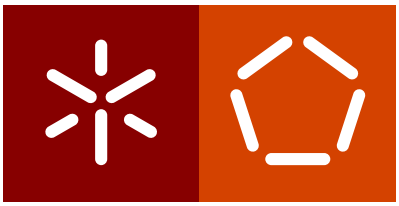


Universidade do Minho

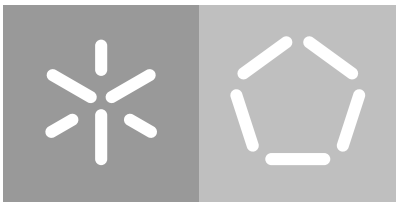
Escola de Engenharia

Departamento de Sistemas de Informação

Carlos Jorge Bravo Silva

Sistema de Apoio à Decisão sobre Transportes Urbanos

Novembro de 2019



Universidade do Minho
Escola de Engenharia
Departamento de Sistemas de Informação

Carlos Jorge Bravo Silva

Sistema de Apoio à Decisão sobre Transportes Urbanos

Dissertação de Mestrado
Mestrado em Engenharia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação do
Professor Doutor Paulo Alexandre Ribeiro Cortez

Novembro de 2019

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

AGRADECIMENTOS

Concluindo com este trabalho o percurso do mestrado, é necessário agradecer a todos aqueles que possibilitaram a concretização deste objetivo.

Em primeiro lugar devo agradecer ao Professor Paulo Cortez que como orientador esteve sempre disponível para ajudar com todas as questões que foram colocadas ao longo da dissertação e também lhe agradeço como professor por transmitir o conhecimento necessário sobre sistemas inteligentes que foi aplicado ao longo deste projeto.

Devo também agradecer aos Transportes Urbanos de Braga por terem disponibilizado os dados necessários para a realização desta dissertação e à sua equipa de engenheiros que ajudaram na compreensão tanto dos dados como do negócio, em especial ao Engenheiro Rui por se ter sempre disponibilizado para ajudar com qualquer dúvida.

Quero também agradecer a toda a equipa do LabSecIoT não só por terem fornecido a infraestrutura utilizada durante o desenvolvimento mas também por serem bons companheiros durante a realização deste projeto de dissertação e também pelas ajudas e opiniões que foram dando acerca do projeto.

E um grande obrigado a toda a minha família que sempre me apoiou, em especial aos meus pais que sem eles nada disto seria possível.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

RESUMO

Com o aumento do número de veículos nas estradas o engarrafamento de trânsito nas zonas urbanas tem-se tornado um problema. Os engarrafamentos levam a prejuízos, poluem o ambiente e causam riscos para a saúde pública. Existe uma necessidade de gerir o trânsito de forma a evitar o congestionamento nas vias sem aumentar o número de infraestruturas, o que se tem demonstrado desafiador. Para combater estas dificuldades têm sido desenvolvidas novas soluções de gestão de trânsito, como a que será apresentada nesta dissertação.

Para este projeto de dissertação os Transportes Urbanos de Braga (TUB) forneceram dados relativos às suas rotas desde 2016, até ao presente ano. O que se pretende com esta tese é recorrendo a técnicas de Data Mining, alimentados com estes dados e com dados de datasets complementares sobre o ambiente, como por exemplo um calendário de eventos na cidade, se consiga prever o comportamento do trânsito, de modo a otimizar as rotas feitas pelos autocarros dos Transportes Urbanos de Braga (TUB), reduzindo assim gastos em combustível, baixando a poluição e ter horários mais precisos para os utilizadores destes transportes.

Palavras-Chave: Data Mining, Machine Learning, Smart City, Smart Mobility, Traffic Prediction

ABSTRACT

With the raise of the number of vehicles on the roads, traffic jams on urban areas became a problem. Traffic jams lead to financial losses, pollute the environment and cause risks to the public health. There is a necessity to manage traffic in order to avoid road congestion without increasing the number of infrastructures, which has been challenging. To combat these difficulties new traffic management solutions have been developed, like the one that will be presented in this dissertation.

For this dissertation project the TUB provided data relative to their routes since 2016, until the present year. The objective of this thesis is using Data Mining techniques, powered by this data and with data from complementary datasets about the environment like, for example a calendar with the city events, we will be able to predict the behavior of the traffic in the following days in order to optimize the TUB bus routes accordingly with what is expected, reducing the fuel costs, lowering the pollution levels and have more accurate schedules for the costumers of these transports.

Keywords: Data Mining, Machine Learning, Smart City, Smart Mobility, Traffic Prediction

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

DM	Data Mining.
GPS	Sistema de Posicionamento Global.
GTFS	General Transit Feed Specification.
IA	Inteligência Artificial.
IoT	Internet of Things.
KNN	K-Nearest Neighbors.
ML	Machine Learning.
SQL	Structured Query Language.
SVM	Support Vector Machine.
TUB	Transportes Urbanos de Braga.
UTM	Sistema Universal Transverso de Mercator.

1

Introdução

1.1 Enquadramento e Motivação

A previsão do fluxo de trânsito é um problema crítico nos sistemas de transportes inteligentes (Zhao et al., 2017). Hoje em dia os carros e outros veículos não são apenas uma comodidade, são praticamente uma necessidade. São tantos e tão comuns que uma família geralmente tem mais que um, na verdade existe à volta de um carro por pessoa mesmo quando a maioria pode transportar múltiplos passageiros. Com o desenvolvimento da economia social, o número de veículos nas cidades está a aumentar drasticamente e as estradas estão a ficar sem capacidade para o acompanhar, originando engarrafamentos de trânsito (Zhao et al., 2017). O setor de transportes é relevante no contexto deste trabalho devido ao crescente número de veículos nas grandes cidades. Tal facto torna uma boa gestão de tráfego, uma tarefa desafiadora no âmbito do contexto das Smart Cities (Ning et al., 2017).

Sem dúvida que esta revolução da mobilidade veio trazer um novo conforto e possibilitou a partilha de conhecimentos e experiências mesmo antes da Internet se tornar um meio de comunicação popular. Estes meios de locomoção proporcionaram também uma maior facilidade de movimentação para os locais de trabalho, que agora podiam ser mais longe de casa. No entanto esta mobilidade, ao longo do tempo foi tornando-se talvez demasiado difundida. Vários veículos ocupam as estradas ao mesmo tempo, nas horas de maior afluência, as chamadas horas de ponta, chega-se a ter atrasos enormes, dezenas de vezes superiores ao tempo necessário para percorrer aquele troço de estrada.

Todos os anos o engarrafamento de trânsito causa prejuízos de vários milhões em combustível, poluição e redução de produtividade (Ma et al., 2015). Para uma empresa de transportes de passageiros, neste caso em particular de autocarros, o engarrafamento de trânsito é um verdadeiro problema. As rotas são planeadas de acordo com um dia normal, e espera-se que o autocarro chegue à hora marcada na paragem. Contudo, podem existir dias anómalos de aumento de intensidade de tráfego. Por exemplo, se um grande número de utilizadores da via decidir ficar em casa porque no dia anterior foi feriado e no dia seguinte

é fim-de-semana, as chamadas pontes, é expectável que o trânsito não seja tão concentrado o que pode levar um autocarro a adiantar-se. Por outro lado, se existe um jogo de futebol na área e os adeptos da zona estão a dirigir-se ao estádio, o trânsito naquela zona vai ser provavelmente mais concentrado, o que pode levar ao autocarro atrasar-se. Este tipo de imprevistos ocorre mais frequentemente do que se possa pensar. Existem constantemente mudanças na via, de uma maneira geral, todos os dias ocorrem acidentes, todos os dias em alguma parte da cidade há um evento e todos os anos há feriados no calendário, sendo que o clima varia constantemente, que por si também influência o fluxo de trânsito.

Por vezes não se pensa nos transportes públicos como um negócio, apenas são vistos como um serviço que está sempre ali disponível, mas na verdade cada pessoa que espera na paragem é um cliente. Este cliente se frequentemente perder o autocarro ou chegar atrasado por causa do mesmo vai potencialmente procurar alternativas. Talvez comece a usar o seu carro, o que irá agravar o problema de trânsito. É necessário otimizar as rotas oferecidas para que estes problemas sejam minimizados.

Neste trabalho, propõe-se o uso de técnicas e metodologias de Data Mining, que podem ser aplicadas a diferentes conjuntos de dados (Kantardzic, 2011), sobre dados históricos, de modo que seja possível obter um modelo que consiga, sabendo que tipo de eventos e tempo irão ocorrer no futuro ou estão a ocorrer no presente, indicar o trânsito que se fará sentir de forma a que os TUB possam escolher a melhor rota a fazer para servir todos os clientes e ao mesmo tempo não os frustrar com imprevistos. Espera-se que os modelos obtidos por esta abordagem façam parte do núcleo do sistema de apoio à decisão a desenvolver para os TUB.

1.2 Objetivos e Resultados Esperados

O principal objetivo deste projeto de dissertação é o desenvolvimento de um sistema de apoio à decisão para os TUB. Este sistema deve conseguir prever o tipo de trânsito na cidade de Braga, tendo em conta o impacto de determinadas ocorrências, nomeadamente meteorologia, feriados e eventos na cidade.

Este projeto vai ter como núcleo, um sistema que recorrendo a Data Mining, vai utilizar os dados históricos que foram fornecidos pelos TUB para determinar o que, em condições semelhantes, poderá ocorrer no futuro, conseguindo assim auxiliar na tomada de decisão de forma a evitar congestionamentos que ocorram em pontos não cruciais das rotas tomadas. É esperado que este sistema possa ser adaptado tanto para funcionar com novos dados como também para funcionar com dados que cheguem a ele em tempo real obtendo-se assim uma previsão mais precisa.

Para além dos já mencionados dados dos TUB, existe a oportunidade de enquadrar este projeto numa vertente de SmartCity. Trabalhando este projeto dentro do LabSecIoT que tem um sistema chamado Edge, o valor deste projeto é aumentado. O Edge é assim chamado porque é baseado em Edge Computing, de uma maneira simples faz a computação dos dados nos locais onde são recolhidos. A maneira como este projeto vê o Edge é simples, várias parecerias com câmaras, empresas de recolha de lixo, de transportes (inclusive os TUB), fazem com que o Edge contenha dados muito variados relativos a cidades especialmente dados recolhidos por veículos. Assim sendo poder-se-á saber com mais precisão por exemplo que temperatura e que humidade está numa estrada em específico, o que pode influenciar o trânsito.

Aproveitando assim esta oportunidade, o produto final esperado é a criação de mais um módulo para o Edge que contenha informações sobre o fluxo de trânsito especialmente em rotas dos TUB, sendo para o efeito de obtenção desta informação realizado o Data Mining dos dados fornecidos pelos TUB para a criação do modelo e utilizados os dados em tempo real existentes no Edge (onde estão também dispositivos Internet of Things (IoT) colocados nos autocarros dos TUB, assumindo a evolução planeada para este sistema) para a previsão do que vai acontecer tanto a curto prazo como a longo prazo. A contribuição deste trabalho (elemento contornado a azul) para o Edge está representada na Figura 1. Espera-se que este serviço a desenvolver possa ser pelo menos um molde daquilo que será o futuro com os dispositivos conectados numa Smart City.

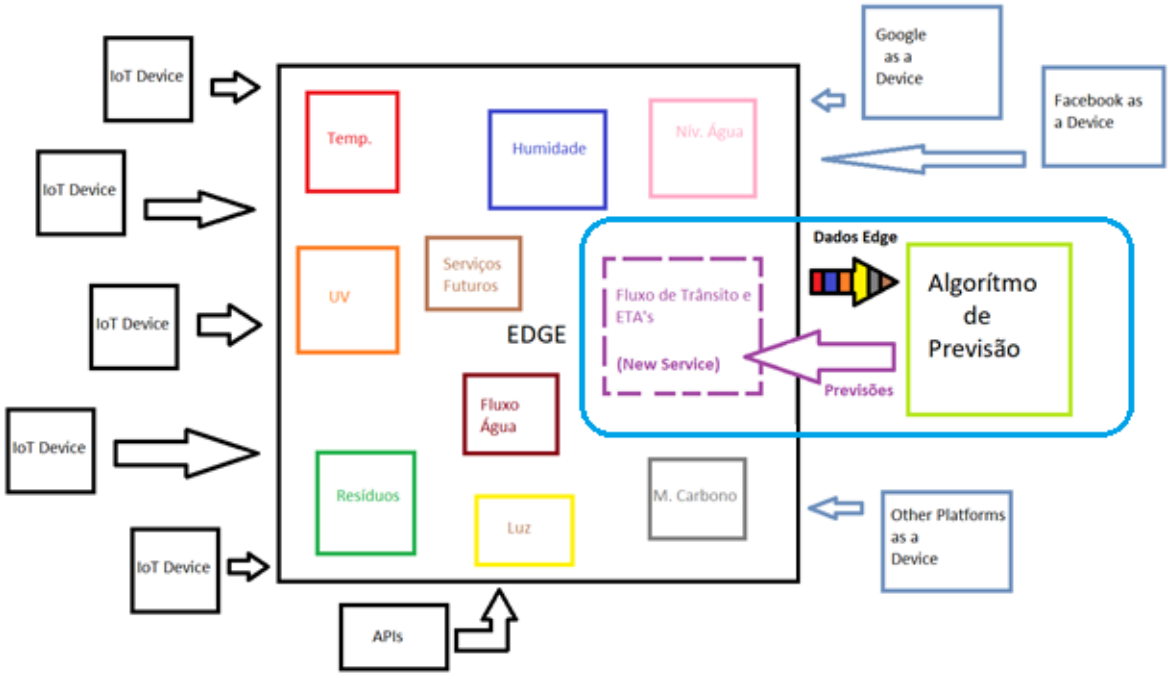


Figura 1: Arquitetura vaga do projeto (contornado a azul)

1.3 Abordagem Metodológica

1.3.1 Estratégia de Pesquisa Bibliográfica

Para se poder desenvolver um bom projeto de dissertação, é necessário a pesquisa dos conceitos necessários e relevantes para a escrita coerente da mesma, assim como saber o que já foi investigado dentro da área para que se possa construir sobre isso e encontrar uma vantagem diferenciadora para o projeto em relação aos anteriores. Para tal efeito é relevante a leitura de livros de referência na área e artigos de outros investigadores. É também bastante importante dar atenção aos artigos e outros meios de informação para o qual o orientador vai chamando a atenção, sabendo que à partida estes serão os mais informativos possíveis.

Na busca de informação serão utilizados motores de busca que irão facilitar bastante no processo de busca de informação, a escolha destes baseia-se maioritariamente na preferência pessoal e experiência prévia de utilização. Sendo assim o motor de busca principal foi o Scopus, cujo o interface é bastante intuitivo e contém milhares de artigos ou abertos ou que com autenticação universitária podem ser acedidos. Em alternativa ao Scopus existe o Web of Science, que é muito semelhante ao Scopus ao nível de conteúdo e beneficia igualmente da autenticação universitária. Os únicos inconvenientes que se pode

apontar a estes dois motores de pesquisa é que o seus sistema automático de querys pode criar filtragens contraditórias, o que pode ser resolvido com edição manual das mesmas abrindo assim novas possibilidades como o uso de aspas para procurar estritamente um termo; por exemplo ao pesquisar por Data Mining obtém-se resultados para Data e para Mining e tem-se de confiar no algoritmo para dar o resultado que se espera, no entanto se se pesquisar por "Data Mining" sabe-se que os resultados serão sobre este termo em particular e não as duas palavras que o compõem. Esta técnica será bastante utilizada nas pesquisas a realizar. Foi ainda utilizado o Google Scholar, que tem um bom algoritmo de pesquisa e aceita todas as técnicas de pesquisa que se pode utilizar no Google, no entanto tem como grande contra a dificuldade de encontrar documentos abertos, levando a que fosse o "último recurso", apenas a ser utilizado em caso de dificuldade a encontrar artigos sobre um tema nos outros dois. Para pesquisa complementar foi também utilizado o motor de pesquisa Google, que é útil para encontrar informação para poder-se iniciar num tema sob o qual não se tenha experiência ou para relembrar conceitos. Também é praticamente uma necessidade para encontrar informação em websites relevantes para pesquisa pois redireciona diretamente para a página que se procura no website a utilizar.

Para além das já mencionadas técnicas de querys como o uso de aspas, existem mais dois pontos fundamentais na pesquisa e seleção de artigos, sendo estes as palavras-chave e as filtragens. Começando pelas palavras-chave, a sua escolha é fundamental mas é também geralmente simples. Quando é necessário informação sobre um conceito como o Data Mining basta apenas incluir "Data Mining" na pesquisa, mas isto gera inúmeros resultados. A partir deste ponto existem duas opções, ou utiliza-se mais palavras-chave, ou inicia-se o processo de filtragem; no caso de se querer algo específico como por exemplo algoritmos deve-se pesquisar por "Data Mining" Algorithms, note-se que não se usa aspas na segunda palavra pois neste caso pode-se aceitar diferentes grafias como o singular ou então sinónimos que eventualmente a pesquisa possa considerar. No caso da filtragem, tendo em conta o grande número de documentos, existem escolhas que têm de ser feitas. Se se está a abordar um contexto histórico por exemplo pode ser uma boa ideia filtrar por artigos mais antigos, por exemplo até ao ano 2000, caso estas bases sejam importantes para compreender o porquê de algo ser o que é hoje em dia. A Tabela 2 sumariza diversos elementos que foram considerados para a seleção de artigos científicos.

No que toca à outra fonte de informação relevante, os websites, existem certos critérios aos quais têm que obedecer para serem considerado relevantes como fonte de informação para este projeto. Em primeiro lugar a fonte é relevante se esta for a página oficial de uma companhia reconhecida no mundo das tecnologias como é o caso da Google, Microsoft e Amazon por exemplo que, as próprias usam nos seus produtos e disponibilizam serviços relacionados com o tema em questão, sendo assim também têm

Ordem de Consideração	Objeto	Motivo
1	Linguagem	Se o artigo está numa linguagem como inglês ou português tem maiores chances de ser considerado. É preferível ler um artigo com um "mau inglês" do que ter que confiar num serviço online de tradução.
2	Título	O título, no que toca a artigos é extremamente poderoso. Não só dá para saber imediatamente o que aborda/investiga mas através da quantidade de palavras-chave no mesmo pode-se ter uma ideia do quão relevante vai ser para o trabalho em questão.
3	Resumo	Depois de passar do título tem-se o resumo, onde após a sua leitura a maioria dos artigos são ou marcados como leitura relevante ou automaticamente escolhidos.
4	Número de Citações	Assume-se de que quanto mais citações tem um artigo mais informativo, relevante e fiável é (apesar de se ter em consideração que artigos mais antigos tiveram mais tempo para serem citados que os recentes).
5	Ano	A não ser em contexto histórico, quanto mais recente for o artigo melhor. Em caso de escolha são preferíveis os artigos mais recentes que 2016 havendo ainda maior preferência pelos que estão marcados como sendo de 2018 e 2019.
6	Área do Conhecimento	Apesar de ser relevante saber para o que certas tecnologias estão a ser utilizadas em várias áreas, são dadas preferências a artigos na área do trabalho, especialmente à Ciência dos Computadores.

Tabela 2: Considerações na escolha de artigos

nos seus sítios alguma documentação e descrições sobre conceitos e temas importantes. O mesmo se aplica às tecnologias, se for procurar informação sobre Python o website oficial do Python é uma fonte de informação relevante assim como o Pypi, que com ele está diretamente relacionado. Outra categoria de websites relevantes para o projeto são aqueles mantidos por comunidades que são reconhecidos como uma fonte de informação fiável, como por exemplo o KDNuggets. O website online (e boletim informativo associado) www.kdnuggets.com fornece vários recursos online que cobrem tanto atividades comerciais como de pesquisa em Data Mining (Smyth, 2000).

1.3.2 Secção de Metodologias de Investigação e/ou Desenvolvimento

Para um bom desenvolvimento do projeto de dissertação é recomendado que se recorra a uma metodologia. Tendo em conta que esta tese vai envolver o desenvolvimento de um sistema com base no Data Mining, foi adotada a metodologia de desenvolvimento CRoss Industry Standard Process for Data Mining (CRISP-DM). O CRISP-DM foi desenvolvido no final dos anos 90 por um consórcio de empresas para facilitar o processo de Data Mining de ponta a ponta. Esta metodologia tem seis fases no seu ciclo de vida e estabeleceu o modelo padrão para a ciência de Data Mining (Grady, 2016).

