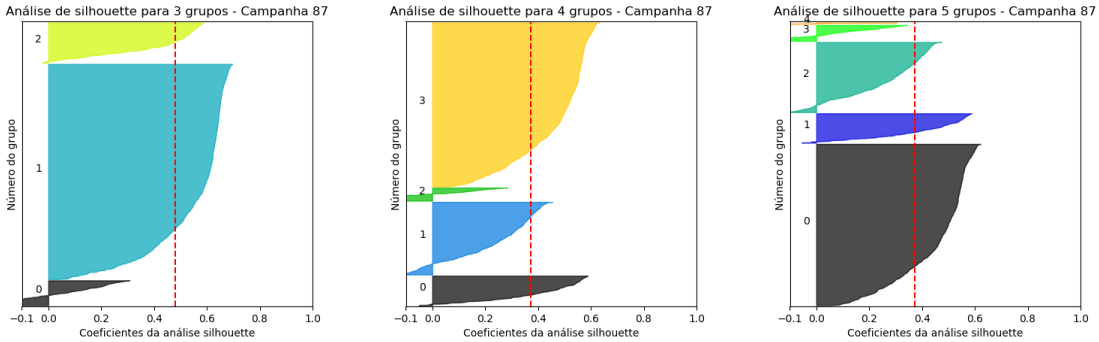


Figura 30 - Análise de Silhouette da campanha 87

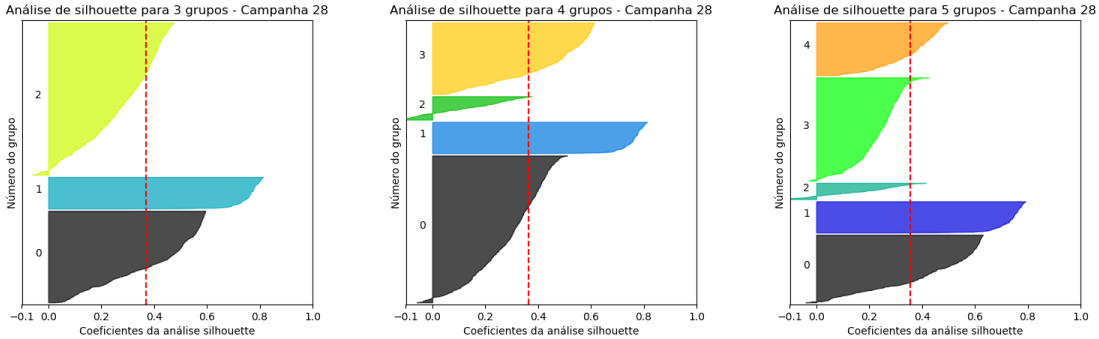


Figura 31 - Análise de Silhouette da campanha 28

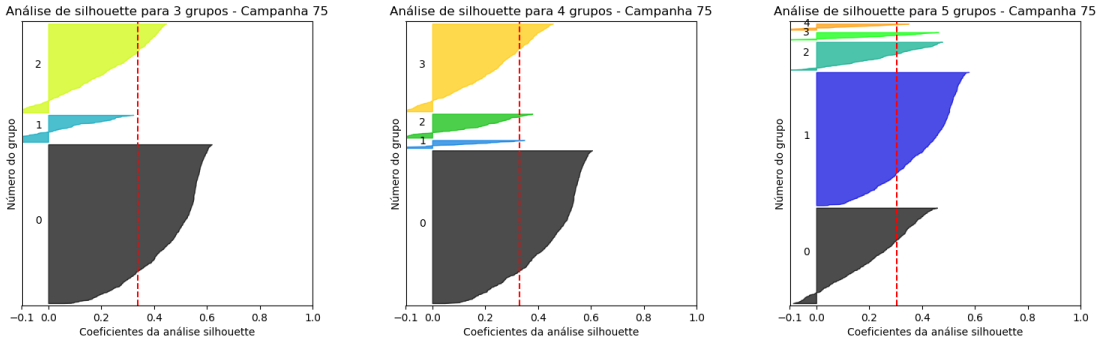


Figura 32 - Análise de Silhouette da campanha 75

B.1.5. Resultados

As Figuras 33 e 34 e Tabelas 41 e 42, apresentam os perfis dos clientes representados pelos valores médios dos atributos relevantes para quatro grupos na campanha 87 e 75 respetivamente.

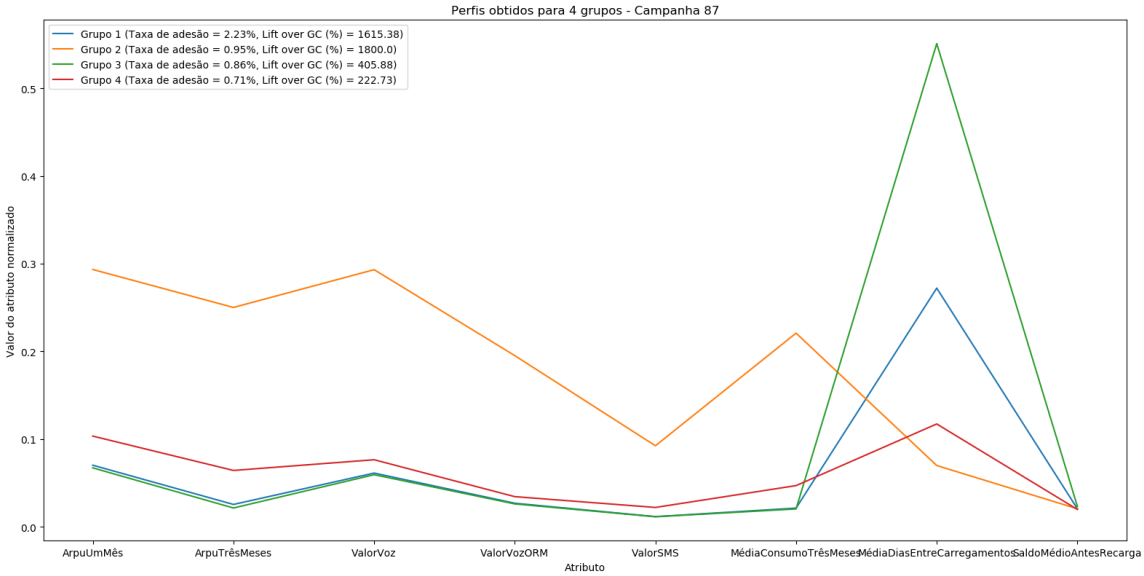


Figura 33 - Gráfico de perfis de clientes obtidos na campanha 87 para quatro grupos

Grupo	1	2	3	4
Atributos de perfil de cliente				
ARPU a um mês	4.92	52.2	4.28	11.94
ARPU a três meses	5.08	49.94	4.27	12.81
Valor voz	1.82	8.71	1.77	2.27
Valor voz ORM	1.98	14.37	1.92	2.52
Valor SMS	1.08	8.64	1.07	2.06
Média de consumo a três meses	5.04	52.37	4.82	11.11
Média de dias entre carregamentos	35.3	9.8	70.42	15.8
Saldo médio antes da recarga	3.86	3.88	7.32	2.91
Métricas da campanha				
Taxa de adesão (universo de teste)	45.65%	76.74%	40.79%	60.79%
Taxa de adesão (universo total)	2.23%	0.95%	0.86%	0.71%
Lift over GC no universo total (global) = 920%				
Lift over GC	1615.38%	1800.0%	405.88%	222.73%

Tabela 41 - Perfis de clientes e métricas para campanha 87 para quatro grupos

Nesta campanha o lift over GC global foi de 910%, a partir dos grupos obtidos no gráfico da Figura 33 e da Tabela 41, verifica-se que o grupo 1 e 2, representam o perfil de clientes que mudaram mais o seu comportamento fase ao grupo de controlo, ou seja, os clientes dependentes do incentivo. Os restantes grupos representam os clientes indiferentes ao incentivo.