De acordo com Chapman et al. (2000) as fases que compõem a metodologia CRISP-DM são:

-**Compreensão do Negócio** - Esta fase inicial foca-se na compreensão dos objetivos do projeto e dos requisitos da perspetiva do negócio, depois converte este conhecimento na definição de um problema de Data Mining e de um plano preliminar desenhado para alcançar os objetivos;

-**Compreensão dos Dados** - Esta fase começa com uma recolha dos dados iniciais e procede com atividade que te permitem ficar familiarizado com os dados, identificar problemas da qualidade dos dados, descobrir primeiras perceções dos dados, e/ou detetar subconjuntos interessantes para formar uma hipótese relacionada com a informação escondida;

-**Preparação dos Dados** - Esta fase cobre todas as atividades necessárias para construir o dataset final, que vai ser introduzido na ferramenta de modelação, a partir dos dados iniciais. As tarefas de preparação dos dados serão possivelmente repetidas múltiplas vezes.

-**Modelação** - Nesta fase, várias técnicas de modelação são selecionadas e aplicadas, e os seus parâmetros são calibrados para valores ótimos. Tipicamente, existem várias técnicas para o mesmo tipo de problema. Algumas destas técnicas requerem formas especiais de dados, obrigando a voltar à fase de **Preparação dos Dados**.

-**Avaliação** - Neste ponto do projeto, já existe um modelo que aparenta ser de alta qualidade de uma perspetiva de análise de dados. Antes de avançar para a **Implementação** do modelo. é importante avaliar e rever os passos que levarão à construção do mesmo, para ter a certeza de que o modelo consegue alcançar os objetivos do negócio. O objetivo chave é determinar se existe alguma parte do negócio que não tenha sido devidamente considerada.

-**Implementação** - A criação do modelo geralmente não é o final do projeto. Mesmo que o propósito do modelo seja aumentar o conhecimento dos dados, o conhecimento ganho vai necessitar de ser organizado e apresentado de uma maneira que o consumidor possa usa-lo. Dependendo dos requisitos, esta fase pode ser tão simples como gerar um relatório ou tão complexo como implementar um processo de Data Mining recorrente.

O fluxo destas fases está representado na Figura 2, que representa as diferentes fases do processo desta metodologia.

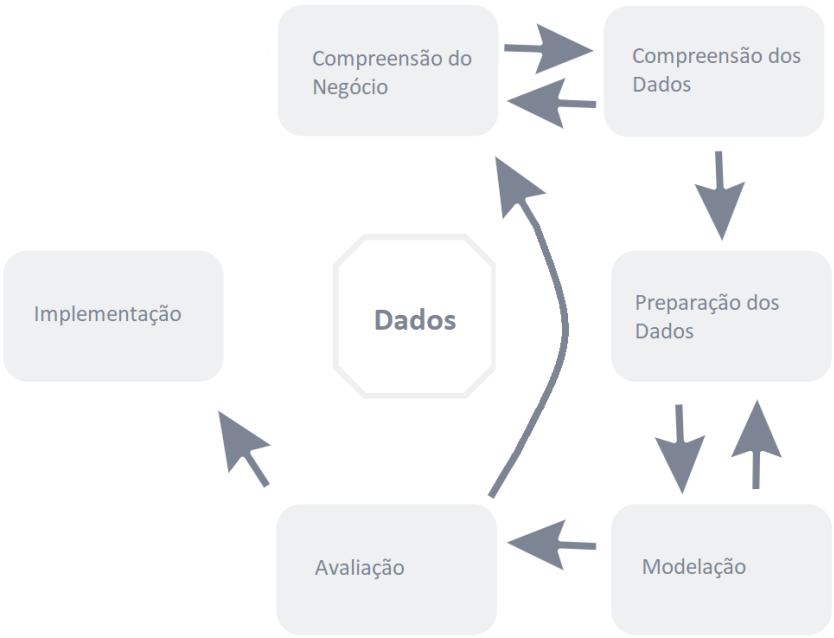


Figura 2: Ciclo da metodologia CRISP-DM

O CRISP-DM define um projeto como um processo cíclico, onde vária iterações podem ser usada para permitir um resultado final melhor ajustado para com os objetivos do negócio (Moro et al., 2011). A metodologia CRISP-DM é independente da ferramenta utilizada e as suas seis fases são flexíveis (Azevedo and Santos, 2008).

1.3.3 Organização do Documento

Este documento é constituído por cinco capítulos. O primeiro capítulo "Introdução" é composto por três secções, o "Enquadramento e Motivação", os "Objetivos e Resultados Esperados" e a "Abordagem Metodológica", onde são identificadas as estratégias de pesquisa, as metodologias a utilizar e a estrutura do documento. No segundo capítulo "Enquadramento Conceptual" existem sete secções, a "Introdução", "Sistemas de Apoio à Decisão" que explica o que são e qual a sua função, "Smart City e Smart Mobility" onde são explicados os conceitos de Smart City e de Smart Mobility, "Inteligência Artificial e Machine Learning" onde são descritos estes conceitos e a sua relação, "Data Mining" onde é explicado o Data Mining, os seus tipos e algoritmos, "Internet das Coisas e Big Data" que esclarece estes conceitos e como se relacionam entre si e a "Previsão e Simulação no Contexto de Trânsito" onde olha-se para trabalho desenvolvido na área. No terceiro capítulo há seis secções que descrevem brevemente a tecnologia a utilizar e o porquê, essas secções são "SQL e MongoDB", "R", "Python", "Algoritmos", "Ferramentas de Visualização" e "Dados". O quarto capítulo "Desenvolvimento" contém oito secções que descrevem o processo completo de desenvolvimento, essas secções são "Compreensão do Negócio", "Compreensão dos Dados", "Preparação dos Dados", "Modelação", "Avaliação", "Discussão de Resultados", "Componente Servidor" e "Componente de Visualização". Por último no quinto capítulo "Conclusão" são feitas observações sobre o decorrer do trabalho, as limitações incorridas e como dar continuidade ao mesmo, as secções que o constituem são "Síntese do Trabalho Efetuado", "Limitações" e "Perspetivas para Trabalho Futuro".

2

Enquadramento Conceptual

2.1 Introdução

Neste capítulo serão abordados diferentes conceitos relacionados com a área de estudo em questão, com um foco principal nas partes relativas à previsão de tráfego, Data Mining e mobilidade. Serão apresentados inicialmente diversos conceitos, a sua descrição de acordo com outros autores e como diferentes conceitos se comparam, para que haja uma compreensão dos mesmos, relativamente ao projeto em questão. Haverá uma revisão de leitura onde será identificado o que já foi feito na área por outros investigadores. Também se pode encontrar neste capítulo uma secção dedicada a identificar trabalhos desenvolvidos relacionados com este projeto, assim como o que já existe no mercado que propõe fazer algo semelhante ao que está a ser desenvolvido, pois não adiante desenvolver algo que já existe, será necessário pelo menos um ponto que diferencie este projeto para melhor; é necessário estudar o estado da arte tanto nas áreas de otimização de rotas como na previsão de tráfego.

2.2 Sistemas de Apoio à Decisão

A tomada de decisões é considerada uma das atividades mais críticas feita nas organizações. Para ajudar este processo complexo para os indivíduos, uma variedade de sistemas de informação chamados de Sistemas de Apoio à Decisão (DSS) foram desenvolvidos, computadores baseados em ferramentas usadas para ajudar na tomada de decisões complexas e na resolução de problemas (Shirgaonkar et al., 2010). Os DSS, juntamente com ferramentas de Data Mining são praticamente essenciais para lidar com grandes quantidades de dados (Rubin et al., 2001). Um DSS pode ser definido como um sistema interativo, baseado em computadores, que ajuda tomadores de decisão utilizar dados e modelos para resolver problemas não estruturados, apesar de que qualquer sistema que apoie uma decisão, de qualquer

maneira, pode ser considerado um DSS (Zaraté, 1991). A utilidade dos DSS é particularmente alta em áreas onde os problemas são mais abertos e mal estruturados. Estes sistemas estão desenhados para ajudar o gestor na modelação do processo de decisão e fornecem flexibilidade e interfaces intuitivas que ajudam em análises mais difíceis com maior facilidade. Um dos benefícios chave poderá ser o permitir que o gestor entenda melhor o negócio e melhore a sua intuição (Davis and Sundaram, 1995). Um DSS consiste geralmente numa grande quantidade de variados métodos e modelos, incluindo programação matemática, análise estatística, teoria das decisões estatísticas e tomada de decisões sob incerteza, abordagens heurísticas, entre outras (Pashkevich et al., 2019).

Se um DSS for implementado com sucesso e o seu potencial utilizado, não só vai beneficiar os seres humanos como também vai ser extremamente importante na tomada de decisões críticas (Shirgaonkar et al., 2010). O DSS a desenvolver neste projeto de dissertação enquadra-se melhor no tipo de DSS inteligentes baseados na descoberta de conhecimento (IDSSKD), definidos pelo seu uso de várias técnicas como Data Mining e Web Mining para extrair conhecimento útil para a tomada de decisão (Shirgaonkar et al., 2010). Um DSS pode avaliar os efeitos de uma decisão e gerar apenas aquelas que satisfazem os requisitos do utilizador. Um tomador de decisão equipado com um DSS pode simular decisões e ver o efeito das mesmas (van Hee et al., 1991). DSS são utilizados em áreas tais como avaliação de créditos (IÇ and Yurdakul, 2010), triagem de doenças (Chu et al., 2017), publicidade e promoções (Davis and Sundaram, 1995) e processos de entregas em transportes de estrada (Pashkevich et al., 2019).

2.3 Smart City e Smart Mobility

Os conceitos de Smart City e Smart Mobility são recentes e estão estritamente ligados entre si. Nesta secção será explicada a sua relação e dependência assim como definidos os mesmos no âmbito deste projeto.

2.3.1 Smart City

Várias cidades em todo o mundo estão num estado constante de fluxo e exibem dinâmicas complexas (Caragliu et al., 2011). Esta complexidade e constante fluxo tornam-nas num meio difícil de gerir. Surgiu assim a necessidade para soluções de controlo e gestão, a solução para este problema veio com o nome Smart City. O rótulo Smart City deveria apontar para soluções inteligentes que permitam as cidades modernas prosperar (Caragliu et al., 2011), mas, tal como as cidades que devem ser melhoradas, o

conceito de Smart City é complexo e também ele necessita de ser melhorado. Smart City é um conceito vago, não bem definido, contudo, todas as definições concordam no facto de uma Smart City ser um espaço urbano que tende a melhorar a vida diária dos cidadãos (Negre et al., 2015). Apesar de não haver uma definição formal e geralmente aceite de Smart City o objetivo final é fazer melhor uso de recursos públicos, melhorar a qualidade dos serviços oferecidos aos cidadãos, enquanto se reduz os custos operacionais das administrações públicas (Zanella et al., 2014).

2.3.2 Smart Mobility

Um sistema urbano da IoT, pode trazer um número de benefícios na gestão e otimização dos serviços públicos tradicionais, como **transporte** e estacionamento, iluminação, vigilância, entre outros (Zanella et al., 2014). A parte que traz inteligência à secção de transportes é aquilo a que se chama de Smart Mobility. Um projeto do Centro de Ciência Regional na Universidade de Tecnologia de Viena, de acordo com o artigo "Smart cities in Europe", Caragliu et al. (2011) identificam Smart Mobility como um dos seis eixos principais das Smart Cities. Segundo o autor de "Cagliari and smart urban mobility: Analysis and comparison", é afirmado por Chun and Lee (2015) que Smart Mobility é um conceito de um serviço de tráfego futuro mais inteligente e compreensivo em combinação com tecnologia mais inteligente (Garau et al., 2016). O que foi dado a entender por estes autores é que Smart Mobility é uma parte do sistema global de uma Smart City, sendo que é dentro do Smart Mobility que se enquadram as melhorias relacionadas com transportes.

2.4 Inteligência Artificial (IA) e Machine Learning (ML)

Muito recentemente têm havido um grande interesse nas áreas da Inteligência Artificial (IA) e de Machine Learning (ML). De acordo com o Google Trends (Figura 3) nos últimos 5 anos têm aumentado o interesse dos utilizadores nestas nestas áreas.

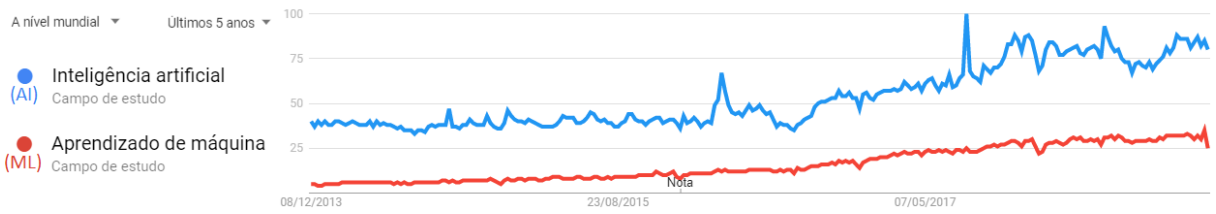


Figura 3: Google Trends: ML vs IA, adaptado de (Trends, 2018)

2.4.1 Inteligência Artificial (IA)

A IA não é um conceito novo (Minsky, 1961). Nesse artigo, chama-se a estes computadores de "cérebros mecânicos", sendo admitido o potencial destas máquinas para jogar certos jogos, e mesmo derrotar os seus criadores, fenómeno que foi recentemente comprovado com o sistema AlphaGO, o primeiro programa de Go a atingir uma performance sobre humana e vencer o campeão europeu do jogo Fan Hui em outubro de 2015, recorrendo a ML, nomeadamente métodos de Reinforcement Learning (Silver et al., 2017). Ainda antes do artigo de Minsky (1961), em 1950 Alan Turing, matemático, propôs o que agora é chamado de Teste de Turing para verificar se uma máquina é realmente inteligente. Se uma máquina conseguir enganar um ser humano, ao fazer passar-se por um ser humano, da mesma maneira que um ser humano consegue enganar outro, a máquina passaria o teste (Alfonseca, 2014). Ou seja, se uma máquina falar com um ser humano e este não conseguir perceber que está a falar com uma máquina, a máquina pode ser considerada inteligente. Pode-se ver uma representação deste conceito adaptada do artigo "Can machines think? A report on Turing test experiments at the Royal Society" (Warwick and Shah, 2016) na Figura 4, onde um ser humano avalia duas entidades onde uma é uma máquina e a outra é um humano e este tem que decidir qual é qual só através da comunicação com cada uma.

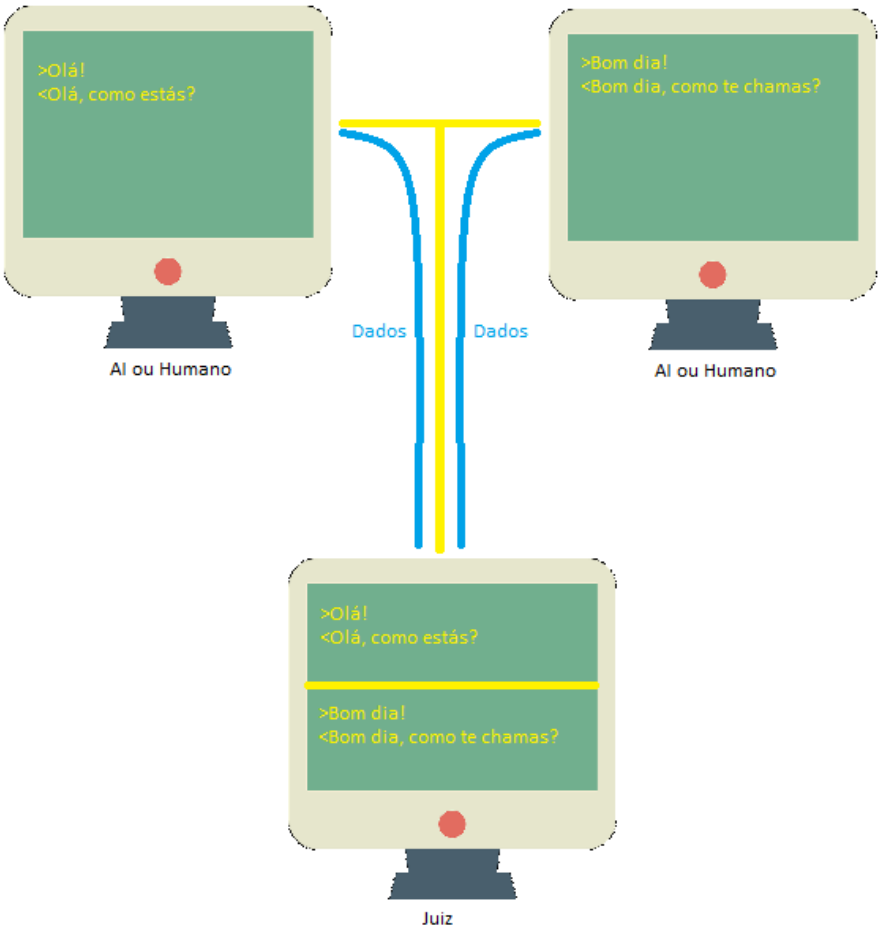


Figura 4: Teste de Turing envolvendo um juiz a interrogar duas entidades: uma máquina e um humano, adaptado de (Warwick and Shah, 2016)

Mais recentemente, em 2012, é utilizada no artigo ""Closing" Some Doors For the Open Semantic Web" (Pan, 2012), a definição de John McCarthy de IA, sendo esta a ciência e engenharia para fazer máquinas inteligentes, especialmente sistemas de computadores, ou seja, McCarthy define IA não como o conceito de máquinas inteligentes mas como a ciência de as fazer. Esta ideia é reforçada por um artigo no website KDNuggets, que afirma que a IA lida com a área de desenvolver sistemas de computação capazes de realizar tarefas que seres humanos são muitos bons a fazer (Adam, 2018).

2.4.2 Machine Learning (ML)

No mesmo artigo do website referenciado anteriormente, há uma explicação do que é ML que, segundo o autor é definido como o campo de IA que aplica métodos estatísticos para permitir que sistemas de computador aprendam dos dados para um objetivo final. De acordo com a mesma fonte o termo ML foi introduzido por Arthur Samuel em 1959. ML é uma das áreas chaves associadas à IA. O ML tenta mel-

horar as capacidades de aprendizagem de sistemas inteligentes (Sankar et al., 2016). ML é uma técnica para reconhecer padrões (Erickson et al., 2017). ML é um meio para derivar IA por meio de descobrir padrões em dados existentes (Caliskan et al., 2017).

Hoje em dia utiliza-se o ML em vários aspetos da sociedade e da evolução tecnológica, por exemplo, no campo da medicina uma das utilizações é na deteção de carcinoma nasofaringeal em que usam redes neuronais para auxiliar na identificação deste tumor como é descrito no artigo de 2018 (Mohammed et al., 2018). No campo das finanças utilizam ML para previsão do mercado financeiro como descrito no artigo (Fischer and Krauss, 2018). Também é utilizado ML no campo dos carros autónomos, no caso mencionado no artigo (Brunetti et al., 2018) na deteção de peões. Também relacionado com a melhoria de deteção de objetos, a Google tem utilizado ML para criar conjuntos de dados para treinar máquinas através do ReCAPTCHA como eles referem no seu site "reCAPTCHA faz um uso positivo deste esforço humano canalizando o tempo despendido a resolver os CAPTCHAs em anotar as imagens e criar conjuntos de dados de ML. Isto ajuda a melhorar mapas e resolver problemas de IA mais difíceis." (reCAPTCHA, 2018). ML é utilizado por inúmeras aplicações do mundo real, seja pelo Netflix para sugerir filmes com base nos que já foram vistos ou pela Amazon para recomendar livros baseados nos que já foram comprados anteriormente (Le, 2016).

Existem vários algoritmos de ML. Segundo "Toward interactive search in remote sensing imagery", a maioria dos algoritmos de ML tenta substituir os utilizadores (Porter et al., 2010). Existem 3 tipos de ML, o tipo Supervised que faz previsão baseado em exemplos, como preços históricos para a previsão de preços futuros, Unsupervised em que os dados não têm rótulos associados e o algoritmo tem que dar estrutura a esses dados e Reinforcement Learning em que o algoritmo tem que escolher a ação que acha apropriada para um certo caso em que se acertar receberá um sinal de recompensa baseado no quão boa foi a decisão (Ericlicoding, 2018). Como exemplo de algoritmos Supervised existem as árvores de decisão, regras de classificação e a regressão linear; como exemplos de algoritmos Unsupervised têm-se os algoritmos de Clustering (que juntam objetos por similaridade) e decomposição em valores singulares (baseado em decomposição de matrizes) (Le, 2016). Os modelos de Reinforcment Learning são mais utilizados na robótica e aplicações da IoT (Ericlicoding, 2018).