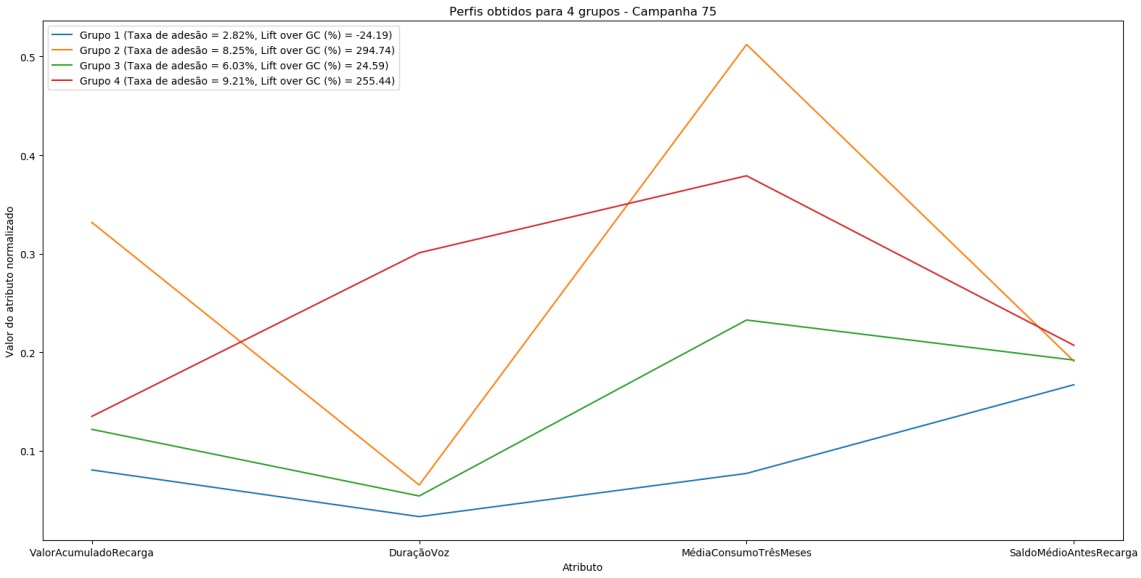


Figura 34 - Gráfico de perfis de clientes obtidos na campanha 75 para quatro grupos

Grupo	1	2	3	4
Atributos de perfil de cliente				
Valor acumulado na recarga	71.7	279.98	105.87	116.76
Duração voz	575.07	1132.39	938.74	5221.06
Média de consumo a três meses	3.97	26.42	11.99	19.56
Saldo médio antes da recarga	0.88	2.24	2.29	3.15
Métricas da campanha				
Taxa de adesão (universo de teste)	43.51%	74.19%	55.61%	72.22%
Taxa de adesão (universo total)	2.82%	8.25%	6.03%	9.21%
Lift over GC no universo total (global) = 84.82%				
Lift over GC	-24.19%	294.74%	24.59%	225.44%

Tabela 42 – Perfis de clientes e métricas para campanha 75 para quatro grupos

Uma vez que o valor do lift over GC global desta campanha é de 84.82%, a partir dos grupos obtidos no gráfico da Figura 34 e da Tabela 42, verifica-se que o grupo 2 e 4 apresentam um valor do lift superior ao global.

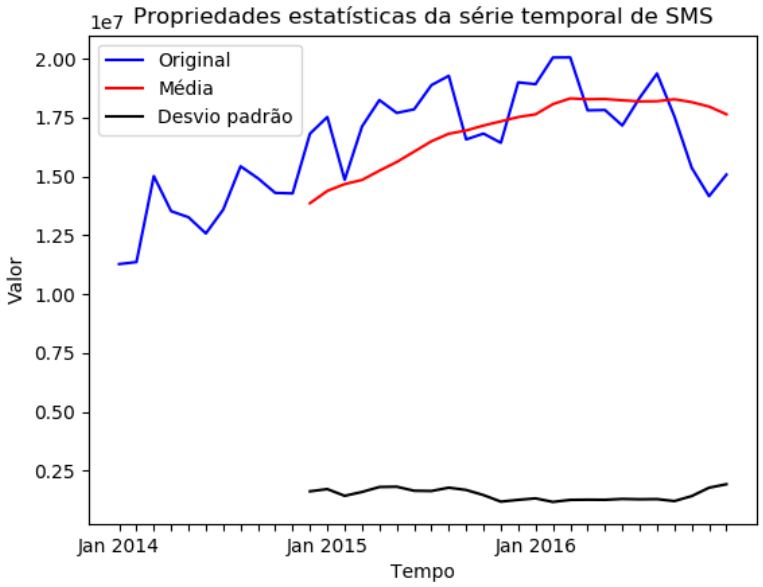
Ou seja, representam o perfil de clientes que mudaram mais o seu comportamento fase ao grupo de controlo, ou seja, clientes dependentes do incentivo. Os restantes, representam o perfil de clientes indiferentes ao incentivo.

B.2. Previsão de consumo

Esta secção apresenta a análise das séries temporais de consumo de SMS e de dados de internet. Finalmente a última secção, apresenta os resultados dos modelos ARIMA descritos na secção 7.2 do relatório variando a sua parametrização.

B.2.1. Análise da série temporal de consumo de SMS

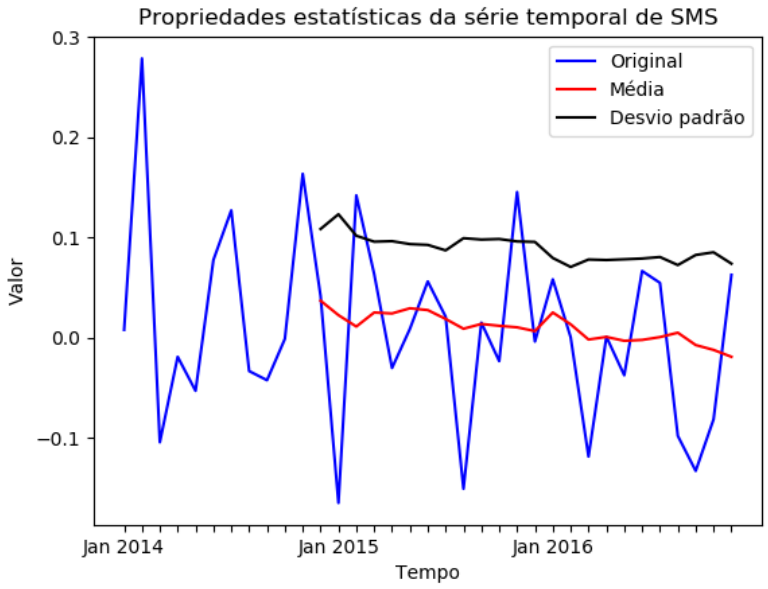
A Figura 35 apresenta o teste de estacionariedade da série temporal de consumo de SMS com recurso a dois métodos. O primeiro por visualização gráfica das propriedades estatísticas da série, a média móvel e o desvio padrão. O segundo através do teste estatístico de Dickey-Fuller [13].




```
Results of Dickey-Fuller Test:
Test Statistic                -2.602418
p-value                        0.092498
#lags Used                     0.000000
Number of Observations Used   35.000000
Critical Value (1%)           -3.632743
Critical Value (5%)           -2.948510
Critical Value (10%)          -2.613017
```