2.5 Data Mining (DM)

A habilidade de extrair conhecimento útil escondido em dados e atuar sobre esse conhecimento é cada vez mais importante num mundo competitivo. O processo de aplicar uma metodologia baseada

em computadores para descobrir esse conhecimento é chamada de Data Mining (DM), um dos campos de estudo com um crescimento muito rápido na indústria da computação. DM consiste na procura por informações novas, valiosas e não triviais em grandes volumes de dados (Kantardzic, 2011).

O termo "Data Mining" teve uma história variada. Nos anos 60, enquanto computadores digitais começavam a ser aplicados a problemas de análise de dados, foi descoberto que se alguém pesquisasse tempo suficiente (usando um computador) poder-se-ia sempre encontrar um modelo relativamente complexo que se adequasse relativamente bem a um conjunto de dados. Então termos como DM e "Data Dredging" foram utilizados para descrever essas atividades. Nos inícios dos anos 90, o termo DM foi independentemente adotado por cientistas de computadores para descrever algoritmos e métodos orientados às base de dados que pesquisavam por estruturas inesperadas e padrões nos dados. Geralmente estes conjuntos de dados eram massivos (Smyth, 2000).

DM tem várias aplicações como por exemplo: deteção de spam, diagnósticos médicos, anúncios relevantes, acessão de risco, astronomia, geologia, biologia, análise de sentimentos, identificação de fraude fiscal e classificação de imagens (Tkachenko et al., 2019). DM, como um campo multidisciplinar, é utilizado por várias áreas, tais como estatística, ML, sistemas de base de dados, redes neuronais, conjuntos difíceis, entre outros (Sun, 2014). Um foco recente da atividade do DM envolve a aplicação dos conceitos de DM a coleções online de documentos de texto e objetos multimédia tais como imagens, vídeos e áudio (Smyth, 2000).

De acordo com o autor de "Privacy-Preserving Data Mining", a tarefa principal do DM é o desenvolvimento de modelos sobre dados agregados (Agrawal and Srikant, 2000). DM visa compreender os hábitos do consumidor prever a procura de um produto, construção e gestão de marca, seguir a resposta dos consumidores e a performance dos produtos no mercado, entre outros e depois transformar esses dados em informação e depois transformar essa informação em conhecimento. Pode ajudar analistas a descobrir tendências anormais ou que poderiam passar despercebidas de outra forma (Sun, 2014).

As principais técnicas de DM são de classificação, que consiste em associar dados a classes, regressão que associa itens a previsões de valor real, clustering que agrega os dados dentro de um número finito de categorias, resumo que encontra descrições para conjuntos de dados, modelos de dependências e deteção de mudança e desvios (Kantardzic, 2011). Classificação é uma das tarefas de DM mais utilizadas (Tkachenko et al., 2019).

Existem diversos algoritmos de classificação. Classificadores de árvores de decisão são relativamente rápidos, produzem modelos compreensivos, e obtêm níveis de precisão similares ou por vezes melhores que outros modelos de classificação (Agrawal and Srikant, 2000). Regressão Logística, uma das variantes

de classificação, é uma das mais simples mas também das mais eficazes e é capaz de resolver tarefas particionadas linearmente. Support Vector Machine (SVM) é outro sistema de classificação linear, em que, baseado numa teoria estatística de aprendizagem, pode construir um hiperplano num espaço transformado por uma transformação de kernel, de modo que as classes sejam linearmente separáveis nesse espaço transformado. Random Forest é um dos mais populares métodos de classificação, tem uma boa precisão, das melhores entre os métodos de classificação, sendo constituído por um agregado (ensemble) de árvores de decisão. Todos estes métodos oferecem proteção contra sobre-ajuste (Tkachenko et al., 2019). O autor do artigo referenciado comparou estes métodos e obteve os resultados observados na Figura 5.

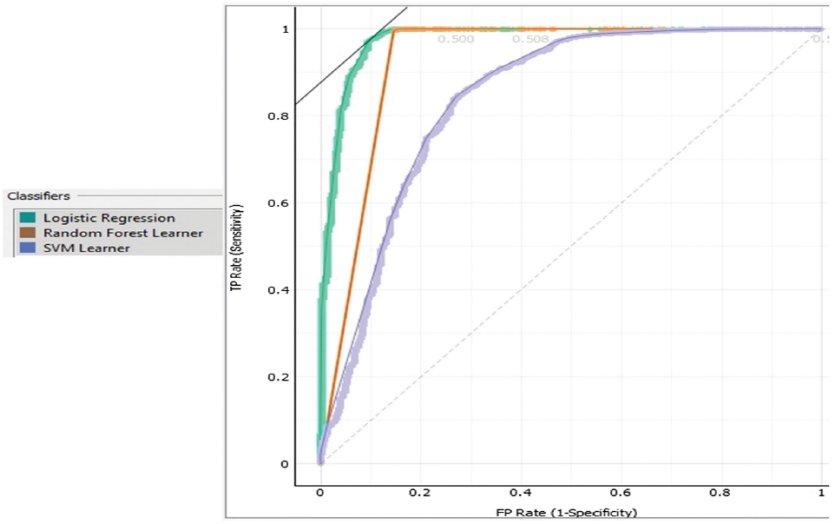


Figura 5: Exemplo de uma comparação de três classificadores, via curvas ROC, retirado de (Tkachenko et al., 2019).

O autor conclui que o algoritmo Random Forest mostra os melhores resultados em geral mas, como a amostra não esta equilibrada, a precisão para reconhecer classes menores não é alta o suficiente (Tkachenko et al., 2019). Resultados como estes serão tidos em conta na escolha preliminar dos algoritmos a testar.

Os melhores resultados são obtidos quando se equilibra o conhecimento de humanos peritos em descrever problemas e objetivos com as capacidades de pesquisa dos computadores (Kantardzic, 2011). O processo de DM geralmente envolve os passos: definir o problema e formular hipóteses, recolher dados, tratar os dados, estimar o modelo e interpretar o modelo e retirar conclusões, sendo que a maioria do esforço do processo consiste nos três primeiros passos (Kantardzic, 2011).

2.6 Internet das Coisas (IoT) e Big Data

2.6.1 Internet das Coisas (IoT)

A IoT é um paradigma de comunicações recente que visa um futuro próximo onde objetos do dia a dia serão estarão equipados com micro-controladores, transmissores para comunicações digitais e conjuntos de protocolos apropriados que possibilitarão estes dispositivos a comunicarem uns com os outros e com os utilizadores, tornando-se numa parte integral da Internet (Zanella et al., 2014).

A IoT é uma nova tecnologia que está a crescer rapidamente no campo das telecomunicações (Stergiou et al., 2018). Hoje em dia a tecnologia da IoT está a tornar-se cada vez mais e mais importante, o que traz uma enorme conveniência para a vida das pessoas e para o desenvolvimento da cidade. IoT tem um enorme conjunto de aplicações como por exemplo casas inteligentes, transportes inteligentes, redes inteligentes e sistemas de saúde inteligentes (Shen et al., 2018).

O termo IoT foi inicialmente utilizado por Kevin Ashton em 1999 no contexto de gestão de cadeias de fornecimento; no entanto na última década, esta definição passou a ser mais inclusiva e a cobrir mais aplicações como saúde, utilidades e transportes. O principal objetivo de fazer estes dispositivos que criam a IoT sentirem informação é ajudar a intervenção humana. A IoT pode ser vista por três paradigmas: orientada à Internet (middleware), orientada aos objetos (sensores) e orientada à semântica (conhecimento) (Gubbi et al., 2013). A IoT deverá conseguir incorporar de maneira transparente e discreta um grande número de diferentes sistemas heterogéneos, enquanto fornece acesso aberto a subconjuntos de dados para o desenvolvimento de múltiplos serviços digitais (Zanella et al., 2014). O principal objetivo da interação e da cooperação entre coisas e objetos é cumprir o objetivo que lhes foi dado como uma entidade combinada (Stergiou et al., 2018).

A IoT é uma rede de objetos físicos, dispositivos, veículos, edifícios e outros itens que estão embutidos com eletrónica, software, sensores e conexões pela rede, permitindo a estes objetos recolher e trocar dados. A IoT é composta por três partes principais: os objetos, a rede que os conecta e os sistemas de computadores que transmitem os dados de e para os objetos (Stergiou et al., 2018). De acordo com a curva de Gartner sobre tecnologia emergentes de julho de 2018 (Panetta, 2018), as plataformas IoT estão ainda no pico das expectativas mas espera-se que o seu ponto de estabilização seja atingido nos próximos 5 a 10 anos como se pode observar na Figura 6.

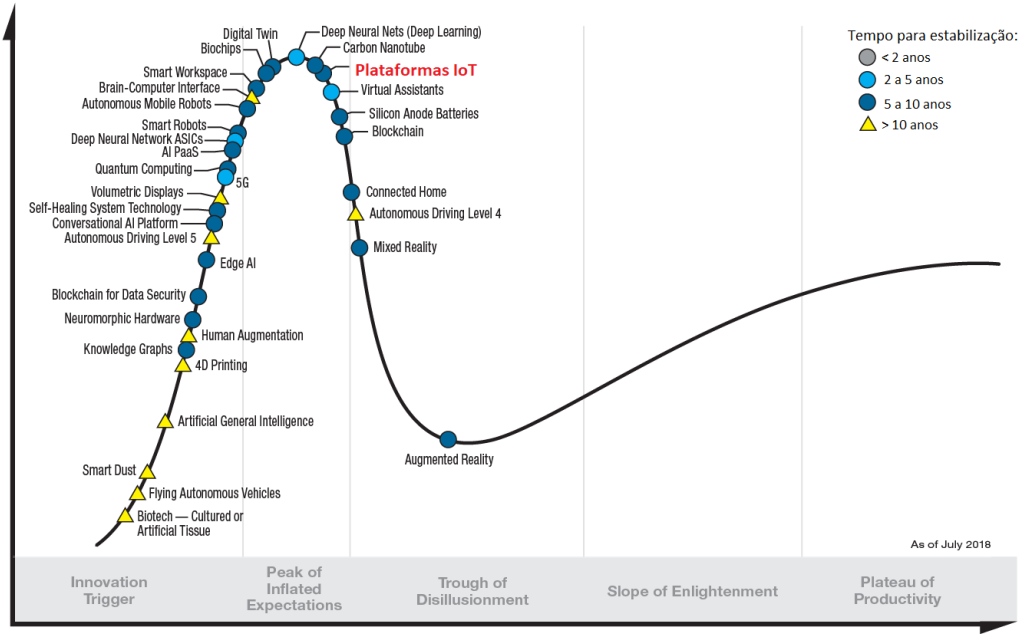


Figura 6: Hype Cycle para tecnologias emergentes em 2018, adaptado de (Panetta, 2018).

No caso específico dos transportes, IoT pode proporcionar reduções a nível de trânsito nas estradas, reduzir o consumo de combustível, estabelecer prioridades nos programas de reparação dos veículos e salvar vidas (Stergiou et al., 2018).

2.6.2 Big Data

Nos últimos 20 anos, o número de dados tem aumentado numa grande escala em vários campos. Debaixo deste aumento explosivo de dados globais, o termo Big Data é especialmente usado para descrever conjuntos de dados enormes. Comparados com tradicionais conjuntos de dados, Big Data geralmente inclui massas de dados não estruturados que necessitam de análise em tempo real. Em geral Big Data deve significar os conjuntos de dados que não podem ser compreendidos, adquiridos, geridos e processados em tempo tolerável por meios tradicionais (Chen et al., 2014).

Com o crescimento do uso de computadores, existe uma maior quantidade de dados a serem gerados por estes sistemas. Recentes avanços na computação, comunicações e armazenamento digital, juntamente com as altas taxas de transferência de tecnologias de aquisição de dados, possibilitam o armazenamento de enormes volumes de dados. Na Figura 7 pode-se ver como exemplo o número de Internet Hosts (anfitriões), que cresceu dramaticamente nos últimos anos, números estes que estão di-

retamente relacionados proporcionalmente à quantidade de dados armazenada na Internet (Kantardzic, 2011).

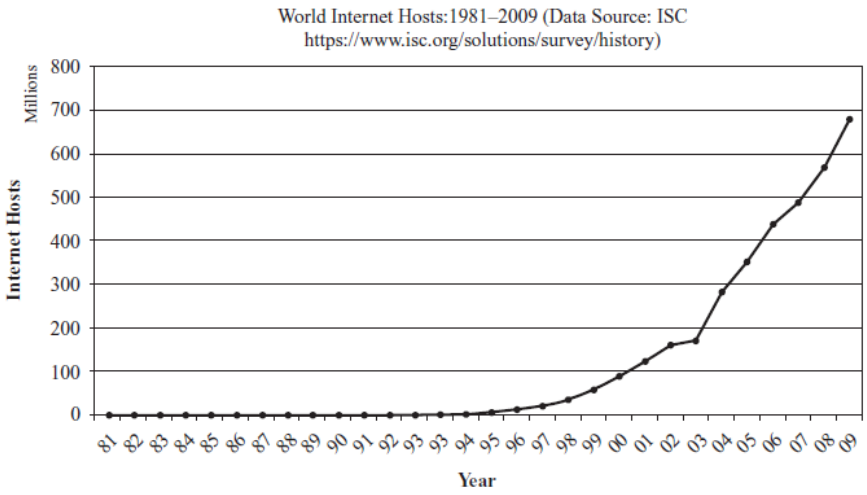


Figura 7: Crescimento de anfitriões na Internet, retirado de (Kantardzic, 2011).

Big Data é um termo recente para as tecnologias de informação, usado para se referir a um aumento no volume de dados que são difíceis de armazenar, processar e analisar por meios tradicionais. É um conjunto de técnicas e tecnologias que requer novas formas de integrar para descobrir valores escondidos em conjuntos de dados complexos, diversos e massivos em escala (Hashem et al., 2015). Big Data diz respeito aos volumosos, complexos e crescentes conjuntos de dados com múltiplas fontes. Com o rápido desenvolvimento das redes, armazenamento de dados e capacidade de recolher dados, Big Data está a expandir-se rapidamente nos campos da ciência e engenharia. A capacidade de recolher dados cresceu tanto que está para além das capacidades dos softwares comuns capturar, gerir e processar estes dados em tempo útil. Wu et al. (2014) apresentam um teorema HACE que consiste em começar com dados de fontes heterogéneas e autónomas e procura explorar relações complexas e evolutivas nos dados; estas características criam um desafio para a descoberta de conhecimento útil em Big Data.

De acordo com o Gartner Group, Big Data é alto volume, alta velocidade e/ou alta variedade de informação, exigindo medidas inovadoras e económicas de processamento de informação que possibilitem uma melhor visão, tomada de decisão e automação de processos (Gartner, 2016). Existem três dimensões que são desafios à gestão de dados, conhecidos como os três V's: Volume, Variedade (Heterogeneidade) e Velocidade (a que os dados são gerados) (Gandomi and Haider, 2015). Entre as Smart Cities construídas sobre IoT, Big Data pode surgir da indústria, agricultura, trânsito, transportes, saúde, entre outros. O Big Data gerado pela IoT tem características diferentes, onde incluem heterogeneidade, variedade, falta de estrutura, ruído e redundância. Apesar de atualmente os dados da IoT não serem uma parte dominante do

Big Data, em 2030, a quantidade de sensores vai ser tão alta que estes serão os dados mais importantes do Big Data (Chen et al., 2014).

Algoritmos de DM típicos, requerem que todos os dados sejam carregados na memória, no entanto, isso está a tornar-se numa barreira ao Big Data pois mover tantos dados entre diferentes localizações tem custos mesmo num computador com muita memória. Para um sistema de base de dados inteligente lidar com Big Data, é essencial escalar para grandes volumes de dados e fornecer tratamentos para as características indicada pelo teorema HACE. Primeiro, dados escassos, heterogéneos, incertos, incompletos e de fontes múltiplas devem ser processados com técnicas de fusão de dados. Segundo, dados dinâmicos e complexos são minerados após o pré processamento. Terceiro, o conhecimento global obtido por aprendizagem local e fusão de modelos é testado e informação relevante é dada como feedback para a fase de pré processamento. Depois os modelos e parâmetros são ajustados de acordo com o feedback. Em todo o processo, troca de informação não é só uma promessa de melhor desenvolvimento em cada fase, mas também um propósito do processamento de Big Data (Wu et al., 2014).

2.7 Previsão e Simulação no Contexto de Trânsito

A previsão precisa e robusta dos parâmetros de velocidade, tempo de viagem, fluxo de trânsito, entre outros são um problema crítico, de tal forma que qualquer melhoria poderia trazer mais eficiência à gestão de transportes, podendo levar a melhorias de transportes de pessoas e mercadorias, encontrando as rotas mais rápidas e evitando congestão. O que torna o trânsito tão difícil de prever é que este exibe comportamentos esperados e inesperados, dependendo das circunstâncias. Como por exemplo, estado do tempo, eventos e comportamentos dos condutores, o que afeta a precisão dos modelos de previsão (Bezuglov and Comert, 2016). Estas previsões são essenciais para aplicações de sistemas de transportes inteligentes (ITS) (Fusco et al., 2016). Os avanços nas tecnologias de comunicação sem fios e em Cloud Computing fazem com que o paradigma da Internet dos Veículos passe de um problema de otimização de rotas regular para uma previsão em tempo real (Wan et al., 2016).

No artigo "Short-term freeway traffic parameter prediction: Application of grey system theory models" (Bezuglov and Comert, 2016), propõem-se a solucionar este problema utilizando a teoria Grey Systems, que segundo os autores se enquadra muito bem no problema. Esta teoria foi utilizada anteriormente para prever vários assuntos no mercado. Muitos investigadores propõem que esta teoria funciona melhor em sistemas híbridos. Eles utilizaram dois conjuntos de dados diferentes de localizações distintas e organizaram os dados em conjuntos de tempo. Para comparar os diferentes modelos que utilizaram,

foram usados os modelos LSTAR, SETAR, NNETS e AAR. Para testar o modelo utilizaram séries de tempo numa janela deslizante, prevendo resultados e comparando-os aos reais. No fim concluíram que os modelos Grey tinham melhores performance e precisão em geral sendo o melhor o EFGVM (Grey Verhulst com correção de Fourier), e acreditam que existe espaço para melhoria com estes modelos no futuro (Bezuglov and Comert, 2016).

Uma das primeiras investigações feitas na previsão de tráfico utilizando ML foi de Dougherty and Cobbett (1997) na Holanda, com o objetivo de prever o fluxo de trânsito, a velocidade e ocupação. O autor descreve o problema como multi-dimensional envolvendo múltiplos pontos no espaço-tempo. São apontados os problemas de ruído inerentes a este tipo de dados. Este adiciona que as capacidades preditivas a curto prazo são superiores quando são fornecidos dados do tempo imediatamente anterior à previsão e que existe uma custo-benefício associado à qualidade do modelo e o tempo e esforço requerido para o treinar e usar. Em especial são apontados dois problemas com a utilização da variável velocidade sendo estes as distorções cridas por veículos muito rápidos ou muito lentos e a possibilidade de falta de dados num período de tempo.

Outros investigadores para a previsão do fluxo de trânsito propuseram-se a utilizar LSSVM (Least squares support vector machine) que é um novo tipo de SVM que pode ser usado para aproximar sistemas não lineares com uma maior precisão, o que é uma ferramenta poderosa para modelar esse tipo de sistemas. De acordo com eles o fluxo de trânsito no momento está estritamente relacionado com o fluxo de trânsito de minutos antes nas ruas, logo o fluxo de trânsito anterior pode ser utilizado para prever o atual e por consequência o futuro. Dividiram os dados em dados de treino e dados de teste e testaram os modelos. Concluíram que o modelo LSSVM-FOA (abordagem híbrida do LSSVM com o algoritmo de otimização mosca da fruta) tem vantagens óbvias na previsão do fluxo de trânsito apesar de demorar mais que o LSSVM por si só (Cong et al., 2016).