Figura 35 - Teste de estacionariedade da série temporal de SMS original

A partir do gráfico apresentado verifica-se uma clara variação da média ao longo do tempo. O resultado do teste estatístico apresenta um valor maior que o valor crítico com um nível de confiança de 95% (-2.602418 > -2.948510), indicando também que a série é não estacionária no tempo. Para tornar a série estacionária aplicou-se uma diferenciação de primeira ordem. A Figura 36 apresenta o teste de estacionariedade após a transformação.




```
Results of Dickey-Fuller Test:
Test Statistic                -6.167324e+00
p-value                        6.939140e-08
#lags Used                     1.000000e+00
Number of Observations Used    3.300000e+01
Critical Value (1%)           -3.646135e+00
Critical Value (5%)           -2.954127e+00
Critical Value (10%)          -2.615968e+00
```

Figura 36 - Teste de estacionariedade da série temporal de SMS após a transformação

O gráfico anterior mostra que as propriedades estatísticas da série temporal tornaram-se constantes ao longo do tempo. O teste estatístico é menor que o valor crítico para um nível de confiança de 95% (-6.167324 < -2.954127), indicando que a série se aproximou da estacionariedade. Uma vez que se estimou o parâmetro de diferenciação, o próximo passo é determinar os parâmetros p (ordem da componente AR do modelo) e q (ordem da componente MA do modelo) do ARIMA, com recurso às funções ACF e PACF. A Figura 37 apresenta os gráficos destas duas funções.

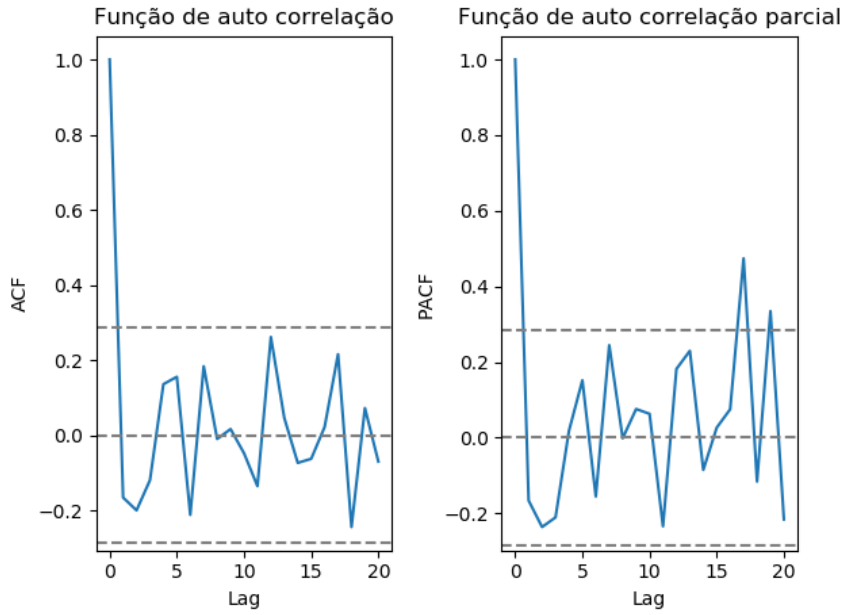


Figura 37 - Gráficos das funções ACF e PACF da série temporal de SMS

O gráfico da função de auto correlação apresentada do lado esquerdo da Figura 37, permite estimar o valor do parâmetro p, no momento em que cruza o intervalo de confiança, ou seja, entre p = 0 e p = 1. O outro gráfico, da função de auto correlação parcial do lado direito, permite estimar o valor do parâmetro q, no momento em que cruza o intervalo de confiança, ou seja, entre q = 0 e q = 1. A Tabela 44 do anexo B.2.3 contém os testes deste modelo com as várias combinações dos parâmetros p e q. O parâmetro d foi configurado com o valor de 1, e o de sazonalidade com o valor de doze, ou seja, o período de sazonalidade estudado. A Figura 38 apresenta os resultados de previsão do modelo ARIMA de forma gráfica, comparando os valores reais e previstos para a melhor combinação de parâmetros encontrada.

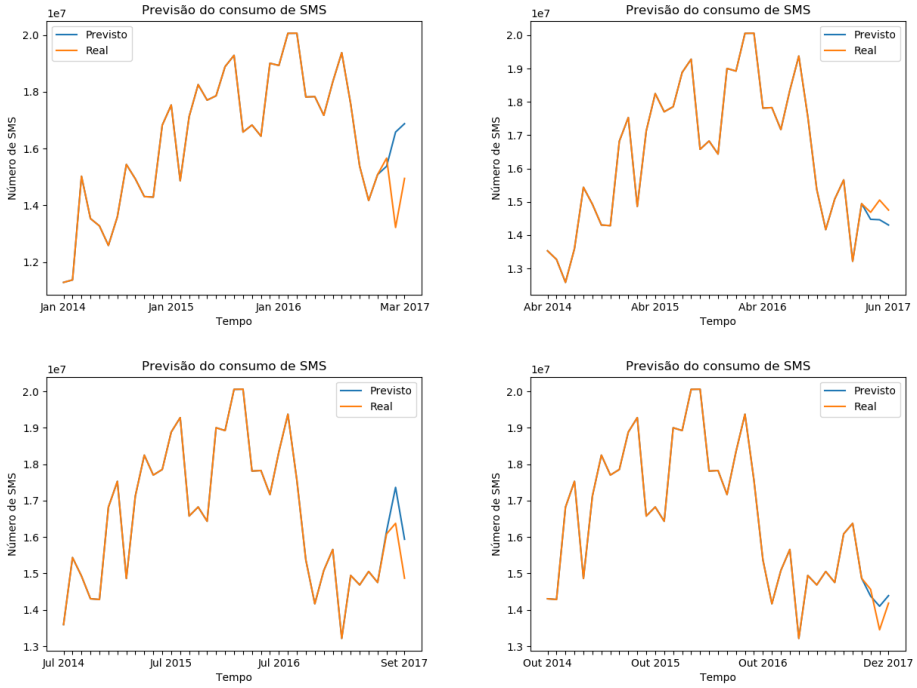
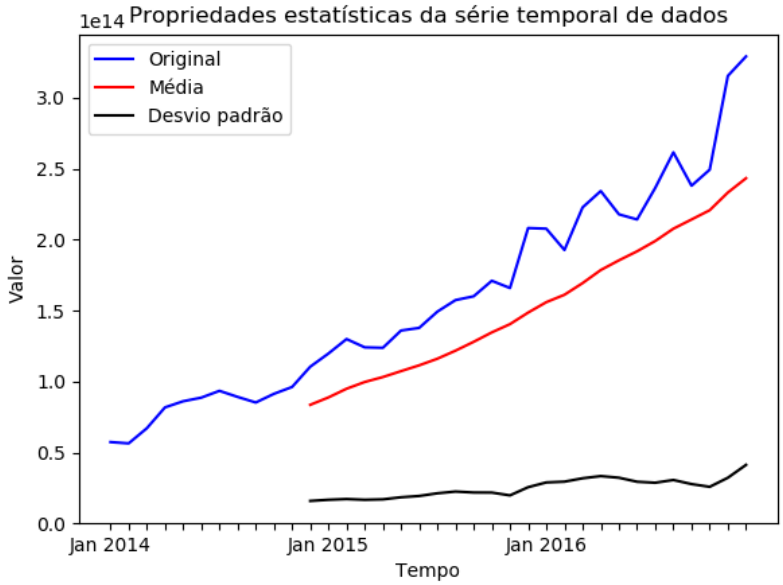


Figura 38 - Previsão de SMS a três meses

Os gráficos das previsões a três meses, mostram uma previsão aproximada boa em comparação com os valores reais.

B.2.2. Análise da série temporal de consumo de dados de internet

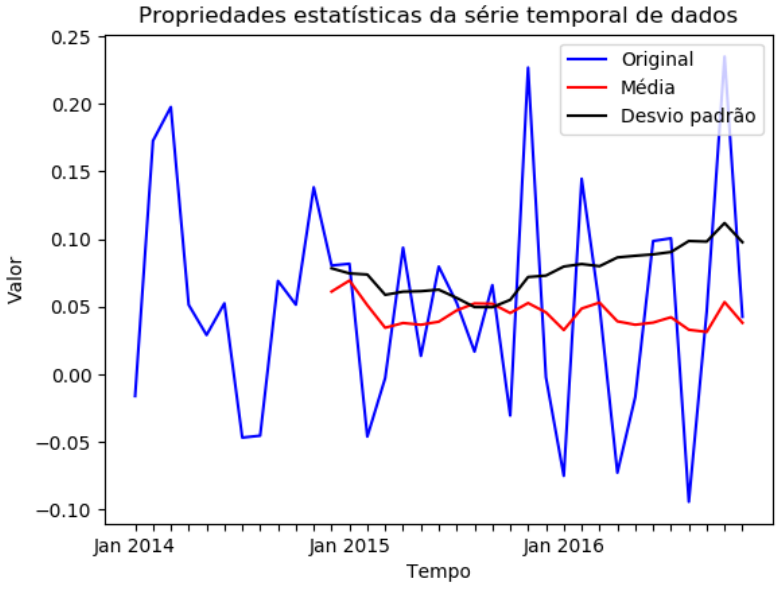
A Figura 39 apresenta o teste de estacionariedade da série temporal de consumo de dados de internet.




```
Results of Dickey-Fuller Test:
Test Statistic                  2.892761
p-value                         1.000000
#lags Used                      6.000000
Number of Observations Used    29.000000
Critical Value (1%)            -3.679060
Critical Value (5%)            -2.967882
Critical Value (10%)           -2.623158
```

Figura 39 - Teste de estacionariedade da série temporal de dados original

Os resultados mostram que a série temporal é não estacionária no tempo. A Figura 40 apresenta o teste de estacionariedade após a aplicação da transformação de diferenciação de primeira ordem.



```
Results of Dickey-Fuller Test:
Test Statistic                -6.150495e+00
p-value                        7.583067e-08
#lags Used                     5.000000e+00
Number of Observations Used    2.900000e+01
Critical Value (1%)           -3.679060e+00
Critical Value (5%)           -2.967882e+00
Critical Value (10%)          -2.623158e+00
```

Figura 40 - Teste de estacionariedade da série temporal de SMS após a transformação

Os resultados mostram que a série temporal está estacionária no tempo. As funções apresentadas nos gráficos da Figura 41, permitem estimar os valores dos parâmetros p (ordem da componente AR do modelo) e q (ordem da componente MA do modelo) do ARIMA.

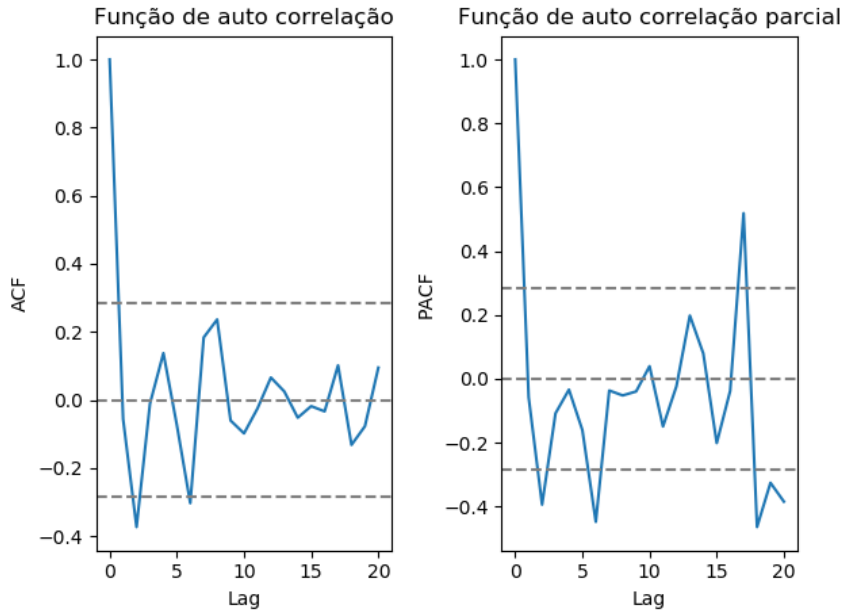
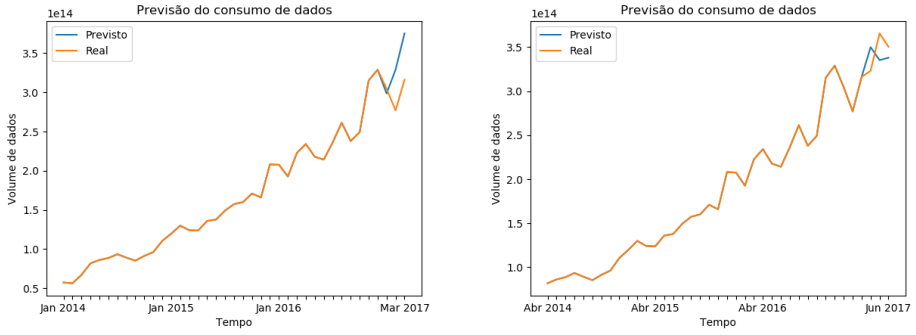


Figura 41 - Gráficos das funções ACF e PACF da série temporal de dados

Os gráficos mostram que os valores dos parâmetros p e q estão entre 1 e 2. A Tabela 45 do anexo B.2.3 contém os testes deste modelo com as várias combinações destes dois parâmetros. O parâmetro d foi configurado com o valor de 1, e o de sazonalidade com o valor de doze, ou seja, o período de sazonalidade. A Figura 42 apresenta os resultados de previsão do modelo ARIMA de forma gráfica comparando os valores reais e previstos para a melhor combinação de parâmetros encontrada.



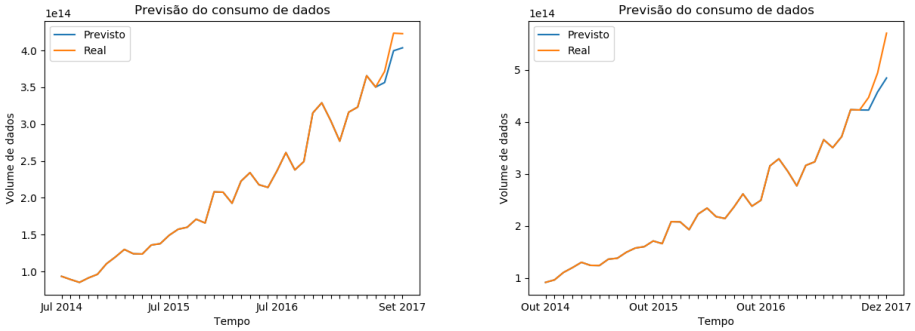


Figura 42 - Previsão de dados a três meses

Também aqui, os gráficos das previsões a três meses, mostram uma previsão aproximada boa em comparação com os valores reais.

B.2.3. Comparação do erro dos modelos ARIMA variando a sua parametrização

As tabelas 43, 44 e 45 mostram o RMSE e MAPE obtido usando o ARIMA com as várias combinações de valores dos parâmetros p e q, estimados usando os gráficos das funções ACF e PACF.