Em 2016 um grupo de investigadores tentou prever as velocidades recorrendo ao Big Data gerado por carros flutuantes. Esta previsão é a curto prazo. Os dados foram obtidos a partir de Sistema de Posicionamento Global (GPS), o que causa um problema nas zonas não viajadas pelos veículos equipados. Os autores afirmam que o que tem sido proposto na resolução destes problemas são redes neuronais e redes Bayesianas, que procuram estabelecer correlações existentes. Poderiam ser usados dois tipos de previsões a curto prazo, explicitas que são baseadas em modelos matemáticos que representam a interação entre objetos físicos e implícitas que derivam relações dinâmicas dos dados observados, sendo que os explícitos são superiores em termos de capacidades de interpretação. Foram testadas as performances com diferentes condições de congestionamento. Foi utilizado o modelo SARIMA, uma variação

do modelo ARIMA, que é muito aplicado na previsão de fluxo de trânsito. Foram utilizadas redes de Bayesian que são modelos gráficos probabilísticos, onde cada nodo corresponde a uma variável e estes nodos estão conectados por ligações. Também foram utilizadas redes neuronais artificiais em particular uma Feed-Forward neural Network, que são melhores que as recursivas para esta previsão. Os dados recolhidos por GPS foram de cerca de 100.000 veículos e registados valores a cada 2 minutos, que posteriormente foram organizados em intervalos de cinco minutos. Os modelos foram validados ao comparar velocidades com 2 modelos distintos com a média correspondente aos dias úteis da última semana do mês. Os resultados obtidos sugeriram que poderiam utilizar um modelo combinado supervisionado de tomada de decisões. Sendo assim o melhor seria uma junção de redes Bayesianas e o modelo SARMA sob um sistema supervisionado (Fusco et al., 2016).

No artigo "Real-time traffic state estimation in urban corridors from heterogeneous data" (Nantes et al., 2016) é sugerido que um modelo de estimação do fluxo de trânsito pode resolver o problema de estimar quantidades de dados de sensores. É dito que filtros Kalman e métodos de Montecarlo são bastante utilizados para estimar o fluxo de trânsito. É uma solução mais económica utilizar os dados destes sensores do que instalar hardware dedicado para o efeito nas estradas. Adiantam que uma das maiores falhas de utilizar uma abordagem aos dados é que os modelos tendem a falhar fora do contexto onde foram treinados. No artigo eles propõem uma nova versão de filtros Kalman (EKF) que possa incorporar medida heterogéneas quando disponíveis. Eles definem uma estrada como um conjunto de ligações entre nodos sendo os nodos as interseções. No final tiveram que recorrer a simular dados de Bluetooth e GPS de acordo com dados existentes devido a dados insuficientes. Ainda assim concluíram que não há benefícios do uso de dados Bluetooth em relação aos GPS e mesmo quando estes são fundidos não há vantagens (Nantes et al., 2016).

No artigo de 2017 "Deep learning for short-term traffic flow prediction", os autores tentam modelar os efeitos espaço-temporais que ocorrem devido a eventos como zonas de construção, eventos e acidentes. Através de Deep Learning é possível prever congestionamento a longo prazo e é possível incorporar fontes de dados como a meteorologia.

Com veículos equipados com sensores, Smartphones e acesso virtual aos mesmos e a convergência de comunicações móveis e tecnologia de inteligência terminal é cada vez mais possível aliviar a congestão nas estradas através de tecnologia de Mobile Crowd Sensing (MCS), ou seja recolher dados de um conjuntos de sensores da mesma área, em múltiplas áreas. Foram propostos por outros trabalhos a utilização de VANET's que consiste na comunicação dos sensores com pontos de controlo junto às estradas, um problema é quando os veículos não estão no alcance destes pontos que os investigadores dizem poder

ser solucionado através de comunicar com o vizinho ou seja com outros sensores próximos que estejam no alcance deste ponto. É também proposto que o sistema possa preencher as lacunas com tempos codificados nos mapas, pois segundo eles é aceitável fazê-lo porque o sistema vai atualizando até chegar a um ponto onde os caminhos tenham informação mais precisa. Os autores apontam que os dois maiores desafios na previsão de trânsito são como agregar os dados e prever o futuro usando a VANET e como melhorar a eficiência com DM dos dados. Os investigadores fizeram um teste real e concluíram que o algoritmo baseado em MCS utilizou dados em tempo real para evitar congestão. E apesar de lhes fornecer uma distância maior conseguiu que chegassem do ponto A a B em 25 minutos em vez dos 45 minutos habituais (Wan et al., 2016). Noutro tipo de investigação foram adaptados redes neuronais para seguir carros para simular oscilações no trânsito. Foi concluído que não era possível fazer essa previsão pois o comportamento dos condutores não é fácil de prever (Zhou et al., 2017).

Dentro do que é um produto comercial que existe no mercado, a empresa Armis tem um produto chamado NEXT. De acordo com o seu website "O NEXT é uma solução para análise, simulação e previsão de dados de tráfego, meteorológicos, de incidentes, entre outros, permitindo a representação de uma rede Real "Urbana e Interurbana" através de uma rede Virtual integrando técnicas e algoritmos de previsão, com recurso a simuladores" (Administrator, 2018). As suas "funcionalidades incluem: Monitor - Monitorização da rede física real e integração de dados; Predictor - Disponibiliza previsões até 2 horas futuras; Advisor - Recomenda ações, para mitigar impactos, baseado em representações virtuais da rede; Planner - Permite efetuar a simulação e planeamento com base em eventos hipotéticos, e gerar planos de gestão e operação; Auditor - Coordena a interação entre os diversos módulos e disponibiliza análises comparativas entre o real e previsto." (Administrator, 2018). Também existem mapas online como o Bing maps que preveem o fluxo de trânsito através de ML (Polson and Sokolov, 2017), geralmente com dados obtidos por exemplo dos Smartphones dos condutores.

No contexto de previsão de trânsito a curto prazo já existem várias pesquisas sobre este assunto, (Liu et al., 2009) apresenta um modelo baseado numa rede neuronal para prever o tempo de viagem em áreas urbanas, sendo estas relacionadas com o trânsito a diferentes horas. Depois de rever a literatura de outros autores sobre a previsão do tempo de viagem baseada em modelos de regressão tais como K-Nearest Neighbors (KNN), concluindo que ainda haveria muito potencial para desenvolvimento nesta área. Os autores acreditam que um modelo indireto para a previsão do trânsito, baseado em várias variáveis e não exclusivamente no trânsito em si pode resultar em previsões superiores, pois tanto o trânsito como o tempo de viagem não têm relações causais consigo próprias mas sim com o ambiente que as rodeia.

Goves et al. (2016) estudaram a utilização de redes neuronais na previsão das condições de trânsito no Reino Unido até 15 minutos no futuro. Estes teorizam que um sistema assim poderá trazer benefícios de poder antecipar a congestão de maneira a poder tomar medidas pro-ativas para mitigar a situação. De facto, os autores afirmam que o uso de redes neuronais na previsão de trânsito não é novidade mas apenas utilizando uma rede neuronal para previsão de um único ponto ou corredor geográfico. É referido também que o uso de redes neuronais grandes tem como problemas o tempo que necessitam para treinar, os recursos necessários para as correr e os problemas herdados das características de dados de trânsito. Para este estudo foram usados cerca de 20 KM de dados de 3 estradas e resultou num modelo que 90% das vezes prevê a densidade de trânsito com sucesso.

Zhao et al. (2017) referem que durante as últimas décadas vários investigadores propuseram resolver os problemas de previsão de trânsito recorrendo a médias históricas, métodos de regressão e técnicas de ML. A previsão de trânsito recorrendo a IA está a tornar-se uma tendência. Segundo os autores, a base para um projeto de previsão e trânsito é a quantidade de dados disponíveis, no caso do seu projeto as variáveis utilizadas foram a posição do veículo, a sua velocidade, rota, etc. Para uma análise temporal com sucesso é necessário fornecer o histórico de dados como conhecimento prévio. Nesse projeto foram utilizados 25.11 milhões de dados recolhidos em 6 meses, em três pontos da cidade.

Em 2015 um estudo propôs um método de previsão da congestão de trânsito e a sua evolução, descobrindo padrões através de dados recolhidos, sendo os resultados destes posteriormente mostrados num mapa GIS de forma a poder facilmente investigar esta evolução. Os autores deste estudo afirmam que a compreensão de padrões temporais e espaciais é crucial para aliviar a congestão (Ma et al., 2015). Mais recentemente, Fabrikant (2019) desenvolveu para o Google Maps um modelo que utilizando ML treinado com várias posições de autocarros de várias empresas consegue prever os atrasos de autocarros.

Tendo em conta a literatura apresenta nesta secção, este projeto de dissertação irá distinguir-se de outros trabalhos, oferecendo uma nova perspetiva de previsão de trânsito citadino através do ponto de vista de autocarros, para uma cidade inteira e não apenas para uma área específica e permitindo previsões não só a curto prazo como também a longo prazo, podendo o produto final ser utilizado não só para análise de trânsito mas também para outras aplicações dependentes desta informação, a critério dos decisores.

3

Enquadramento Tecnológico

Neste capítulo serão abordadas as diferentes tecnologias a utilizar no desenvolvimento do projeto de dissertação, o que são, o seu propósito e o porquê de terem sido escolhidas.

3.1 SQL e MongoDB

Structured Query Language (SQL) no contexto deste projeto é uma necessidade, não uma escolha. SQL tem grandes limitações na expressão de termos linguísticos (Agarwal et al., 2017). MongoDB é uma base de dados NoSQL, é utilizado por grandes companhias como a Disney e o GitHub. Para além disso, para uma grande quantidade de dados é extremamente mais rápido e mais escalável do que outros sistemas de bases de dados do tipo SQL. Enquanto que por exemplo uma operação de inserir 1000000 dados leva 882078 milissegundos em SQL, no MongoDB leva apenas 57871 milissegundos (Van Der Veen et al., 2012). Tendo em conta o grande número de registos com o qual se vai lidar neste projeto, o MongoDB foi considerada a melhor opção, sendo que os dados deverão ser movidos do SQL para a MongoDB logo que possível.

3.2 R

R é uma linguagem de programação estatística Open Source. Esta linguagem permite profissionais correr praticamente qualquer análise imaginável num qualquer conjunto de dados. Devido à comunidade que constantemente adiciona conteúdo a esta ferramenta, ela é agora uma referência para investigadores de ML e DM. É muito provável encontrar um bom algoritmo já desenvolvido nesta ferramenta (Berthold, 2012). De acordo com a página web do CRAN, o seu repositório é neste momento de quase 14000

pacotes para o R (CRAN). O facto desta ferramenta ser excelente para DM, ser Open Source e já incluir imensos algoritmos faz dela um recurso extremamente útil no desenvolvimento deste projeto.

3.3 Python

Python é uma excelente linguagem de coordenação por si só. É uma linguagem interpretada com uma sintaxe expressiva. É também uma linguagem Open Source que corre em múltiplas plataformas e tem imensos módulos com os quais pode-se transformar numa linguagem de alto nível própria para uso científico (Oliphant, 2007). Por exemplo, para o trabalho em questão existe um módulo denominado de Orange que pode ser usado para ML e DM que inclui ferramentas de tratamento de dados, classificação, regressão, clustering entre outros (Demšar et al., 2013).

De acordo com o repositório PyPi existem quase 165000 projetos (PyPI). Um destes projetos denominado de rpy2 permite executar funções R dentro de scripts Python (rpy). Como Python é extremamente versátil, neste projeto vai servir o papel de estrutura, tudo aquilo que não é base de dados e visualização poderá ser programado em Python, incluindo, se necessário os algoritmos de DM.

3.4 Algoritmos

Antes de escolher os algoritmos a utilizar será necessário testar vários e comparar o seu desempenho. Tendo em conta o objetivo de previsão via uma abordagem de regressão, os algoritmos escolhidos foram regressão múltipla, as árvores de decisão, o random forest, rede neuronal, SVM e KNN, sendo que o funcionamento de cada um destes algoritmos está resumido na Tabela 5.

3.5 Ferramentas de Visualização

A visualização de dados é um componente essencial de qualquer sistema informático. É a componente que os utilizadores vão ver e interagir e convém ser o mais simples e intuitiva possível. Para este trabalho serão utilizadas ferramentas de desenvolvimento web, como HTML, CSS e JavaScript, não só porque é necessário para integrar como um serviço no já mencionado sistema Edge, mas também porque permite desenhar o interface à medida do que se pretende mostrar e mais importante ainda é compatível com qualquer sistema operativo que possa utilizar um browser. Sendo que neste momento a maioria dos acessos à internet são feitos através de dispositivos móveis, é importante que se possa visualizar estas

informações a partir de qualquer local, pois decisões de negócio não precisam de ser feitas necessariamente a partir de um escritório. Além disso, se surgir uma nova categoria de dispositivos no mercado que venha a destronar os smartphones, é muito provável que esta tenha também um browser no seu sistema.

3.6 Dados

Como já foi referido anteriormente os TUB enviaram dados em formato SQL. Estes dados ocupam um espaço à volta de 50 Gigabytes, dos quais já foram selecionados aqueles importantes para o trabalho em questão. Existem duas tabelas que são úteis para este projeto, a que contém a localização das paragens dos autocarros e a que contém os dados das posições dos autocarros.

Sendo que estes dados são parte de uma solução genérica adquirida pelos TUB e contêm campos não utilizados com valores nulos, colunas com valores repetidos entre outros, devido ao facto de não ser algo feito à medida para a empresa, só serão referidos os atributos que têm interesse para este projeto dentro das tabelas importantes. Relativamente às paragens, cada uma tem a sua própria entrada na tabela. Estas contêm identificadores de zona numéricos que poderão ser interessantes para relacionar com o fluxo de trânsito, uma descrição em formato de texto com o nome da paragem, um raio em formato numérico que se refere à distância que o autocarro pode estar da paragem e ainda ser detetado como estando nela, e valores de X e Y, que serão explorados mais à frente.

Para a tabela sobre as posições dos autocarros, existe um X, Y e Z tal como nas paragens, um identificador do serviço em formato numérico, uma timestamp com data e hora à qual foi registada aquela entrada, uma coluna de estado, numérica, que se refere a se o autocarro está a ir ou a voltar e uma coluna com a velocidade a que o autocarro ia no momento, numérica. Quanto à questão dos formatos X, Y e Z, estes causaram um pequeno problema inicialmente pois não são os habituais valores de latitude e longitude com os quais a maioria dos sistemas posicionais trabalha. Depois de serem inseridos em conversores online de vários formatos para latitude e longitude e comparados os valores com as paragens que estavam representas no Google Maps, foi descoberto que o tipo de posicionamento utilizado por este sistema é Sistema Universal Transverso de Mercator (UTM) utilizando como zona para a conversão, o valor 29T que inclui a zona norte de Portugal. Foi criada uma pequena script Python, recorrendo ao módulo utm para a conversão destes valores, obtendo-se resultados muito bons, sendo a diferença compensável com o valor do raio.

4

Desenvolvimento

Neste capítulo será detalhado todo o processo de desenvolvimento constituído pelas diversas fases, tanto respetivas à componente de DM como também os componentes relativos ao servidor e visualização. Este capítulo será estruturado de acordo com a metodologia CRISP-DM.

A secção correspondente à compreensão do negócio detalha o funcionamento da área de negócio que se está a modelar. A secção de compreensão dos dados apresenta a informação relativa ao tipo de dados, quantidade e qualidade com que se está a trabalhar e as respetivas fontes. Durante a secção da preparação de dados explica-se como os dados foram convertidos, transformados e juntados de maneira a produzir informação relevante para a modelação. Para a secção da modelação demonstra-se o processo de DM que ocorreu. Na secção de avaliação descreve-se o porquê da seleção do modelo escolhido e as garantias do mesmo. Na secção de discussão de resultados detalha-se todo o processo decorrido, observações sobre os modelos trabalhados e tentativas de resolução do problema em questão de previsão de trânsito. Nas secções relativas às componentes de servidor e visualização, explica-se de modo sucinto como foram criadas e qual o seu funcionamento. Na Figura 8 está representada a arquitetura da solução final.

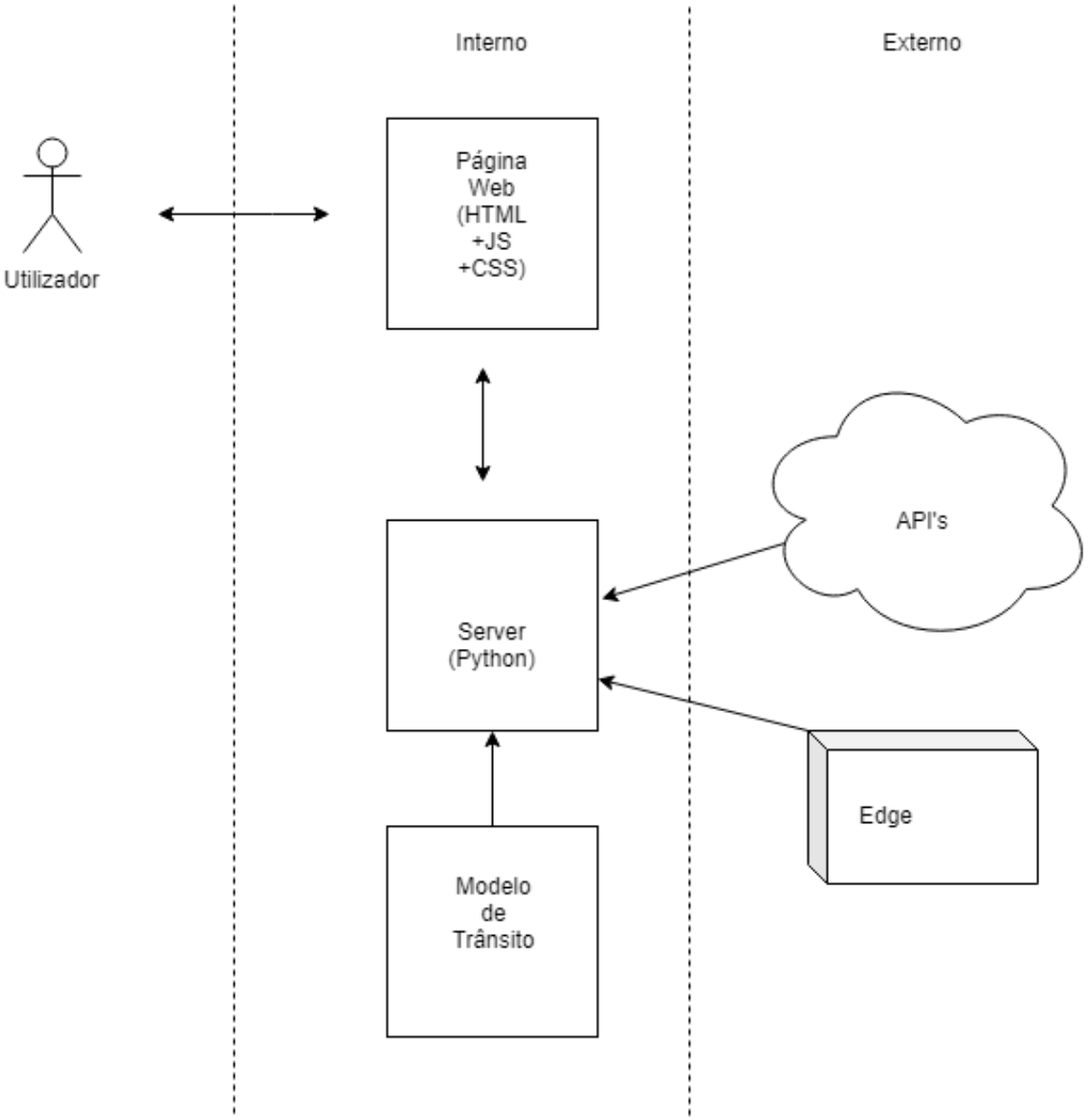


Figura 8: Arquitetura da solução final

4.1 Compreensão do Negócio

Quando se tenta compreender o tipo de viagens realizadas por um motorista de transportes públicos, por vezes há uma tendência de comparar estas às comutas realizadas por um condutor de veículos ligeiros, ignorando assim as particularidades das viagens de uma empresa como os TUB. A parte deste negócio que se está a modelar funciona da seguinte maneira: um condutor inicia uma rota começando por ligar o autocarro com os devidos procedimentos, segue uma rota pré-definida que passa obrigatoriamente nas paragens pré-designadas onde abranda e pára para recolher e deixar passageiros. Se o condutor estiver

adiantado, este tem que parar o autocarro e esperar pela hora em que deve continuar viagem. Em geral quando um autocarro termina uma rota este deve regressar à central. Na central ou outros lugares definidos pelos TUB os autocarros podem sofrer diferentes procedimentos como manutenção e lavagem. Quando os autocarros não estão em circulação, estão em lugares designados como a central. As rotas não são flexíveis, não podendo ser alteradas a qualquer momento.