Consumo de recargas				
ARIMA (p, d, q)	**RMSE**		**MAPE**	
	Média	**Máximo**	**Média**	**Máximo**
(0, 1, 0)	3.87e+4 ± 1.57e+4	5.62e+4	3.99 ± 1.54	5.65
(1, 1, 0)	3.86e+4± 1.47e+4	5.29e+4	3.96 ± 1.42	5.21
(0, 1, 1)	3.69e+4± 1.31e+4	4.97e+4	3.71 ± 1.25	5.01
(1, 1, 1)	3.77e+4± 1.19e+4	4.99e+5	3.84 ± 1.01	4.94

Tabela 43 - Comparação dos modelos ARIMA de previsão de recargas

Consumo de SMS				
ARIMA (p, d, q)	**RMSE**		**MAPE**	
	Média	**Máximo**	**Média**	**Máximo**
(0, 1, 0)	7.95e+5 ± 5.5e+5	1.66e+6	4.67 ± 3.09	9.43
(1, 1, 0)	8.34e+5 ± 4.82e+5	1.62e+6	4.89 ± 2.65	9.18
(0, 1, 1)	8.82e+5 ± 4.37e+5	1.59e+6	5.19 ± 2.3	8.95
(1, 1, 1)	8.18e+5 ± 4.6e+5	1.49e+6	4.71 ± 2.63	8.45

Tabela 44 - Comparação dos modelos ARIMA de previsão de SMS

Consumo de dados				
ARIMA (p, d, q)	**RMSE**		**MAPE**	
	Média	**Máximo**	**Média**	**Máximo**
(1, 1, 1)	2.59e+13 ± 1.48e+13	5.12e+13	6.99 ± 5.79	17.0
(2, 1, 1)	3.55e+13 ± 2.18e+13	7.2e+13	8.94 ± 6.81	20.52
(1, 1, 2)	2.84e+13 ± 1.37e+13	5.01e+13	7.17 ± 5.03	15.6
(2, 1, 2)	3.96e+13 ± 1.99e+13	6.57e+13	10.02 ± 6.34	20.23

Tabela 45 - Comparação dos modelos ARIMA de previsão de dados

Anexo C

Artigo: Experimental Comparison and Tuning of Time Series Prediction for Telecom Analysis

Este anexo mostra o artigo submetido na International Conference on Time Series and Forecasting (ITISE 2018).

Experimental Comparison and Tuning of Time Series Prediction for Telecom Analysis

André Pinho (apinho@student.dei.uc.pt), Helena Silva (lena@alticelabs.com), Pedro Furtado (pnf@dei.uc.pt), Ricardo Filipe (ricardo-a-filipe@alticelabs.com)

University of Coimbra, Altice Labs, Portugal

Abstract. Prediction of consumption is fundamental in telecommunications, for efficient management of network resources, and for guaranteeing quality of service. In this work we investigate the use of time series models to forecast consumption. Two time series forecasting algorithms are compared, Auto-Regressive Integrated Moving Average (ARIMA) and Prophet, launched by Facebook in 2017. We also developed a simple automated parameterization solution for ARIMA, which is important in practical deployments. The work described was developed in the context of tool development effort within Altice Labs that provides actual software to associated Telecom operators, in collaboration with University of Coimbra. To validate results we used real data from a Telecom operator. The forecast results showed that ARIMA was better than Prophet with a Mean Absolute Percentage Error (MAPE) of 3.71% in the three-month forecast and 4.14% in the twelve-month forecast.

Keywords: ARIMA, Forecasting, Prophet

1 Introduction

Currently, telecom operators face competition from other operators and from new services made available through the internet. Operators need to be one step ahead of competition, and they need to offer reliable services to avoid migration of customers and a fall in profits. One important opportunity is to create tools to analyze the huge and valuable data that they collect using data science techniques, with great potential for decision support. In this context, consumption forecasting is critical to provide information that helps the operator efficiently plan and manage network resources and provide an improvement in quality of service.

Consumption forecast in telecommunications presents its challenges, as it is necessary to deal with seasonality, trends, and with the variation of the number of clients. Taking into account these challenges, this work developed at Altice Labs in collaboration with University of Coimbra had two objectives. First, to compare two time series forecasting models, ARIMA and Prophet, in order to verify which one best fits the context of telecommunications. Since ARIMA was superior and since it requires manual parameterization of the model, the second objective was to create an automatic parameterization mechanism. This mechanism consists of a set of steps, selection,

adfa, p. 90, 2011.

transformation, parameterization by exhaustive search, and application of the model chosen in forecasting. With the choice of model and the automated parameterization, the approach is ready to be integrated in decision support tools to be used by managers that do not need to know any details of the forecasting model to use it. Instead, they simply view charts with the forecasts they desire, and the approach adjusts automatically the parameterization to changes in behavior of the data series. In terms of software details, the tool was developed in python, and in particular using its time-series libraries statsmodels [1] and fbprophet [2], containing the ARIMA and Prophet methods respectively.

This paper is divided into 7 sections. Section II presents state of the art. Section III discusses how to apply time series forecasting in the context of telecommunications, reviewing the relevant details of ARIMA and Prophet. In section IV we describe the need for manual parameterization of the time series model, and in section V proposes an automatic parameterization approach for ARIMA using exhaustive search. Section VI reports and analyses our experimental results. Finally, section VII concludes and discusses future work.

2 State of the art

In this section we review works on time-series forecasting in the context of Telecommunications that are most related to ours.

In 2008, S. T., & Sampaio, R. J. B proposed a model to predict short-term consumption of a telecommunications service [3]. Since service consumption presents a non-linear behavior caused by the existence of tendency and seasonality, the authors used two neural network algorithms, the Multilayer Perceptron and the Radial Basis Function network (RBF). The per-month dataset was divided into two sets, training set consisting of a history of 3, 4 and 6 months, and test set consisting of only 1 month ahead. The metric used in the evaluation of the results was Mean Squared Error (MSE). From the experiments performed for different historical periods, the Multilayer Perceptron model presented better prediction quality, although with worse computational performance.

In 2015, Wang, M., Wang, Y., Wang, X., & Wei, Z proposed a model based on the Auto-Regressive Integrated Moving Average (ARIMA), with the objective of predicting performance in telecommunications [4]. In this study they used monthly aggregated data corresponding to periods of two and a half years. The data of the first two years was used for analysis of the time series and for training the model, and the remaining six months were used for validation of the model. The method used to verify the seasonality of the time series was the analysis of the statistical properties, mean, variance and correlation coefficient, verifying if they remain constant over time. In our study it was verified that the series was non-stationary, so it had to be adapted using the typical manual procedure we describe later. The model obtained by [4], had an average error of 1% for five months.

In 2017, Hideaki Hayashi compared the performance of Prophet and ARIMA in the different context of prediction of number of flights in the United States [5]. Prophet

was inferior to ARIMA with the parameters configured manually, and superior to ARIMA with the parameters configured automatically with the values by default. The author concluded that the ARIMA method, unlike Prophet, requires manual configuration of the model parameters in order to have good results. This means that the ARIMA requires a lot of knowledge in the domain to be configured manually.

The two works [4] and [5] do show that ARIMA could be the best choice for time-series analysis and forecasting in Telecom and other contexts, but has the big drawback that it requires manual configuration, which is undesirable for integration into a managerial decision support tool as we desired. Furthermore, in our work it was important to evaluate the two alternatives (ARIMA and Prophet) in the context of real Telecom consumption data, and to devise which to integrate into a decision support tool and how to automate its use.

3 Application of time series forecasting in the context of telecommunications

In telecommunications, time series forecasting is typically applied in the forecast of consumption and also in the detection of anomalies in real time. The remainder of this section reviews the concepts of stationarity, the Auto-Regressive Integrated Moving Average (ARIMA) [6] [7] and Prophet [2]. The later was launched by Facebook in 2017 to allow its use by people with less knowledge in the field, since ARIMA requires manual tuning of fundamental parameters.

3.1 Stationarity in ARIMA

An important concept in the application of the ARIMA method is stationarity, since the model can only be constructed with stationary time series. A series is stationary if its statistical properties remain constant over time. The existence of trend and seasonality are two of the reasons that lead the series to be non-stationary [6] [7].

There are two methods that allow you to check whether a series is stationary or not. The first method consists in the graphical visualization of the variation of the statistical properties of the series, such as the moving average (calculation at each instant of the average of the values corresponding to the last seasonal period, typically of twelve consecutive months) and the moving standard deviation over time. If the properties of the series do not change over time, then the series is stationary. The second method, the Dickey-Fuller test, assumes that the null hypothesis is that the series is non-stationary. This test calculates the value of the statistical test and some critical values for different levels of confidence. If the value of the statistical test is less than the critical value, then the series is stationary [6].

Differentiation [6] [7] is one of the existing techniques that allows us to deal with seasonality and trend of the time series, bringing it closer to stationarity in time. At each instant in the series, differentiation subtracts the original observation, Y_t, from that of the previous instant, Y_{t-1}, using the following formula:

$$Y'_t = Y_t - Y_{t-1}$$

3.2 Time series forecasting with ARIMA and with Prophet

This subsection begins by describing the Auto-Regressive (AR) and Moving Average (MA) models, before describing the Auto-Regressive Integrated Moving Average (ARIMA) method.

1. The AR model [7] extracts the influence of the values of the previous periods from those of the current period. This model is developed using the following linear equation.

$$Y_t = \mathrm{c} + \varphi_1 . Y_{t-1} + \ldots + \varphi_p . Y_{t-p} + e_t$$

The parameter p indicates the AR order in the model and represents the delayed time period of the dependent variable. The remaining parameters of the equation, φ which represents the AR coefficient, y, which is the observed value, e the deviation of the series at the current instant, c is a constant [7].

2. The MA model [7] extracts the influence of the error terms from the previous period in the current period. This model is developed using the following linear equation:

$$Y_t = \mathrm{c} + e_t + \theta_1 . e_{t-1} - \ldots - \theta_q . e_{t-q}$$

The parameter q indicates the MA order in the model and represents the delayed forecast errors. The remaining parameters of the equation, θ which represents the MA coefficient, y, the observed value, e the deviation of the series at the current instant, c is a constant [7].

3. The non-seasonal ARIMA model [6] [7] consists of three components, AR, Integrated (I) and MA, each component represented by a positive integer parameter, p, d and q respectively. These three components are combined in the following linear equation:

$$Y_t = c + \varphi_1 . Y_{d\ t-1} + \ldots + \varphi_p . Y_{d\ t-p} + e_t + \theta_1 . e_{t-1} - \ldots - \theta_q . e_{t-q}$$

I component [6], represented by parameter d and indicating the number of times the series has been differentiated to approximate stationary in time; AR component [4] [6], represented by parameter p means the delayed time period, estimated by Autocorrelation Function (ACF); MA component [4] [6] represented by parameter q indicates the order of the MA component and represents the delayed forecast errors, estimated by the Partial Autocorrelation Function (PACF).

It should be noted that parameters p and q are determined when the respective functions, ACF and PACF, cross the upper confidence interval for the first time. The confidence interval of the two functions is calculated as $\pm 1.96 / \sqrt{n}$, where the variable n, corresponds to the size of the historical data [6] [7]. Finally, seasonal ARIMA [7] extends the previous model, combining its components along with the seasonal component.

Time series forecasting with Prophet is more automated, due to its ability to find automatically inflection points in the data originated by changes in trend. A novelty of this method in relation to the previous one is the possibility of accommodating the existence of seasonal festive periods. The method combines three components, the trend, the seasonality and the festive periods, each modelled by some function [8]:

$$\mathrm{y(t)} = \mathrm{g(t)} + \mathrm{s(t)} + \mathrm{h(t)} + \epsilon \mathrm{t}$$

The trend component, g(t), is modelled by a logistic function. The seasonality component, s(t), by a Fourier series. The festive periods, h(t), are adjusted by parameterization in the model. Finally, the error term, εt, represents the changes originated by circumstances that are not accommodated by the model [8]. Further information on the formulation details of each of these components can be found in [8].

4 Manual Parameterization of the Time Series

Figure 1 shows real data from a Telecom company graphically. It includes the variation of the number of recharges (used in telecommunications services), and the volume of internet data consumed (data) over the period from 2014 to 2017. There is a tendency of recharge decrease and data consumption increase. Seasonality exists there as temporary increases and decreases in certain months of each year. In the series of recharges, there is a lower consumption in the months of February, October and November, and a higher consumption in the months of January, August and December. First 3 years were used for analysis and training, the fourth year for prediction testing.

Figure 2 shows the stationary test of the time series of recharges using 2 methods. The figure above shows graphical visualization of moving average and deviation. The second, shows Dickey-Fuller statistical test presented below the figure.

Fig. 1. Consumption variation over 4 years

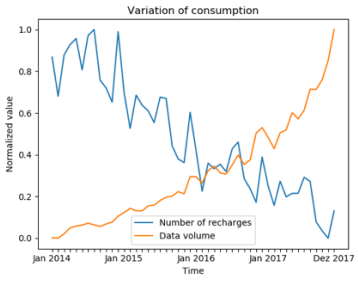
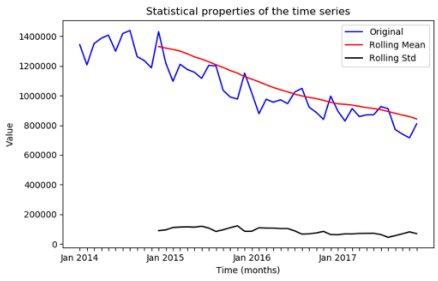


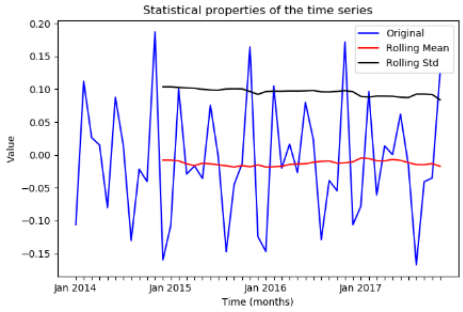
Fig. 2. Stationarity test of time series




```
Results of Dickey-Fuller Test:                  Number of Observations Used     29.000000
Test Statistic                 -1.149969        Critical Value (1%)            -3.679060
p-value                         0.694740        Critical Value (5%)            -2.967882
#lags Used                      6.000000        Critical Value (10%)           -2.623158
```

We can see a decrease of the average over time. This variation over time indicates that the series is non-stationary. The result of the statistical test has a value greater than the critical value with a confidence level of 95% (-1.149969> -2.967882), also indicating that the series is non-stationary in time. Then a first-order differentiation transformation was applied to make the time series stationary in time. Figure 3 presents the stationary test of the series after this transformation.

Fig. 3. Stationarity test of the transformed time series

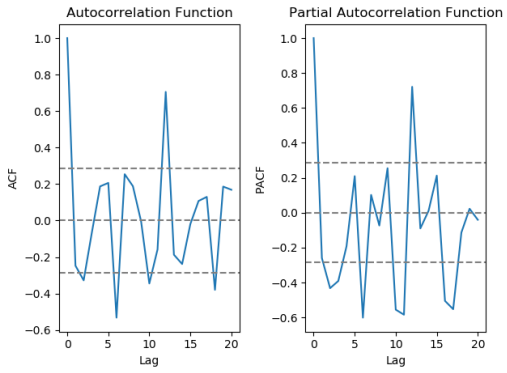



```
Results of Dickey-Fuller Test:
Test Statistic                  -4.815318
p-value                          0.000051
#lags Used                      10.000000
Number of Observations Used     24.000000
Critical Value (1%)             -3.737709
Critical Value (5%)             -2.992216
Critical Value (10%)            -2.635747
```

From the analysis of the previous graph, it is verified that the statistical properties of the time series have become approximately constant over time. The result of the statistical test is also less than the critical value for a 95% confidence level (-4.815318 <-2.992216), also indicating that the time series approached stationarity. Then the parameter d of the model is equal to one, which is the number of differentiations that allowed the series to become approximately stationary.