Os TUB operam na sua maioria em Braga mas há casos especiais onde os seus autocarros são registados a sair em viagens para outras regiões. As diversas rotas pré-definidas têm uma particularidade de terem designações internas diferentes internamente das dadas a conhecer publicamente, por exemplo uma rota que ao público é conhecida como rota 30 que é realizadas às 14h, 16h e 20h, para os TUB são conhecidas como trips e têm três identificadores diferentes como 30_14, 30_16 e 30_20 (números exemplificativos). Existem sempre rotas a serem efetuadas em dias úteis, sábados, domingos e feriados, tanto de dia como de noite, a quase todas as horas. Muitas destes casos só puderam ser descobertos após observações efetuadas em modelos já treinados, na fase de avaliação, levando a que fosse necessário adotar medidas para a identificação destes problemas específicos, comunicar novamente com os especialistas do negócio e continuar o processo a partir da fase de Compreensão do Negócio com uma nova perspetiva sobre o mesmo. É também necessário considerar sempre que a perspetiva pela qual este projeto vê o negócio é sempre através dos dados GPS recolhidos pelos autocarros. Autocarros não são apenas carros grandes. Eles param em paragens de autocarros; demoram mais a acelerar, desacelerar, e virar; e por vezes até têm privilégios especais nas estradas (Fabrikant, 2019).

4.2 Compreensão dos Dados

Os TUB, para a execução deste projeto de dissertação, forneceram dados relativos à posição da sua frota de autocarros desde 1 de janeiro de 2016 até o dia em que ocorreu a extração dos dados, em agosto de 2018. Ao todo este extrato de uma base de dados SQL continha mais de 223 milhões de entradas, cada uma correspondendo a um único ponto de um único autocarro num único espaço e num único momento. Para além destes dados fornecidos pelos TUB, foi necessário para este projeto obter dados complementares, dados estes que devem ser adquiridos antes da preparação de ambos os conjuntos, pois conjunto de saída necessita de ser uma junção de ambos. Dos conjuntos de dados adicionais procurados foi possível obter dados relativos à meteorologia, eventos e feriados. Relativamente à meteorologia foram descarregados dados em formato xlsx da página web rp5, que contem dados históricos de várias estações meteorológicas. Os eventos foram criados manualmente utilizando informações da página

oficial da câmara municipal de Braga e páginas de desporto, esta análise seria muito provavelmente superior caso estas informações estivessem em formatos digitais legíveis que não exigissem uma abordagem forçada para a sua aquisição. Os dados relativos a feriados foram obtidos em tempo de execução do tratamento recorrendo ao pacote disponível para Python chamado holidays, que não só documenta feriados portugueses como também contém uma lista de dias com propriedades semelhantes.

Cada entrada na tabela, representa uma posição de um autocarro, esta contém a data e hora a que foi retirada e a sua posição em formato UTM, entre outros dados não relevantes para o projeto em questão e que não serão lidos da base de dados. Apesar de virem incluídos, em vez de utilizar os dados relativos às paragens da base de dados, foram utilizados os dados General Transit Feed Specification (GTFS) referentes às paragens de autocarros das rotas dos TUB. Este novo formato é não só mais simples de compreender, mas também é muito mais eficiente para a execução da tarefa de tratamento dos dados, discutida a detalhe na secção seguinte, para além de ser o padrão para o qual a indústria está a evoluir e, caso seja necessário atualizar os mesmos basta apenas descarregar a informação mais recente. Na Tabela 3 pode ser consultada a estrutura dos dados recebidos.

Todos os dados utilizados são bastante simples de compreender por si, sendo que a sua complexidade vem do contexto em que estão registados. A combinação destes dados gerou por fim um número de atributos utilizados na modelação, que serão detalhados nas próximas secções. Em geral todos os dados apresentam uma grande qualidade sendo que as exceções serão filtradas durante o tratamento dos mesmos. Um extrato dos dados originais pode ser observado na Figura 9. A Figura 10 contém uma representação geográfica parcial dos dados fornecidos pelos TUB.

Results Messages

	POS_IDSERVICIO	POS_X	POS_Y	POS_Z	POS_RUMBO	POS_VELOCIDAD	POS_CALIDAD	POS_FECHA	POS_ESTADO	rowguid
1	[illegible]	548415.00	4599455.00	172.00	172	0	7	2018-02-12 15:54:40.000	2	[illegible]
2	[illegible]	547571.00	4600041.00	180.00	90	0	8	2018-02-12 15:54:40.000	2	[illegible]
3	[illegible]	547291.00	4602738.00	83.00	168	36	9	2018-02-12 15:54:41.000	2	[illegible]
4	[illegible]	548828.00	4599915.00	181.00	240	46	8	2018-02-12 15:54:41.000	1	[illegible]
5	[illegible]	547359.00	4600155.00	173.00	232	33	9	2018-02-12 15:54:41.000	1	[illegible]

Figura 9: Extrato dos dados originais fornecidos pelos TUB.

Coluna	Descrição	Destino	Motivo
Identificador	Identifica a rota que o autocarro estaria a fazer no momento do registo.	Não utilizado.	O identificador não pode ser decifrado em tempo útil.
X/Y/Z	Combinados indicam a posição do autocarro em formato UTM.	Utilizado.	Estes são os dados base sob os quais todo o projeto decorre.
Rumo	Indica o ângulo da direção em que o autocarro segue.	Não utilizado.	Não acrescenta nenhum valor à informação posicional pois não há como ligar as várias posições entre si e o mesmo poderia ser deduzido através das localizações caso necessário tendo em conta a sua frequência.
Velocidade	Indica a velocidade do autocarro no momento.	Utilizado.	Esta é a variável que vai indicar a existência ou não de trânsito na zona.
Qualidade	Indica a qualidade do registo numa escala de 1 a 10.	Não utilizado.	Embora possa parecer algo necessário, se algum registo for terrível este será detetado e removido durante o tratamento, a utilização de geohash no projeto fará com que o desvio do GPS não seja algo com o qual se tenha de lidar.
Data	Data e hora do registo.	Utilizado.	Essencial para se estabelecer as relações necessárias no modelo de previsão.
Estado	Indica se o autocarro está a ir ou a vir da rota.	Não utilizado.	Não é útil sem poder estabelecer relações entre as posições.
Identificador da Base de Dados	Identificador único.	Não utilizado.	Não é útil.

Tabela 3: Considerações na escolha dos atributos dados a utilizar

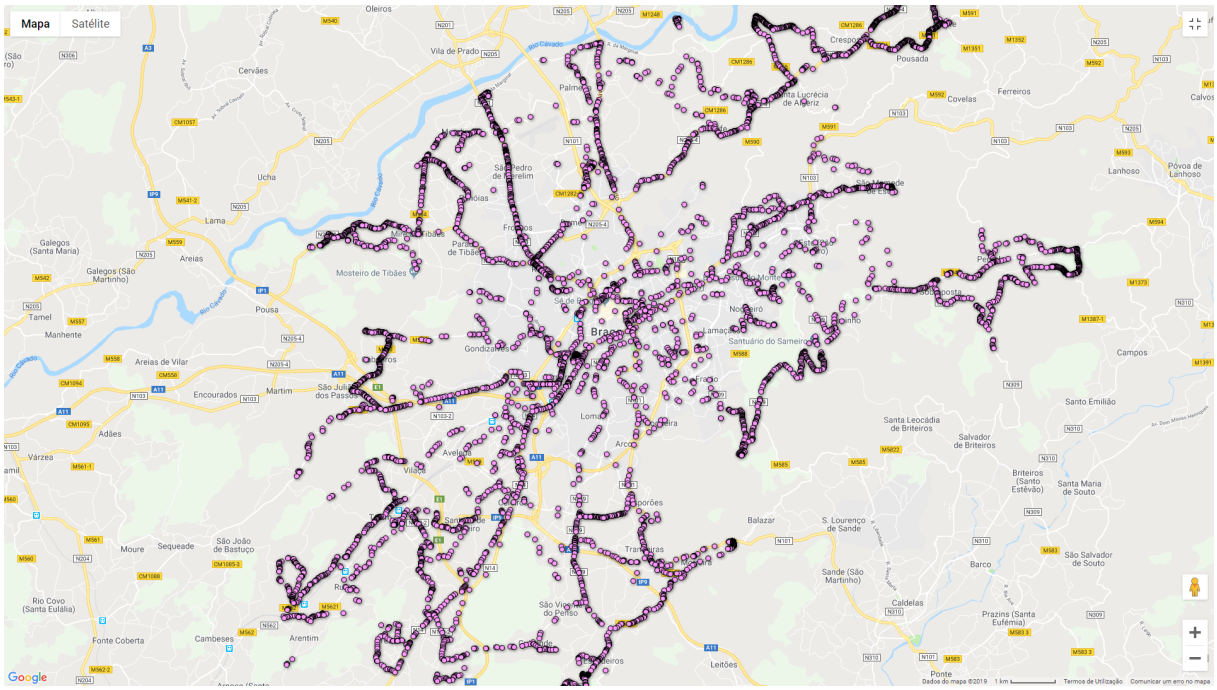


Figura 10: Visualização geográfica de um extrato dos dados obtidos após a sua preparação.

Sendo o principal objetivo deste projeto a criação de um modelo que possa prever o trânsito esperado no futuro dado uma certa combinação de eventos, tem que se ter em conta todas as peculiaridades deste tipo de negócio, pois como é demonstrado pelos testes realizados durante o desenvolvimento deste trabalho, uma entrada na base de dados nem sempre se traduz numa realidade do trânsito. O número de entradas às quais este projeto tem acesso excede os 223 milhões, com imensas combinações possíveis entre as diferentes colunas, sendo qualquer valor geográfico válido, registado em mais de 20 mil horas com cerca de 120 possíveis velocidades, assumindo nenhum erro de registo ou velocidade excessiva. Não sendo este um número elevado do ponto de vista do negócio, para o projeto em questão e utilizando a definição de Chen et al. (2014) este número mais do que excede os requisitos para ser considerado big data, olhando aos recursos disponíveis para a execução deste projeto.

4.3 Preparação dos Dados

Com os dados analisados e escolhidos procedeu-se à preparação dos mesmos. Foi necessário realizar o tratamento para modificar o tipo de dados e juntar os conjuntos complementares aos mesmos formando uma única tabela, de forma a transformar os dados obtidos em informação útil. Tendo em conta que existiram várias iterações desta fase, apenas será detalhada a última versão deste tratamento.

Serão anexadas partes do código utilizado para o tratamento de dados para uma melhor compreensão de como determinadas tarefas foram feitas, este código estará escrito na linguagem de programação Python com algumas modificações para uma compreensão melhor do processo realizado. Esta linguagem foi escolhida pois permite um grande controlo sob o processo de tratamento, pode-se utilizar várias fontes e saídas independentemente do seu formato e os módulos já desenvolvidos disponíveis permitem acelerar o processo de transformação, por exemplo a conversão de dados GPS UTM para latitude e longitude é muito mais simples recorrendo a um pacote do que escrevendo uma função nova.

Enquanto um autocarro está ligado, este recolhe dados, quer esteja a fazer uma rota, à espera de fazer uma rota ou movendo-se dentro de uma instalação dos TUB ou de manutenção. Este sistema de recolha de dados faz com que cerca de 33% de todas as entradas tenham de ser eliminadas. Inicialmente foi verificado se o autocarro numa certa posição com velocidade igual a zero estaria perto de uma paragem, mas posteriormente foi criado uma área vazia à volta de cada paragem que elimina qualquer entrada que entre nesta zona. Foram também removidos os valores posicionais relativos a entradas ocorridas dentro da central dos TUB ou em qualquer outra zona identificada de estacionamento ou manutenção, aplicando funções de maior e menor sobre as latitudes e longitudes de maneira a representar os formatos destas instalações o melhor possível.

Primeiro foi escrita uma query SQL simples para dividir a base de dados inicial em 9 partes. Isto foi necessário principalmente porque não existiria memória suficiente para carregar todos os dados ao mesmo tempo mas teve adicionalmente a vantagem de no caso se ocorrer um erro algumas das partes já estariam tratadas, podendo a próxima execução ser iniciada a partir da parte onde o erro ocorreu. Tal é importante porque o processo demora 3 dias a terminar e um imprevisto poderia atrasar ainda mais o processo, caso esta medida não tivesse sido tomada. Posteriormente foi criada uma script em Python que acede à base de dados e carrega os dados para memória. Para isso foi utilizado o módulo pyodbc com o driver disponibilizado pela Microsoft, (Figura 11). Para além de se transformar os dados, foram aplicadas verificações para garantir que nenhum valor passado contivesse informação imprecisa ou errada. Assim, foram aceites valores não nulos que se enquadrassem no tempo entre 2016 e 2018, sendo o segundo representado pela data retirada no dia em que a script original foi executada, o que eliminou 6 entradas relativas a dados do ano 1990 que também continham valores nulos.

```python
def sqlSelect( from_, select_ = "*", where_ = "", server_="[redacted]",username_="[redacted]",password_="[redacted]",database_="[redacted]"):
    query = "SELECT " + select_ + " FROM " + from_
    if where_ != "":
        query = query + " WHERE " + where_
    try:
        cnxn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};SERVER='+server_+';DATABASE='+database_+';UID='+username_+';PWD='+ password_)
        cursor = cnxn.cursor()
        cursor.execute(query)
        row = cursor.fetchall()
        cnxn.close()
    except Exception as e:
        raise e
    return row
```

Figura 11: Script Python com a função SQL

```python
import pyodbc
import json
import utm
import csv
import datetime
from datetime import date
import holidays
import pygeohash as pgh
from pymongo import MongoClient
import geopy.distance

#extrato de codigo (alterado para representar mais fielmente a versao final)
def sqlLab(data,x=0,y=0,time=0,speed=0,collection=""):
    today = datetime.datetime.today()
    reasonable = datetime.datetime(2016,1,1,00,00)
    for row in data:
        if row[time] < today and row[time] > reasonable and row[x]!=0 and row[x]!=-1:
            day = whatday(row[time].weekday()) #dia da semana
            eferiado = row[time] in feriados  #feriado
            coord = coordinator(row[x],row[y]) #coord 0 lat, coord 1 lon
            area = pgh.encode(coord[0], coord[1], precision=7) #geohash para comparacoes
            myclima = clima(day=row[time].day, month=row[time].month, year=row[time].year, hour=row[time].hour)
            nns = notNearAStop(coord[0], coord[1],area) #funcao "representativa" para apagar valores perto de paragens
            if nns:
                event = evento(row[time].day, row[time].month, row[time].year, row[time].hour, area[:6]) #evento
                jzon = {'lat':coord[0], 'lon':coord[1], 'speed':row[speed], 'month':row[time].month,
                'hour':row[time].hour, 'dayofweek':day, 'feriado':eferiado,
                'temp':myclima['temp'], 'rain':myclima['rain'], 'event':event}
                collection.insert_one(jzon) #base de dados temporaria mongoDB
```

Figura 12: Script Python com a função representativa do tratamento de dados

A data e hora foram divididas e em vez de um formato data, foram utilizados números inteiros, mantendo apenas o mês e a hora, cada uma com a sua coluna, pois o dia e o ano não fariam sentido de um ponto de vista de modelação. Por exemplo, não se espera que por ser dia 17 haja mais trânsito do que se fosse dia 24, o mesmo aplicando-se ao ano, não fazendo sentido haver mais trânsito em 2018 do que em 2016, sendo que se realmente existisse esta diferença, dar-se-ia a fatores externos não relacionados com dados posicionais de trânsito. Por outras palavras, pode até existir uma correlação mas isso não implica a existência de uma causa baseada em factos que é provável manter-se constante no futuro. Os minutos foram inicialmente utilizados mas a sua inclusão não trouxe vantagens sendo removidos na versão final. A coluna mês contém 12 valores possíveis. A sua distribuição varia conforme é mostrado na Tabela 4 e na Figura 13, observando o padrão de que estes valores podem dever-se ao fim do período escolar nos meses de junho a setembro, acompanhado pelos períodos de férias normais no Verão e Inverno e também pelo facto que parte do mês de 2018 não foi registado enquanto a primeira metade foi. Não tendo

conhecimento do quanto esta variação provém do negócio e o quanto provém de fatores externos, estes dados não serão removidos. Isto não deve representar um problema para o modelo pois a quantidade de dados não deve influenciar o trânsito sentido naqueles meses e é uma representação fiel dos dados originais.

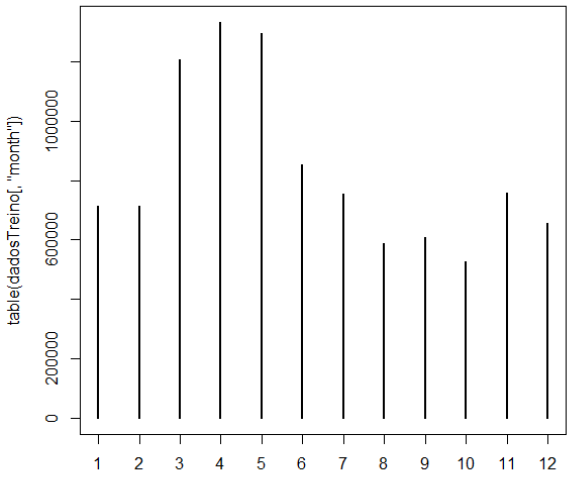


Figura 13: Frequência dos dados da coluna mês.

A coluna hora contém 24 valores possíveis. Representando estes valores a hora, a sua distribuição segue o esperado, havendo muito poucos registos durante a noite e uma grande quantidade durante o dia com máximos na hora de ponta, como pode ser observados na Tabela 4 e na Figura 14.

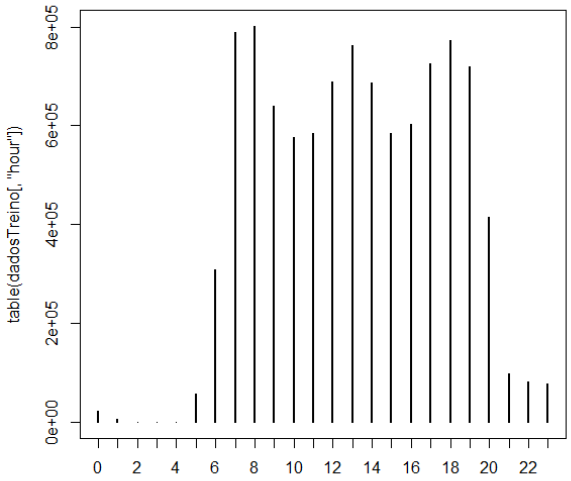


Figura 14: Frequência dos dados da coluna hora.

Utilizando o valor temporal obtido da base de dados e recorrendo a módulos Python foram criadas novas colunas para análise do modelo. Utilizando a função weekday() do Python foi possível obter um número que representa o dia da semana (exemplo: 1=terça). Estes valores foram convertidos para texto, de maneira a facilitar a leitura, mas devido ao desempenho dos valores numéricos estes valores voltaram a ser convertidos em números inteiros sob uma escala ligeiramente diferente de 1 a 7, domingo a sábado. Espera-se com esta coluna obter uma relação entre o dia da semana e o trânsito. A coluna relativa ao dia da semana tem 7 valores possíveis. Pode ser observado que, como seria de esperar, a distribuição é bastante semelhante durante os dias úteis sendo apenas significativamente diferente durante o fim de semana, quando há menos trânsito e menos rotas a serem feitas, tal como pode ser verificado na Tabela 4 e na Figura 15.

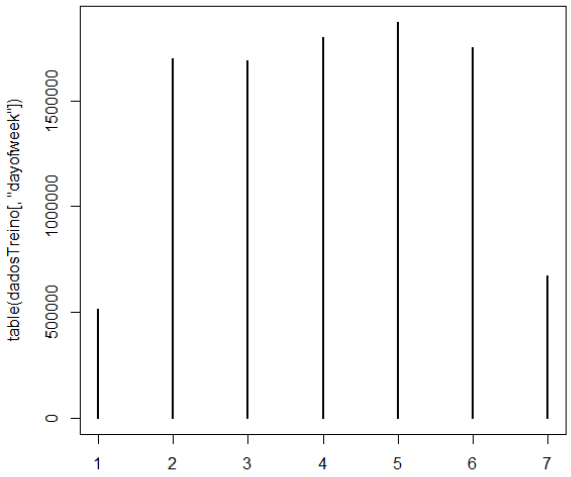


Figura 15: Frequência dos dados da coluna dia da semana.

Utilizando o pacote holidays foi criada uma nova coluna que indica a existência de um feriado nesse dia, se existir o valor será 1 e se não existir o valor será 0. Foi criada também uma variável semelhante que representa a existência de um evento, no entanto para além de verificar o dia do acontecimento, este também verifica a área (geohash) em que o o evento decorre e se coincide com a área (geohash) do autocarro. As colunas feriado e evento têm apenas 2 valores possíveis, 0 e 1. Pode ser observado na Tabela 4 que existem muitos mais valores de 0 do que 1. Isto é esperado visto que existem mais dias onde não existe feriados e eventos do que aqueles que os têm e muitas vezes os dias com eventos coincidem com feriados.

Foram adicionadas também 2 variáveis correspondentes ao clima, a temperatura e a pluviosidade, ambas verificam apenas o dia e hora, pois assume-se que o mesmo clima é sentido por toda a cidade. Utilizando a fonte de dados obtida do clima, foi percorrido o ficheiro para cada entrada e assim obtido o valor da temperatura e a pluviosidade para aquele dia. Pode ser observado nas Figuras 16 e 17 e na Tabela 4, uma representação bastante fiel da distribuição de condições que normalmente se fazem sentir no distrito de Braga, sendo os únicos comentários relevantes a serem feitos, relativos ao facto de existir temperaturas aparentemente demasiado baixas, as quais registam-se maioritariamente durante a noite. Para a coluna temperatura existem valores registados entre -2 e 35 e para a coluna chuva existem 4 valores possíveis de 0 a 3, numa escala onde 0 corresponde à ausência de pluviosidade ou existência de valores os quais não se fazem sentir e 4 correspondente a valores de pluviosidade extrema. Assim, tem-se que: 0 corresponde a valores inferiores a 2.5 mm/h, 1 corresponde a valores iguais ou superiores a 2.5

mm/h e 10 mm/h, 2 corresponde a valores iguais ou superiores a 10 mm/h e 50 mm/h e 3 corresponde a valores superiores a 50 mm/h.

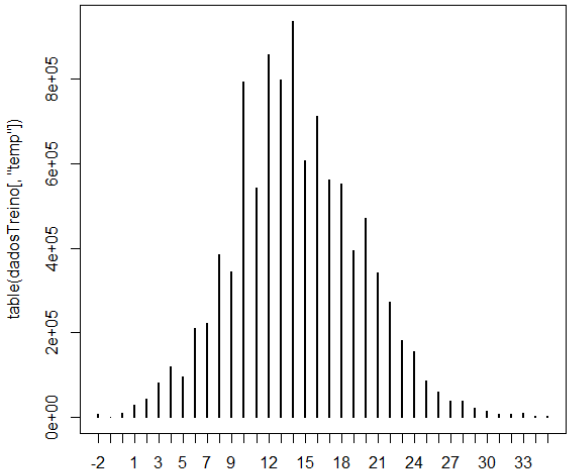


Figura 16: Frequência dos dados da coluna temperatura.

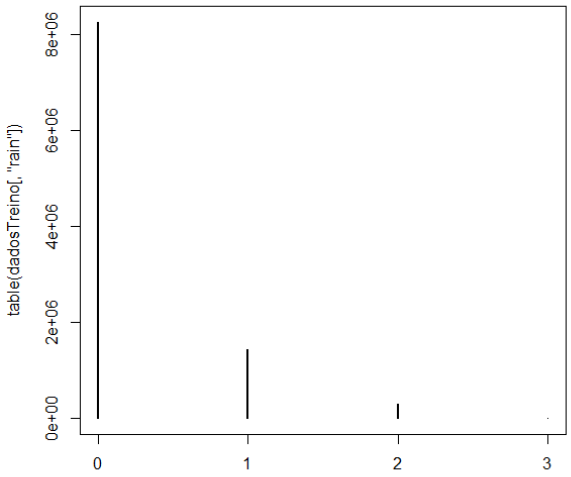


Figura 17: Frequência dos dados da coluna chuva.

Ocorreu também a transformação de dados UTM em dados de latitude e longitude, que são mais comuns e mais simples de entender. As colunas resultantes deste processo: lat e lon, representantes da latitude e longitude também têm um destaque na forma dos gráficos representados nas figuras: Figura 18 e Figura 19, respetivamente. Nestas figuras pode ser observado uma tendência interessante de haver

mais valores no meio do gráfico ao invés de ser distribuído uniformemente por todos os valores. Isto acontece porque a maioria dos dados recolhidos são, por natureza do negócio, provenientes dos espaços mais centrais da cidade, o que implica não só mais trânsito como também representa o local onde os autocarros mais tempo passam em maior quantidade. O facto da concentração de dados se dar no centro de Braga e este ser o mesmo centro do gráfico deve-se apenas à geografia da cidade, podendo variar para análises de cidades diferentes.

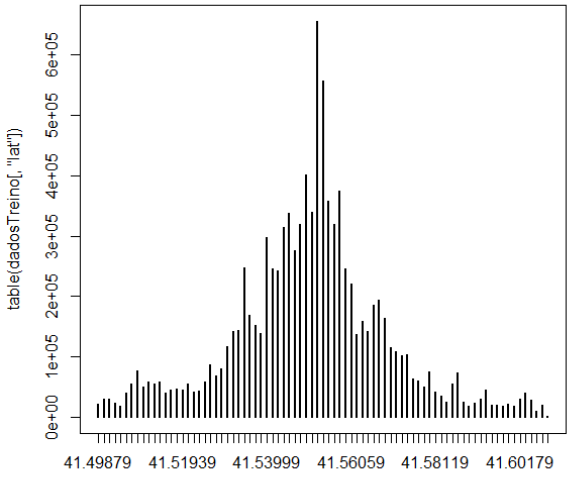


Figura 18: Gráfico da distribuição dos valores da latitude.

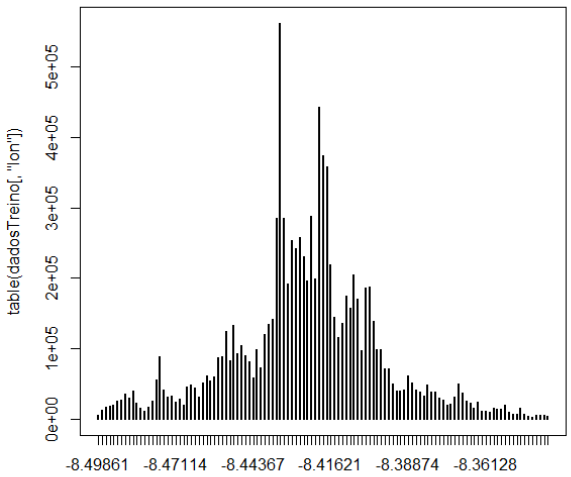


Figura 19: Gráfico da distribuição dos valores da longitude.

Atributo	Valor	Quantidade	Valor	Quantidade	Valor	Quantidade	Valor	Quantidade
rain	0	8257098						
	1	1440531						
	2	295602						
	3	6769						
temp	-2	6340	8	384693	18	551836	28	37136
	-1	691	9	343612	19	393289	29	20652
	0	9770	10	792679	20	471873	30	13002
	1	27399	11	542966	21	340839	31	7712
	2	41678	12	859358	22	271571	32	5930
	3	80866	13	798251	23	182282	33	8705
	4	120135	14	937487	24	155112	34	1199
	5	95841	15	606112	25	84722	35	1798
	6	210775	16	711956	26	59274		
	7	220984	17	562787	27	38688		
event	0	9996719						
	1	3281						
feriado	0	9813698						
	1	186302						
dayofweek	1	512774	5	1871506				
	2	1699732	6	1751662				
	3	1690177	7	675007				
	4	1799142						
hour	0	21835	6	308450	12	688129	18	772504
	1	6482	7	788509	13	761734	19	718649
	2	261	8	801507	14	687145	20	414698
	3	129	9	639213	15	584702	21	98887
	4	570	10	576453	16	603953	22	80625
	5	56826	11	584629	17	725989	23	78121
month	1	713789	7	755756				
	2	711717	8	587360				
	3	1205375	9	609154				
	4	1332209	10	525953				
	5	1294929	11	756142				
	6	853684	12	653932				

Após o tratamento, foi por fim selecionado um conjunto de dados para utilização dos modelos. Devido a restrições de poder computacional, foi necessário reduzir os dados de maneira a poder executar modelos nos computadores disponíveis. Para tal efeito, foram executados comandos incluídos no sistema operativo linux que permitem baralhar e cortar quantidades massivas de dados numa questão de minutos. Para realizar esta última ação recorreu-se ao comando "tail -n +2 dadosTotais.csv | shuf -n 44265534 -o speed20.csv" na bash do linux; este comando pode ser lido como "após ler os dados totais a partir da segunda linha (para evitar o cabeçalho), baralha 44265534 linhas (20% dos dados) e grava-as no ficheiro speed20.csv, reduzindo assim os dados a 44 milhões de dados, um valor que é bastante inferior aos 223 milhões com os quais o projeto foi iniciado. Voltou-se a baralhar esses 40 milhões de dados e retirou-se 20 milhões, que após carregados para o R, metade dos registos foram utilizados como dados de teste, sendo os restantes utilizados como dados de treino. Verificou-se que ambos os conjuntos seguiam uma distribuição semelhante aos 44 milhões dos quais originaram, concluindo-se assim que as amostras estariam balanceadas. O número 10 milhões não foi escolhido arbitrariamente, após vários meses de testes e transformações de dados foi descoberto que um número superior não aumentaria a precisão do modelo o suficiente para compensar o tempo requerido para os treinos, sendo que em alguns casos o tempo de execução aumentaria de 5 dias para valores de meses.

Os dados utilizados para o conjunto de treino serão agora descritos. Sendo estes dados uma boa representação do conjunto inicial, pode-se assumir que todos estes seguem a mesma distribuição e contém as mesmas classes. Qualquer exceção a esta regra não representa um conjunto significativo o suficiente para afetar os resultados finais. Recorrendo às funções nativas do R poder-se-á fazer uma análise detalhada dos conjuntos com facilidade. Em primeiro lugar pode-se observar na Figura 20 uma pequena extração dos dados com os quais foram executados os treinos dos modelos, seguido pela Figura 21 que contém um resumo dos dados, com as suas diversas distribuições representadas por quartis.

	lat	lon	speed	month	hour	dayofweek	feriado	temp	rain	event
1	41.55647	-8.39561	44	12	14	3	0	9	0	0
2	41.56059	-8.41758	37	2	18	4	0	13	2	0
3	41.54823	-8.39012	27	5	13	6	0	14	1	0
4	41.55235	-8.41484	4	2	16	2	0	10	1	0
5	41.55372	-8.42994	30	7	18	3	0	21	0	0
6	41.53861	-8.41209	42	10	18	7	0	16	2	0
7	41.55097	-8.41758	5	12	15	4	0	13	0	0
8	41.55235	-8.41484	32	7	14	6	0	23	0	0
9	41.55509	-8.40110	16	2	8	4	0	12	0	0
10	41.56608	-8.48075	42	10	16	5	0	12	0	0
11	41.55509	-8.40797	6	11	20	2	0	11	0	0
12	41.58668	-8.39286	18	8	23	6	0	18	0	0
13	41.54136	-8.43681	30	3	8	3	0	10	0	0
14	41.54823	-8.42033	1	12	9	4	0	10	0	0
15	41.51115	-8.45466	28	5	9	4	0	21	0	0
16	41.54411	-8.36402	36	4	10	4	0	13	0	0

Figura 20: Extrato dos dados utilizados.

lat	lon	speed	month	hour	dayofweek	feriado	temp
Min. :41.50	Min. :-8.499	Min. : 1.00	Min. : 1.000	Min. : 0.00	Min. :1.000	Min. :0.00000	Min. :-2.00
1st Qu.:41.54	1st Qu.:-8.437	1st Qu.: 18.00	1st Qu.: 3.000	1st Qu.: 9.00	1st Qu.:3.000	1st Qu.:0.00000	1st Qu.:11.00
Median :41.55	Median :-8.424	Median : 28.00	Median : 5.000	Median :13.00	Median :4.000	Median :0.00000	Median :14.00
Mean :41.55	Mean :-8.425	Mean : 28.49	Mean : 5.957	Mean :13.21	Mean :4.077	Mean :0.01863	Mean :14.35
3rd Qu.:41.56	3rd Qu.:-8.411	3rd Qu.: 39.00	3rd Qu.: 9.000	3rd Qu.:17.00	3rd Qu.:5.000	3rd Qu.:0.00000	3rd Qu.:18.00
Max. :41.61	Max. :-8.339	Max. :119.00	Max. :12.000	Max. :23.00	Max. :7.000	Max. :1.00000	Max. :35.00

rain	event
Min. :0.0000	Min. :0.0000000
1st Qu.:0.0000	1st Qu.:0.0000000
Median :0.0000	Median :0.0000000
Mean :0.2052	Mean :0.0003281
3rd Qu.:0.0000	3rd Qu.:0.0000000
Max. :3.0000	Max. :1.0000000

Figura 21: Sumário do conjunto de dados de treino.

Quanto a estes últimos dados, nem todos os modelos foram treinados utilizando a latitude e longitude. Os modelos baseados em regressão linear e redes neuronais foram treinados utilizando 5 colunas representativas de geohashs, pois estes demonstraram um melhor desempenho tanto durante o treino como durante a previsão. Devido à natureza dos outros modelos o mesmo não se demonstrou verdade recorrendo-se para estes novamente à latitude e longitude. Como exemplo do que foi utilizado, descreve-se a geohash "ez6h2z6", sendo que os 2 primeiros caracteres vão ser sempre iguais para qualquer parte de Braga estes são omitidos, em seguida divide-se o resto em diferentes colunas denominadas, "a", "b", "c", "d" e "e" onde na "a" estará o 6, na "b" o h, na "c" o 2, na "d" o z e na "e" o 6. A estrutura do geohash permite que isto seja possível, pois cada letra ou número corresponde a uma divisão do mapa global sendo os caracteres mais à direita os mais detalhados. A Tabela 5 condensa todas as informações relevantes quanto aos modelos usados. Na secção de "Resultados" existe uma outra tabela que contém os valores das várias métricas.

Esta foi sem dúvida uma das fase que mais tempo e esforço exigiu, estando equiparado com a fase de modelação pois estas foram repetidas diversas vezes ao longo da execução do projeto devido à natureza

iterativa do CRISP-DM. Uma execução inicial sobre os dados originais demorava cerca de 3 dias a terminar, devido a isso muito do tratamento decorreu sobre os dados já tratados de outras fases sempre que possível. O formato mais utilizado para este tipo de execuções foi o csv, pois é mais rápido do que ler diretamente da base de dados e tem como vantagem poder utilizar o terminal do sistema operativo Linux para extrair amostras.

4.4 Modelação

Antes de iniciar a modelação por si é necessário ter em consideração que tipo de problema se está a tentar resolver. Existem dois tipos principais de problemas sendo eles classificação onde tenta-se prever uma variável categórica e a regressão onde se tenta prever uma variável contínua. Neste projeto ambas as abordagens seriam possíveis e válidas tendo-se começando por tentar uma abordagem de classificação trocando-se esta por uma abordagem de regressão após a última demonstrar fazer mais sentido e ter melhores resultados. A importância desta determinação inicial está relacionada com os tipos de modelos e algoritmos a utilizar, por exemplo o modelo Naive Bayes é um bom modelo para uma abordagem de classificação mas quando se usa este numa abordagem de regressão apenas faz a média dos valores.

Utilizando por base os modelos de ML citados no estado da arte, dos modelos de regressão disponíveis no pacote rminer foram escolhidos 5 para serem utilizados na modelação final sendo estes: Regressão Múltipla, KNN, Rede Neuronal (Multilayer perceptron), Random Forest e SVM.

O processo de modelação final decorreu numa máquina com o sistema operativo Linux (Xubuntu 18.04.1 LTS), equipado com um processador Intel i7-7700 e 16GB de memória RAM, 1 disco SSD de 256GB e 1 disco HDD 7200RPM de 1TB, sendo que este segundo disco foi separado em 2 partições de 500GB. Após as partições serem criadas, uma delas foi transformada numa partição swap. 16GB de RAM não são suficientes para correr a maioria dos testes realizados sendo que esta partição faz com que a memória RAM seja efetivamente tratada como se se tratasse de 516GB, embora em contrapartida torne a execução mais lenta. Testes mais pequenos foram realizados numa outra máquina com o sistema operativo Windows 10 Pro, com um processador Intel i7-7700HQ e 8GB de memória RAM, enquanto a máquina principal realizava a modelação principal. Mesmo estando a utilizar uma amostra aleatória de apenas 10% dos dados originais, os modelos são treinados e testados com uma grande quantidade de dados e é necessário automatizar o processo para que o treino, testes e métricas sejam realizadas sem necessidade de supervisão constante. Para este efeito foram criadas algumas funções que automatizam este processo, um exemplo do código utilizado pode ser visualizado na Figura 22. Alguns modelos neces-

sitaram de uma atenção especial na criação do código desenvolvido, tornando-o maior e mais confuso do que a versão genérica aqui apresentada. Alguns exemplos destas modificações são os modelos de regressão múltipla e de redes neuronais, que tiveram melhores resultados quando treinados com dados do tipo classe, sendo que o último modelo necessitou ainda de utilizar uma estrutura diferente de colunas, devido à maneira como está implementado.

```
library(rminer)
library(scorer)

auto_fit <- function(dadosTrei,dadosTes,queModelo,pathToSaveWithSlash) {
  Mdl=fit(speed~.,dadosTrei,model=queModelo) #treino do modelo
  ps=subset(dadosTes,select=-speed) #"esconde" o valor original da velocidade
  previsao=predict(Mdl,ps) #previsao da velocidade para o conjunto de teste
  metricas(dadosTes[,"speed"],previsao) #comparacao da velocidade prevista vs original
  doGraphs(previsao,dadosTes[,"speed"],queModelo)
  #gravar e limpar para executar o modelo seguinte
  assign("Mdl", Mdl, envir = .GlobalEnv) #adiciona variavel ao ambiente global
  save.image(paste0(pathToSaveWithSlash,queModelo,"_M.RData"))
  rm_metricas() #funcao extra que remove as metricas
  rm(Mdl, pos = ".GlobalEnv")
}

doGraphs <- function(pred,target,nome) { #graficos
  #target=dadosTeste, pred=previsoes, nome=nome do modelo
  png(filename=paste0(nome,"_REC"),width=600,height=600)
  mgraph(target,pred,graph="REC") #REC
  dev.off()
  png(filename=paste0(nome,"_RSC"),width=600,height=600)
  mgraph(target,pred,graph="RSC") #Scatter Plot
  dev.off()}

metricas <- function(true,pred) { #metricas
  ev<-explained_variance_score(true,pred)
  mae<-mean_absolute_error(true,pred)
  mse<-mean_squared_error(true,pred) #rmse
  mdae<-median_absolute_error(true,pred)
  rs<-r2_score(true,pred)
  #o resto do codigo adiciona variavel ao ambiente global
  assign("ev", ev, envir = .GlobalEnv)
```

Figura 22: Exemplo do código utilizado para a modelação na linguagem R.

Os pacotes utilizados no R foram o scorer para avaliação dos modelos e o rminer para o treino dos mesmos. O rminer incorpora em si vários outros pacotes que já iriam ser utilizados mas fá-lo de uma maneira a que seja possível criar uma única função para o treino de todos os modelos. Foram também utilizadas as parametrizações por omissão, pois todas as tentativas de alterar estes resultaram em modelos inferiores ou em impactos enormes na performance do treino.