The next step is to determine the values of the parameters p (order of the AR component of the model) and q (order of the MA component of the model) of ARIMA. To estimate these two parameters, we used the functions Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). Figure 4 shows the graphs of these two functions.

Fig. 4. Graphs ACF and PACF



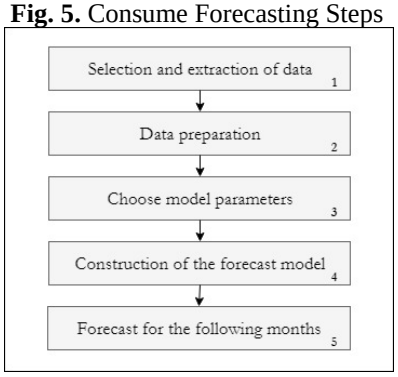
The graph on the left side of Figure 4, representing the ACF function, allows estimation of the value of parameter p, by looking at the position where the function crosses the confidence interval for the first time. In the graph this happens between p = 0 and p = 1. The graph on the right side is the PACF function, which allows to obtain the value of parameter q, also by crossing the function with the confidence interval. In the graph above, it happens between q = 0 and q = 1. In the experimental section we will compare the performance achieved by the forecasting model for this dataset using each of four possible parameter combinations (p = 0, d = 1, q = 0), (p = 1, d = 1, q = 0), (p = 0, d = 1, q = 1) and (p = 1, d = 1, q = 1).

Note that Prophet is much simpler to use than ARIMA, because of its ability to automatically find inflection points in the data, that is, points where trend changes [6].

5 Automation of the ARIMA model using exhaustive search

Automatic parameterization of the seasonal ARIMA model is important to allow its integration in a tool that can be used by managers without requirement of manual configuration or even knowledge of the details. It also accommodates changes in the behavior of the series. The proposed approach follows the steps shown in Figure 5.

Fig. 5. Consume Forecasting Steps



1. Selection of data, extraction and aggregation by month of a three-year historical period, related to the data that is intended to be forecasted.

2. Data preparation, subdivided into three tasks. The first converts the original date format to the format required by the template. The second constructs the input data structure in two-column model with date and the data consumption. The third applies a logarithmic transformation required by the method, which allows to attenuate the trend of the time series [6].

3. Choice of model parameters: This step aims to test and evaluate several models in order to choose the one that minimizes the forecast error. In this step, all combinations of values for parameters p, q and d are generated. The first two parameters can be zero, one or two, and parameter d can be zero or one [6]. This defined range of values takes into account manual parameterization tests as the ones presented in our experimental results, and also a certain flexibility to adapt to future data. The seasonality parameter is defined as 12 (yearly), corresponding to the seasonal period. After generating the combinations of parameters, each model instantiated is tested with historical values. The stationarity of the time series is a requirement for the application of the ARIMA method. Therefore, when the model is constructed with a non-stationary time series, the combination of parameters tested in this iteration is discarded, and the test advances to the next parameter combination.

4. Construction of the forecast model: after the choice of parameters, the forecast model is constructed with a recent three-year history, and with the parameters identified in the previous step.

5. Forecast of consumption of the following months: starts after the construction of the chosen model and consists of predicting the following months. The expected values are converted back to the scale of the original values by performing an exponential transformation (reverse operation of the logarithmic transformation).

6 Experimental Analysis

In the experiments presented in this section, we used real telecommunication data to compare the accuracy of seasonal ARIMA and Prophet methods when forecasting Telecom data, and to validate the parameterization mechanism of the proposed model.

6.1 Experimental Setup

The data used in this evaluation comes from a medium-sized telecommunications operator. Data was aggregated by month for a time period of four years. It consisted of consumption data, recharges (used in telecommunications services) and the volume of internet data consumed (data). Those are the same datasets already described in section IV. For both datasets we used the first three years for training and the fourth for testing the model.

The performance test (evaluation of execution time) was performed on a development machine with the following characteristics:

- Operating system: Windows 8 de 64 bits
- Processor: i5 de 2.50 GHz
- Memory RAM: 8.00 Gb
- Disk HDD: 297 Gb

6.2 Comparison between ARIMA and Prophet on Recharge dataset

Table 1 shows the RMSE and MAPE obtained using ARIMA with different combinations of parameter values, and Table 2 compares the forecasting errors of ARIMA and Prophet, choosing the best ARIMA result.

Tab. 1. Comparison of ARIMA errors on Recharges, 12 months

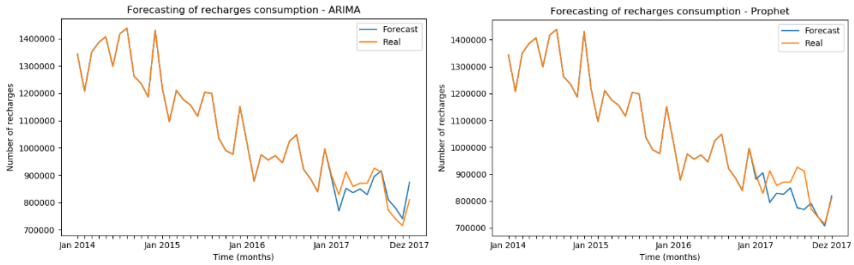
ARIMA (p, d, q)	RMSE	MAPE
(0, 1, 0)	4.08e+4	4.28
(1, 1, 0)	4.04e+4	4.24
(0, 1, 1)	3.99e+4	4.19
(1, 1, 1)	3.96e+4	4.14

Tab. 2. ARIMA versus Prophet on Recharges, 12 months

Method	RMSE	MAPE
ARIMA (1, 1, 1)	3.96e+4	4.14
Prophet	5.59e+4	6.3

From these results we can see that parameter configuration in ARIMA improves the error (RMSE or MAPE), and that ARIMA forecasting errors are much lower than Prophet for the Recharge dataset. Figures 6 show the forecast results graphically, comparing real to predicted values.

Fig. 6. Recharges Forecasting with ARIMA



From visual inspection of the results graphs we can see that both methods are able to deal effectively with both seasonality and trends. It is also clear that ARIMA outperforms Prophet for this dataset.

6.3 Comparison of various configurations

Table 3 presents the results of forecasting 12 months of internet data consumption using ARIMA and Prophet.

Tab. 3. ARIMA vs Prophet on Data Consumption, 12 months

Method	RMSE	MAPE
ARIMA	3.99e+13	8.50
Prophet	2.23e+13	9.88

Table 4 shows the comparison results for recharges and for data consumption (consume) over 3 months. In this case the training data was a whole year and the forecasting covered the next 3 months. Four forecasting runs were used, corresponding to the 4-year trimesters, the tables show the average and standard deviation of the errors over the four runs.