Quando um modelo é gerado são gravadas as variáveis da sessão R no formato RData para uso posterior. Realça-se que este formato é uma boa opção para uso posterior, pois é mais simples de carregar através do módulo rpy2 do Python. Após isso é necessário verificar a qualidade do modelo, para tal são

utilizadas métricas padrão que serão explicadas mais à frente juntamente com os modelos utilizados. Para a execução do ambiente R foi utilizada a interface gráfica RStudio.

Modelo	Descrição	Observações
Árvores de Decisão	Utiliza um formato de árvore que funciona como sucessivas múltiplas condições. Aquando da previsão os inputs seguem o ramo ao qual melhor se aplicam sucessivamente até chegarem a um output.	Consome muita memória. O Random Forest demonstra melhores resultados e maior eficiência. Testado "ctree" e "rpart" do rminer, excluído antes da versão final.
Regressão Linear	Cria uma equação linear que relaciona a variável de previsão com as variáveis de input.	O Regressão Múltipla demonstra melhores resultados. Testado utilizando o modelo "lm" do rminer, excluído antes da versão final.
Regressão Múltipla	Cria um plano que relaciona a variável de previsão com as variáveis de input.	Testado utilizando o modelo "mr" do rminer.
K-Nearest Neighbors (KNN)	Procura os valores vizinhos de um determinado input, faz a média dos valores vizinhos mais próximos e retorna isso como resultado.	Testado utilizando o modelo "knn" do rminer.
Rede Neuronal (Multilayer perceptron)	Simula neurónios para encontrar padrões nos dados. Ao longo das várias iterações que vai fazendo os novos modelos tendem a replicar o melhor modelo da iteração anterior com pequenas mutações para encontrar a fórmula com melhores resultados.	Testado utilizando o modelo "mlp" do rminer.

Random Forest	Utiliza uma "floresta" de árvores de decisão. Os inputs passam por estas várias árvores. Os vários resultados são agrupados e aquele que for a moda do resultado, ou seja, o resultado com "mais votos" é o resultado final.	Testado utilizando o modelo "rf" do rminer. O sistema de "votação" deste modelo apresenta características dos modelos de boosting e bagging.
Support Vector Machine (SVM)	Separa os dados através de linhas/planos, ou vetores que ao colocar novos dados verifica de que valores estes estão mais próximos retornando assim o resultado mais provável.	Este modelo é o mais utilizado na indústria de acordo com a pesquisa efetuada no "Enquadramento Conceptual". Testado utilizando o modelo "svm" do rminer.

Tabela 5: Modelos utilizados neste trabalho

O tempo necessário para o treino e previsão varia com o modelo utilizado. Certos modelos fazem a maioria do seu trabalho durante o treino levando a previsões rápidas enquanto outros levam mais tempo na previsão tendo em contrapartida tempos de treino rápidos, sendo um exemplo deste segundo o KNN. O tempo de execução de certos modelos segue um padrão linear, enquanto outros seguem um padrão exponencial de acordo com a quantidade de dados que lhes é fornecido, por outras palavras assumindo um modelo que demore 1024 unidades de tempo para treinar 10 linhas, se este modelo seguir um padrão linear vai demorar o dobro do tempo para 20 linhas ou seja 2048 unidades de tempo, no entanto se for exponencial pode demorar por exemplo 1048576 unidades de tempo. Tendo em conta a quantidade de dados utilizada, modelos exponenciais tiveram de ser abordados de maneira diferente, limitando o número de exemplos de treino.

4.5 Avaliação

A avaliação de um projeto como o que está a ser desenvolvido no âmbito desta dissertação não é propriamente fácil de alcançar ou pelo menos não é simples de definir. Existem três óticas distintas sob as quais pode-se fazer a apreciação do desempenho do mesmo, do ponto de vista dos modelos podem ser

avaliadas quantitativamente as previsões realizadas. sendo este o foco desta secção. Do ponto de vista da aplicação desenvolvida podem ser qualificadas a sua visualização e tempos de resposta, sendo que o primeiro é maioritariamente subjetivo e o segundo foi otimizado para o menor tempo possível não haverá razões para discutir esta ótica em detalhe. Um último ponto de vista poderá ser a sua utilização e em que medida esta ajuda a cumprir os objetivos de negócio dos TUB, sendo esta uma ótica muito importante que será discutida mais a detalhe na Secção 4.6. Nesta secção serão então demonstrados os resultados obtidos por cada modelo para que estes possam ser avaliados e as medidas utilizadas para a verificação do modelo selecionado.

Para medir a qualidade dos modelos treinados existem duas ações muito importantes a ter em conta pois sem elas os resultados poderão ser condicionados. Em primeiro lugar os dados de treino e os dados de teste devem ser conjuntos diferentes sem sobreposições. Isto foi efetuado colocando um conjunto de dados pré concebido para o efeito, já baralhado, para que ao retirar as amostras de treino e testes destas, ainda que provenientes do mesmo conjunto para ambos os modelos, sejam em si dados novos que o modelo nunca viu e por consequência não sabe a resposta de antemão, pois seria muito fácil criar um modelo com uma precisão perfeita para o conjunto de treino sob o qual já aprendeu mas isso não se traduziria num bom modelo para prever o futuro. O segundo é utilizar as mesmas métricas para a avaliação de desempenho de todos os modelos, para uma comparação eficaz e justa dos mesmos. Assim é garantido que as únicas diferenças que podem existir entre os modelos sãos os seus próprios algoritmos e as possíveis restrições aplicadas a cada um.

De entre as várias métricas disponíveis para a avaliação de modelos de ML foram selecionadas para utilização neste projeto as seguintes:

-**Var. E.**: A variância Explicada é na sua essência a mesma métrica que o R Quadrado mas leva em conta o Erro Médio para determinar se o modelo em questão é condicionado;

-**Erro Med. Abs.**: Erro Médio Absoluto representa a média dos erros cometidos;

-**Erro Quad. Med.**: Erro Quadrático Médio representa o quadrado dos erros da previsão dividido pelo número de ocorrências dos mesmos;

-**Erro Medi. Abs.**: Erro Mediano Absoluto representa a mediana dos erros ocorridos;

-R^2: R Quadrado ou Coeficiente de Determinação representa a proporção da variância entre as variáveis ou seja mede o quão boa é a linha de regressão para as diferentes entradas;

Recorrendo a estas métricas implementas no processo de modelação utilizando o pacote "scorer" disponível para o R, foram obtidos os seguintes resultados conforme expostos na Tabela 6

Modelo	**Observações**	**Var. E.**	**Erro Med. Abs.**	**Erro Quad. Med.**	**Erro Medi. Abs.**	R^2	**Tempo de Resposta (segundos)**
Regressão Múltipla	Ocupa 1,3Gb; treinado com 10 milhões de linhas; usa geohash	4906615	11.61479	202.436	10.22792	0.002418068	1
K-Nearest Neighbors (KNN)	Ocupa 457,8 Mb; treinado com 10 milhões de linhas	213393	9.230217	143.238	7.192696	0.4990334	557
Rede Neuronal (Multilayer perceptron)	Ocupa 1,3Gb; treinado com 10 milhões de linhas; usa geohash	265846355	10.65137	176.1743	9.084123	0.1310139	2
Random Forest	Ocupa 890,2 Mb; treinado com 100 mil linhas	680408128	8.667872	122.5863	7.1116	0.3353173	3
Support Vector Machine (SVM)	Ocupa 20,2 Mb; treinado com 100 mil linhas	321856950	10.67735	177.1282	9.111392	0.1586169	21

Tabela 6: Valores de desempenho obtidos para os diferentes modelos testados.

Para uma melhor compreensão de cada modelo foram gerados gráficos de dispersão e de curva REC que podem ser vistos nas figuras:

-**Regressão Múltipla**: Figura 23 (Gráfico de Dispersão) e Figura 24 (Curva REC);

-**KNN**: Figura 25 (Gráfico de Dispersão) e Figura 26 (Curva REC);

-**Rede Neuronal**: Figura 27 (Gráfico de Dispersão) e Figura 28 (Curva REC);

-**Random Forest**: Figura 29 (Gráfico de Dispersão) e Figura 30 (Curva REC);

-**SVM**: Figura 31 (Gráfico de Dispersão) e Figura 32 (Curva REC);

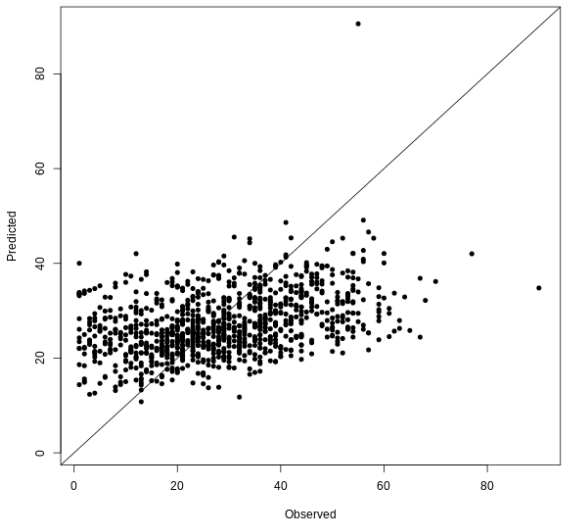


Figura 23: Gráfico de dispersão para a regressão múltipla.

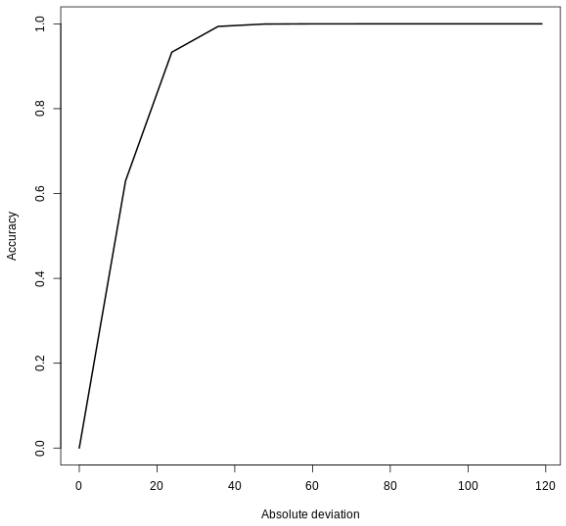


Figura 24: Curva de REC para a regressão múltipla.

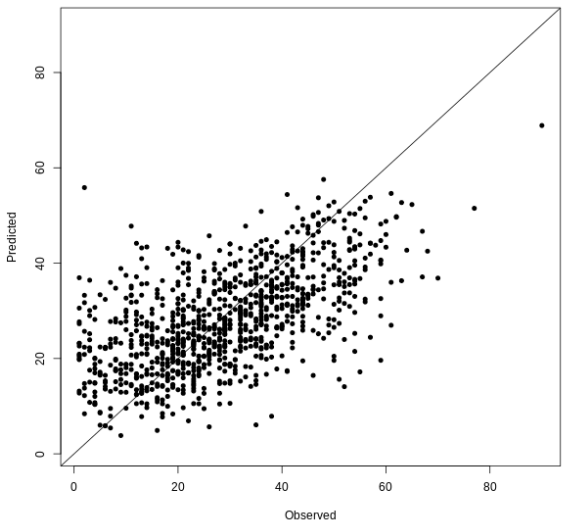


Figura 25: Gráfico de dispersão para o KNN.

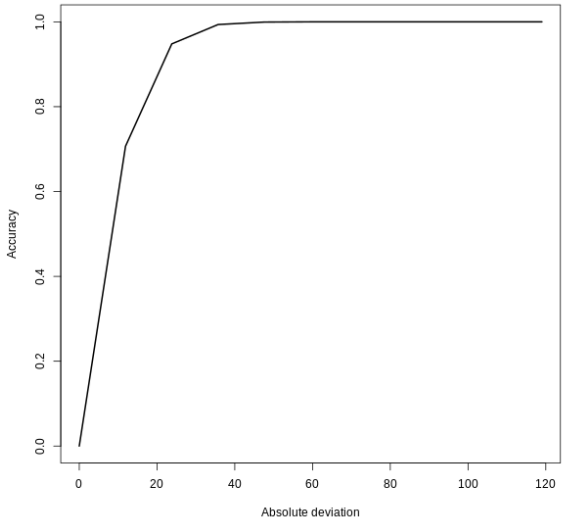


Figura 26: Curva de REC para o KNN.

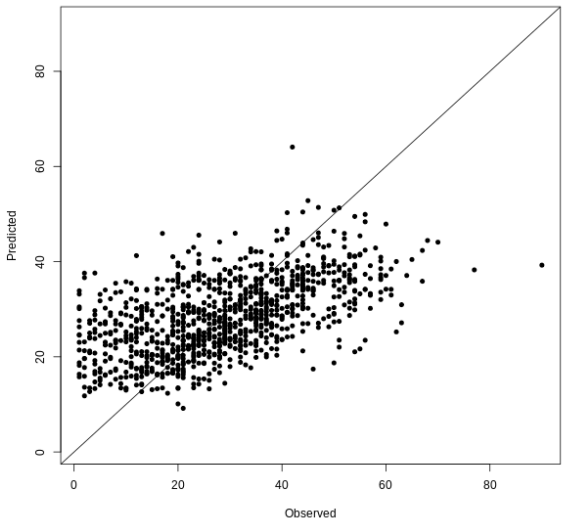


Figura 27: Gráfico de dispersão para a rede neuronal.

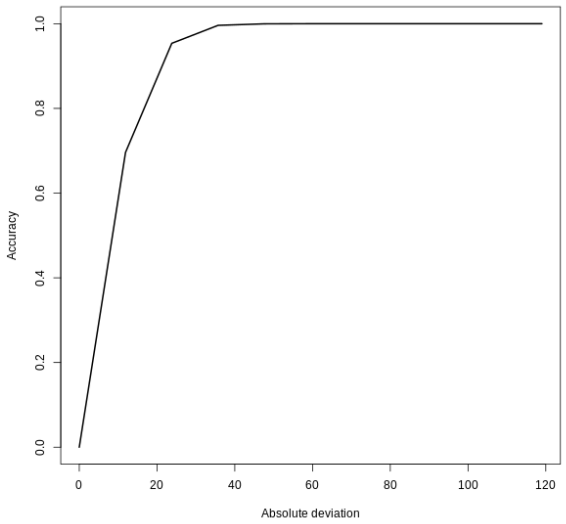


Figura 28: Curva de REC para a rede neuronal.

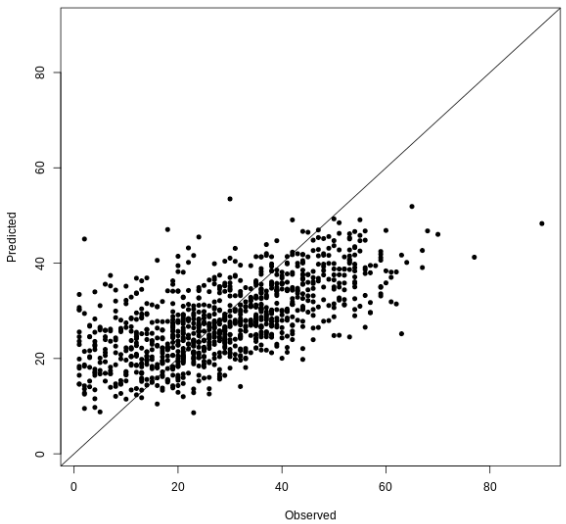


Figura 29: Gráfico de dispersão para o random forest.

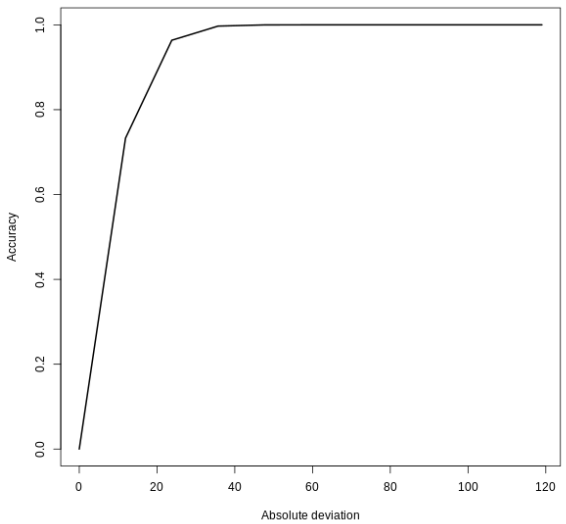


Figura 30: Curva de REC para o random forest.

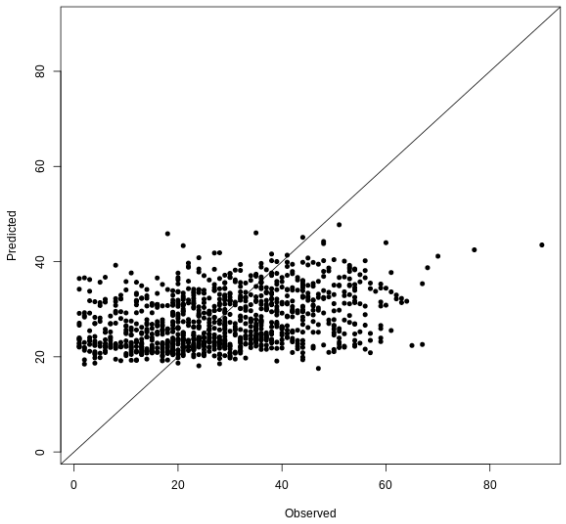


Figura 31: Gráfico de dispersão para o SVM.

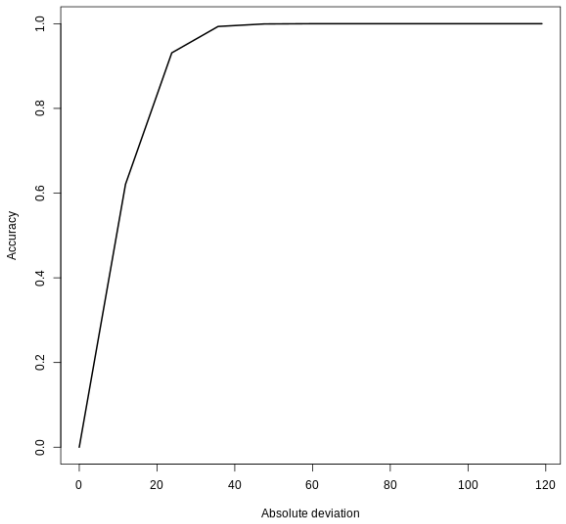


Figura 32: Curva de REC para o SVM.

Tendo em conta estes resultados foi escolhido o modelo Random Forest, sendo esta escolha discutida na Secção 4.6. Tendo o modelo selecionado, é necessário verificar se o trânsito previsto pelo mesmo corresponde à realidade num com margens de erro razoáveis. Para a verificação da integridade do modelo foi utilizado o método de holdout ordenado no tempo. O método de janela deslizante seria também uma boa opção mas requer a execução de treino múltiplas vezes e como já referido este processo demora muito tempo. O método de holdout consiste em dar apenas ao modelo acesso a uma parte do conjunto de dados, treinando-o com esses dados e fazendo-o prever a outra parte. O modelo foi treinado com dados correspondente aos três primeiros trimestres do ano pedindo que prevê-se o último. Foi anotado os valores produzidos e comparados com os reais. Observou-se que os resultados foram consistentes com o modelo anteriormente treinado (resultados na Tabela 6). Os resultados desta validação (em dados de teste) foram:

-**Var. E.**: 1110568;

-**Erro Med. Abs.**: 8.899467;

-**Erro Quad. Med.**: 128.2273;

-**Erro Medi. Abs.**: 7.353696;

-R^2: 0.281174;

Com estes resultados conclui-se que o modelo obtém previsões similares às testadas anteriormente via método aleatório de separação entre dados de treino e teste. Tendo em conta que existe uma sazonalidade neste conjunto de dados e que os meses utilizados para o treino não são muito similares aos meses

utilizados para teste principalmente no que toca a meteorologia é possível que a pequena diferença entre valores do modelo final e desta validação possa ser justificada por esse facto.

Para verificar a correspondência entre os valores previstos e os valores reais do trânsito recorreu-se à função de trânsito do Google Maps. Foram comparados num dia útil e num fim de semana às 8 horas, às 15 horas e às 22 horas o trânsito previsto e o trânsito real através dos mapas produzidos pela aplicação desenvolvida e pelo Google Maps. Foi observado que existem várias semelhanças entre os mesmos, sendo que algumas das diferenças podem ser explicadas pelas previsões do modelo serem feitas para a hora inteira e o trânsito do Google Maps ser em tempo real. Foi também observado que o Google Maps não consegue dar informações de lugares remotos onde têm pouca informação e não apresentam resultados noturnos, sendo essa uma das vantagens diferenciadoras deste trabalho.