Tab. 4. ARIMA vs Prophet on Recharges and Data Consumption, 3 months

Method	RMSE		MAPE	
	Mean	Maximum	Mean	Maximum
ARIMA - Recharges	3.69e+4	4.97e+4	3.71	5.01
Prophet - Recharges	7.63e+4	1.05e+5	7.82	11.08
ARIMA - Data	2.59e+13	5.12e+13	6.99	17.0
Prophet - Data	4.67e+13	6.35e+13	8.11	13.26

We can see from the tables that Data consumption was slightly more challenging than Recharges for both methods (higher average error). ARIMA was always better

than Prophet (lower average error, almost half for both datasets), although ARIMA results have a higher standard deviation when compared with Prophet. Note that while for Recharges the sum of average error plus standard deviation or the maximum error were always smaller for ARIMA, in the case of Data Consumption one of the trimesters had the highest error for ARIMA, maximum error = 17% against 13% for Prophet, and the average error plus standard deviation is slightly higher for ARIMA. This was however an exception, as we could see by the results ARIMA was consistently better than Prophet for all other 12 and 3 months forecasting cases for both datasets.

6.4 Validation of automated ARIMA configuration

This section presents the validation results of the automatic parameterization mechanism of the ARIMA model described in section V. Table 5 shows the error of each combination tested by the exhaustive search approach regarding the 3-month and 12-months forecast, for both datasets tested. The exhaustive search parameter configuration approach obtains these errors for all cases and then chooses the one with lowest error, which is shown in bold. In all the tests it is verified that the implemented mechanism was able to automatically find the combination of parameters that minimizes the prediction error in the historical data.

The parameters chosen by the automatic approach for these 4 cases (Recharges and Data Consumption, 12 and 3 months forecasting) were always consistent with the established manual parameter configuration found (discussed and exemplified in sections III and IV). Note that the established manual approach involves iterative human inspection, running the Dickey-Fuller test or choosing visually, then successively differentiating and again testing using Dickey-Fuller or visual inspection until the decision thresholds are met. The exhaustive search simply replaces that tedious process with the automated version, with good results for these datasets and forecast objectives.

Another relevant issue is the runtime of the exhaustive search procedure, since it has to test a significant number of alternatives. We test the procedure runtime next.

Tab. 5. Automated parameter values finding by exhaustive search, 3 and 12 months

ARIMA	Recharges (RMSE)		Data consumption (RMSE)	
(p, d, q)	3 months	12 months	3 months	12 months
(0, 0, 0)	Discarded	Discarded	Discarded	Discarded
(0, 0, 1)	Discarded	Discarded	Discarded	Discarded
(0, 0, 2)	Discarded	Discarded	Discarded	Discarded
(0, 1, 0)	3.87e+4	4.08e+4	3.98+e13	5.67e+13
(0, 1, 1)	**3.69e+4**	3.99e+4	4.02+e13	4.82e+13
(0, 1, 2)	3.75e+4	4.03e+4	3.84+e13	5.89e+13
(1, 0, 0)	Discarded	Discarded	Discarded	Discarded
(1, 0, 1)	Discarded	Discarded	Discarded	Discarded
(1, 0, 2)	Discarded	Discarded	Discarded	Discarded
(1, 1, 0)	3.86e+4	4.04e+4	4.18e+13	5.13e+13

(1, 1, 1)	3.77e+4	**3.96e+4**	**2.59e+13**	**3.99e+13**
(1, 1, 2)	3.75e+4	4.03e+4	2.84e+13	5.55e+13
(2, 0, 0)	Discarded	Discarded	Discarded	Discarded
(2, 0, 1)	Discarded	Discarded	Discarded	Discarded
(2, 0, 2)	Discarded	Discarded	Discarded	Discarded
(2, 1, 0)	3.76e+4	4.06e+4	3.94e+13	6.48+e13
(2, 1, 1)	3.88e+4	3.97e+4	3.55e+13	7.36+e13
(2, 1, 2)	6.58e+4	4.01e+4	3.96e+13	1.19e+14

Table 6 shows the runtime of the automatic procedure (testing a significant set of parameter values alternatives) and compares it to the runtime of a single test as a ground truth. Note that the single test run is only used here for comparative reference, because running a single test automatically is not sufficient to configure the parameters, it must be done by a user inspecting the result, and most of the times, as happened for the datasets tested in this experimental section, will require further iterations of differentiating and running the test again.

Tab. 6. Runtime of parameter configuration by automated exhaustive search

Approach	**Time (s)**	
	Mean	**Maximum**
Recharges Forecast (automated)	6.38 ± 0.11	6.57
Data consumption (automated)	6.21 ± 0.39	6.92
Single test	0.30 ± 0.16	0.55
Single test	0.61 ± 0.03	0.66

These results show that the exhaustive search for parameter configuration takes about 6 secs, which is perfectly acceptable for the practical purposes of the decision support tool we were developing. The user has to wait for only 6 secs before the forecasting model does the forecast, since it is searching for the correct parameters. Nevertheless, we note that, although the procedure works fine for the sizes of datasets that our tool works with, it is important to develop improved parameter finding approaches in the future, to be able to handle much bigger datasets. Therefore, we identify as future work the possibility of improving the automated parameter configuration procedure. A simple way to scale to large datasets would involve sampling, to reduce the dataset to a size that is tractable by exhaustive search, the other alternative would be to apply heuristics to reduce the search space, and a third alternative would involve both. We reserve this study for future work.

7 Conclusion and future work

In this work we studied the application of time series forecasting methods ARIMA and Prophet to real Telecom data, with the aim of integrating the best performing one in a practical tool for decision support. The two methods were presented along with their parameterization processes. These two methods were compared in the forecast of

consumption using real telecommunications data. Since the ARIMA model requires a great deal of knowledge in its parameterization, an automatic parameterization procedure was proposed and validated. This allows the approach to do every step of the data forecasting pipeline automatically. This way the approach was integrated into a tool used by managers to view the forecasts without requiring any knowledge of the data preparation and parameterization process for ARIMA forecasting.

This work also showed that, despite the great variation of consumption during the year, due to the existence of seasonality and trends, it is possible to make approximate forecasts with consumption data in the telecommunications area. From the results obtained and the comparison of the two time series methods, it was possible to obtain a minimum MAPE of 3.71% in the three-month forecast and of 4.14% in the twelve-month forecast. Of the two methods tested, the ARIMA model presented better prediction results in relation to Prophet.

For future work we intend to develop models of the Holt-Winters time-series method and compare with the methods studied in this work. In addition, we intend to investigate alternatives that improve the exhaustive search method in the automatic parameterization of ARIMA, using heuristics.

References

[1] StatsModels Statistics in Python, available in http://www.statsmodels.org/devel/generated/statsmodels.tsa.arima_model.ARIMA.html, consulted in 2018-03-20.

[2] Prophet Forecasting at scale, available in https://facebook.github.io/prophet/, consulted in 2018-04-16.

[3] BRANCO, S. T., & DE SAMPAIO, R. J. B. A New Artificial Neural Networks Forecast Model in Telecommunications, consulted in 2017-10-28.

[4] Wang, M., Wang, Y., Wang, X., & Wei, Z. (2015). Forecast and Analyze the Telecom Income based on ARIMA Model. The Open Cybernetics & Systemics Journal, 9(1), consulted in 2018-03-15.

[5] Is Prophet Really Better than ARIMA for Forecasting Time Series Data, available in https://blog.exploratory.io/is-prophet-better-than-arima-for-forecasting-time-series-fa9ae08a5851, consulted in 2018-05-16.

[6] A comprehensive beginner's guide to create a Time Series Forecast, available in https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/, consulted in 2018-03-15.

[7] MATH6011: Forecasting, available in https://www.southampton.ac.uk /~abz1e14/papers/Forecasting.pdf, consulted on 2018-03-22.

[8] Taylor, S. J., & Letham, B. (2017). Forecasting at scale. The American Statistician, (just-accepted), consulted in 2018-04-16.

[9] MAE and RMSE - Which Metric is Better, available in https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d, consulted in2018-03-20.