Serão agora apresentadas imagens comparativas de uma previsão do modelo desenvolvido para uma terça-feira às 15 horas com o trânsito real sentido em Braga à mesma hora em determinadas zonas. Na Figura 33 pode-se ver a previsão completa feita pelo modelo.

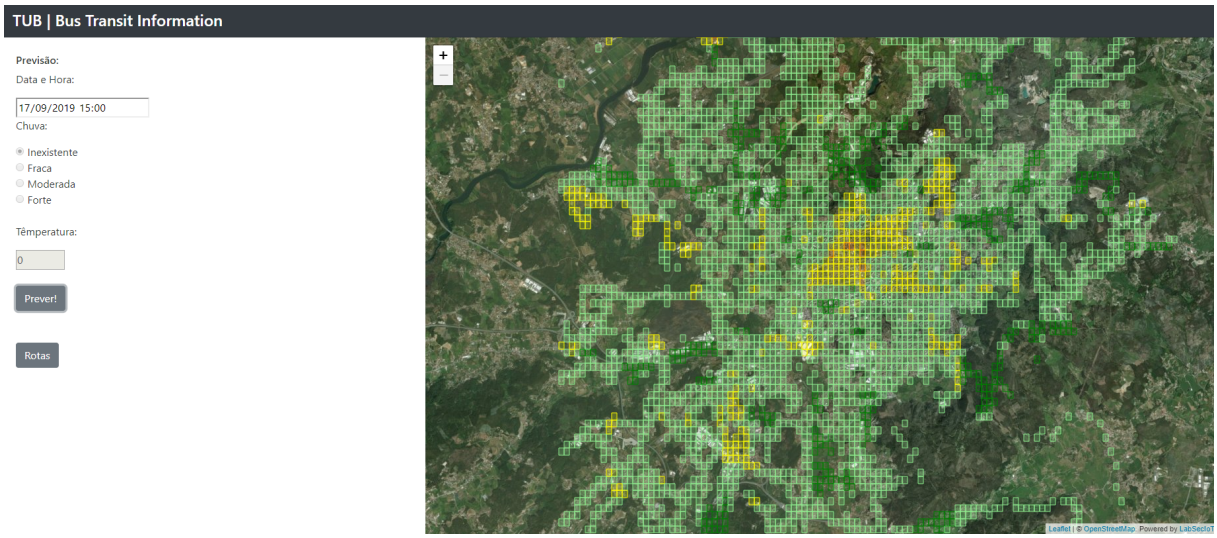


Figura 33: Heatmap de trânsito previsto para uma terça-feira às 15 horas.

Figura 34: Comparação da previsão do modelo com o Google Maps na Rua de Santo André.

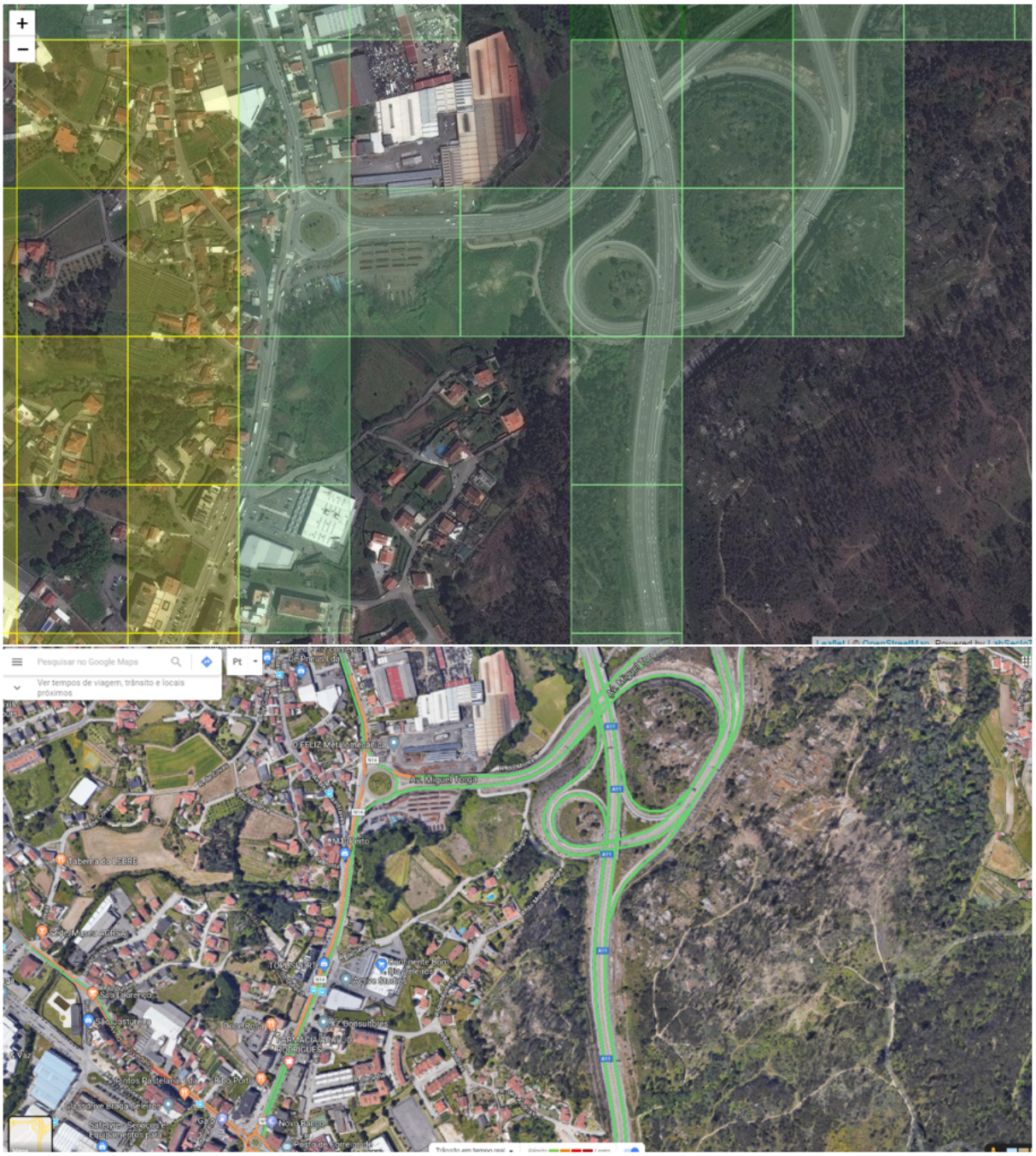


Figura 35: Comparação da previsão do modelo com o Google Maps na Avenida Miguel Torga.

Figura 36: Comparação da previsão do modelo com o Google Maps na Largo Carlos Amarante.

Figura 37: Comparação da previsão do modelo com o Google Maps na Rua de São Martinho.

Como pode ser observados nas Figuras 34, 35, 36 e 37, o modelo consegue prever com bastante fidelidade o trânsito sentido em determinada zona. As diferenças que podem ser visualizadas devem-se a três motivos chave: as previsões não se esperam ser perfeitas, o modelo faz a previsão tendo em conta apenas a hora e não os minutos e por fim podem e provavelmente existem diferenças entre a escala de cores utilizada no projeto e a escala de cores utilizada pelo Google Maps.

4.6 Discussão de Resultados

Estando realizada a modelação e avaliação dos modelos treinados para o efeito de previsão do trânsito deve-se agora discutir os resultados obtidos e constatar onde poderão ser utilizados num contexto real. O modelo de regressão múltipla demonstrou ser inferior aos outros modelos utilizados a nível de métricas. Isto era esperado tendo em conta que é também o modelo mais simples na sua implementação, no entanto tem como vantagem a sua velocidade de treino e previsão. O KNN demonstrou bons resultados na maioria das métricas mas o seu tempo de resposta resultante do esforço que coloca na previsão, aliado a inconsistências observadas tornam-no impossível de justificar como uma escolha final. A rede neuronal do tipo MultiLayer Perceptron teve resultados razoáveis e demonstrou identificar tendências associadas a cada variável, o que o torna interessante de um ponto de vista analítico mas não de um ponto de vista preditivo. O modelo de Random Forest obteve os melhores resultados em geral, mesmo tendo sido treinado com menos dados que os anteriores. O SVM demonstrou um grande potencial. Devido ao seu funcionamento, este demonstrou-se bastante interessante pois cria linhas que delimitam as áreas geográficas, podendo até ter obtido um melhor desempenho caso fosse possível treiná-lo com 10 milhões de linhas.

A seleção final do modelo Random Forest deu-se ao facto de que este é o modelo com melhores resultados. O KNN demonstrou bons resultados mas o esforço computacional preditivo deste tem que ser levado em consideração. Para um decisor dos TUB ficar à espera 10 minutos que o modelo seja executado por cada pedido efetuado, os resultados preditivos teriam que ser praticamente perfeitos, talvez acima de 90%, no ponto atual este não se distingue o suficiente do Random Forest para compensar a diferença abismal de tempo de resposta. De um ponto de vista técnico os outros modelos testados são também bastante interessantes tanto no seu funcionamento como na sua abordagem ao problema, no entanto o facto é que o Random Forest é definitivamente aquele que proporcionou os melhores resultados.

Também é relevante mencionar como referência para trabalhos futuros que a abordagem de classificação inicialmente considerada que preveria o trânsito usando variáveis como "Lento", "Moderado" e "Rápido" provou ser inferior à abordagem de regressão utilizada. Os modelos de regressão provaram-se mais eficientes na gestão de memória, mais rápidos e fornecem resultados superiores em comparação com os testados na abordagem de classificação. A utilização de geohash é também uma boa opção tanto como variável de modelação como também durante o tratamento de dados.

Tendo em conta a avaliação realizada durante a comparação com o Google Maps conclui-se que este modelo, apesar de ter valores relativamente baixos no que toca a métricas, consegue em situações reais

surpreender com resultados fiáveis que poderão ser utilizados para o apoio à decisão em contexto de negócios. Adicionalmente o acontecimento trânsito tem um fator humano associado à sua existência. Modelos de ML geralmente não se dão bem com a previsão do comportamento humano, havendo geralmente neste casos valores de métricas, principalmente de R-Score baixos. Acredita-se que este modelo reúne as condições necessárias para auxiliar um decisor nas escolhas certas para atingir os objetivos de negócio.

A informação proveniente deste modelo, juntamente com os componentes acessórios, permite dotar a cidade e a gestão da empresa de informações que viabilizam uma tomada de decisão suportada por dados do fluxo de trânsito. Assim, é possível:

- Prever e ter uma atitude pró-ativa, em vez de meramente reativa, relativos a constrangimentos de trânsito de acordo com dados históricos;
- Prever constrangimentos de acordo com dados históricos e outras variáveis, como por exemplo as condições meteorológicas;
- Otimização de rotas de acordo com o fluxo de trânsito existente;
- Criação de novas rotas de acordo com o fluxo de trânsito existente.

4.7 Componente Servidor

Após ter sido selecionado o modelo a utilizar este tem de ser incluído no servidor. O servidor será em Python e serve como base para a fase final da visualização de resultados. Sendo esta componente acessória ao trabalho principal que consiste na modelação, o seu funcionamento será agora explicado de maneira sucinta. Para a utilização do modelo gerado em R na linguagem de programação Python recorreu-se ao módulo rpy, carregando-se o ficheiro .RData em que o modelo foi gravado e utilizando um DataFrame Panda em tempo de execução, o rpy converte o DataFrame para R e repete o processo inversamente com a previsão. O servidor é bastante simples, a sua base é o módulo aiohttp, quando é iniciado carrega o modelo, adiciona a si rotas específicas para serem realizados os pedidos web para o modelo e adicionado o caminho para os ficheiros html, js e css e depois este é aberto na porta definida. Para uma melhor experiência para o utilizador, o servidor verifica se houve um evento introduzido na base de dados, verifica os dias da semana, se há feriados e até um período de 5 dias, utilizando a API da OpenWeather ele próprio verifica a temperatura e chuva, fazendo com que o utilizador apenas tenha que dizer a data e hora da previsão. Caso a data exceda 5 dias, terá de ser manualmente inserida a temperatura e chuva, o que permite ao utilizador fazer previsões num futuro longínquo, aumentando a

flexibilidade das suas decisões. Esta API poderia ser substituída por outra versão da mesma ou por outro fornecedor do mesmo tipo de serviço com um mínimo de alterações no código mas para o projeto em questão serve o propósito. Idealmente numa Smart City os valores de temperatura e pluviosidade seriam retirados em tempo real de sensores na cidade.

Quando é recebido um pedido Post com um JSON associado, o servidor verifica o tipo do pedido que está a ser realizado. Se este corresponder a um dos pedidos que está preparado para responder ele continuará a execução dos mesmos retornando o que o cliente pediu. Os pedidos de previsão estão divididos em dois tipos diferentes, os que provêm de coordenadas de latitude e longitude, e os que provêm de GeoHash. Ambos funcionam exatamente da mesma maneira sendo a única diferença que no caso da GeoHash esta tem que ser convertida para latitude e longitude para a previsão e voltar a ser codificada para dar a resposta. Como essencialmente funcionam da mesma maneira será apenas explicado o método que faz a previsão, este método é chamado de returnFuture e recebe como input as variáveis latitude, longitude, mês, hora, classe de minutos (para possível uso em futuros projetos), temperatura, chuva, ano e dia e executa o método "predict" do classificador sobre o retorno de uma outra função chamada makeAnX, assim chamada pois está a fornecer um X para prever um y. Esta função converte as variáveis recebidas num DataFrame do pandas, na ordem correta e acrescenta também os dados em falta. Apesar do método requerer temperatura e chuva estas não são necessárias, o que acontece é que o cliente enviará estas variáveis com o valor "ng" que significa not given ou seja não dado, ao ler esta String a função substitui estes valor pelos fornecidos pelo OpenWeather. As Figuras 38, 39, 40, 41 e 42 referem-se às partes principais do código utilizado no servidor.

```
######################################## OS VARIABLES (CHANGE FROM PC TO PC) ##########################################################################
import os
os.environ['R_HOME'] =
os.environ['R_USER'] =
######################################## ACTUAL CODE ##################################################################################
import rpy2.robjects as robjects
from rpy2.robjects.packages import importr
import asyncio
from aiohttp import web
import aiohttp
import pandas as pd
from rpy2.robjects import r, pandas2ri
import datetime
import holidays
import pyowm
import json
import pymongo
import pygeohash as pgh
import geohash
import pickle

ghbraga = ["ez6h2z6","ez6h25d","ez6h2tv","ez6h24e","ez6h27y","ez6h3pn","ez3uqb8","ez3urwx","ez6h85y","ez3urqm","ez6h2rb","ez3uptq","ez3urnf","ez6h275","ez6h2hn",
```

Figura 38: Código do servidor que inclui os pacotes utilizados.

```
def returnFutureOptimized(data,routes): #funcao principal
    dots = []
    hashes = []
    zzz = []
    for r in routes:
        lat=r[0]
        lon=r[1]
        month=data['month']
        hour=data['hour']
        minuteclass=data['minuteclass']
        year=data['year']
        day=data['day']
        temp=data['temp']
        rain=data['rain']
        feriado = False
        d = datetime.datetime(year,month,day)
        if d in feriados:
            feriado = True
        dayofweek = whatday(d.weekday())
        if temp=="ng" or rain=="ng":
            tr = getTempAndRain(year,month,day,hour)
            temp = tr['temp']
            rain = tr['rain']
        else:
            temp = int(temp)
            rain = int(rain)
        event = False
        for x in events:
            if year == int(x['year']) and month == int(x['month']) and int(day == x['day']):
                if hour >= int(x['hourS']) and hour < int(x['hourE']):
                    if pgh.encode(lat, lon) == x['area']:
                        event = True
        if event:
            event = 1
        else:
            event = 0
```

Figura 39: Código do servidor para criar o objeto para o qual se pretende a previsão (parte 1).

```
        if feriado:
            feriado = 1
        else:
            feriado = 0
        gh = geohash.encode(lat,lon,precision=7)
        a = int(gh[2:3])
        b = gh[3:4]
        c = gh[4:5]
        d = gh[5:6]
        e = gh[6:7]
        dots.append([lat,lon,month,hour,dayofweek,feriado,temp,rain,event])
        zzz.append([lat,lon])
    df = pd.DataFrame(dots , columns = ['lat','lon','month','hour','dayofweek','feriado','temp','rain','event'])
    r_dataframe = pandas2ri.py2ri(df)
    pred = rpred(rmdl,r_dataframe)
    predjson = {}
    ii = 0
    for zz in zzz:
            j = {'lat':zz[0],'lon':zz[1],'predSpeed':pred[ii]}
            predjson[ii] = j
            ii = ii + 1
    return predjson
```

Figura 40: Código do servidor para criar o objeto para o qual se pretende a previsão (parte 2).

```
#requests
routes=web.RouteTableDef()
@routes.post('/predict')
async def hand(request):
    data = await request.json()
    if data['type'] == 'predict':
        pred = returnFutureOptimized(data,data['route'])
        return web.json_response({'routes':pred})
    elif data['type'] == 'box':
        boxes = {}
        i = 0
        for g in ghbraga:
            temporary = geohash.bbox(g)
            temporary['hash'] = g
            boxes[i] = temporary
            i = i + 1
        return web.json_response({'box':boxes})
    elif data['type'] == 'predicth':
        routes = []
        for g in ghbraga:
            m = geohash.decode(g)
            routes.append([m[0],m[1],g])
        pred = returnFutureOptimized(data,routes)
        return web.json_response({'routes':pred})
```

Figura 41: Código do servidor para responder a pedidos.

```
#start server
if __name__ == '__main__':
    pandas2ri.activate()
    r('memory.limit(size=    )')
    myclient = pymongo.MongoClient('                                    )
    mydb = myclient["      "]
    mycol = mydb["eventos"]
    mydoc = mycol.find({})
    events = []
    for x in mydoc:
        events.append(x)
    feriados = holidays.CountryHoliday('PTE')
    owm = pyowm.OWM('                                    ')
    weather=getWInfo()
    robjects.r['load']("                                  ")
    importr("rminer")
    rpred = robjects.r['predict']
    rmdl = robjects.r['mdl']
    app=web.Application()
    app.add_routes(routes)
    app.router.add_static('/static', path=str( ./static / ))
    web.run_app(app,port=    )
```

Figura 42: Código do servidor para iniciar a execução.

4.8 Componente de Visualização

A componente de visualização foi construída utilizando linguagens de programação web. Esta será detalhada maioritariamente de acordo com a perspetiva de um utilizador devido à simplicidade do seu código. Do lado do cliente tem-se um simples website com apenas duas páginas. Ambas têm a mesma base, para CSS foram utilizadas classes já existente no website do laboratório, de modo a criar consistência a nível visual caso esta componente seja permanentemente adicionada ao Edge ou de alguma maneira reaproveitada para um serviço semelhante no futuro. Do lado direito de ambas as páginas existe um mapa criado em Leaflet utilizando o OpenMaps. É neste mapa que toda a informação de previsão acontece. Do lado esquerdo em ambas as páginas tem-se uma caixa para colocar a data e hora da previsão, assim como um conjunto de radio buttons para colocar informações relativas à chuva e uma caixa numérica para a temperatura, abaixo disso existe um botão "Prever" que executa a função que envia o pedido ao servidor. Existe também um botão para navegar entre as duas páginas.

A diferença entre as páginas consiste no facto de que uma delas contêm a previsão por rotas e a outra contém um heatmap baseado em geohash para apresentar as previsões. Ambas usam a mesma gama de cores. A página de rotas contém um seletor de rotas do lado esquerdo também, e quando existe trânsito numa zona da rota a linha fica mais grossa para chamar à atenção daquela zona. Esta página obtém as rotas através de um pedido ao Edge que implementa uma base de dados das rotas e viagens obtidas através dos ficheiros GTFS dos próprios TUB. O geohash está codificado no próprio servidor, pois seria um desperdício de recursos criar este em tempo de execução tendo em conta que a cidade onde os TUB operam é Braga e a cidade não muda a um ritmo rápido o suficiente para justificar outra opção. Nas Figuras 43 e 44 podem ser observados os componentes de visualização por rotas e por heatmap respetivamente.

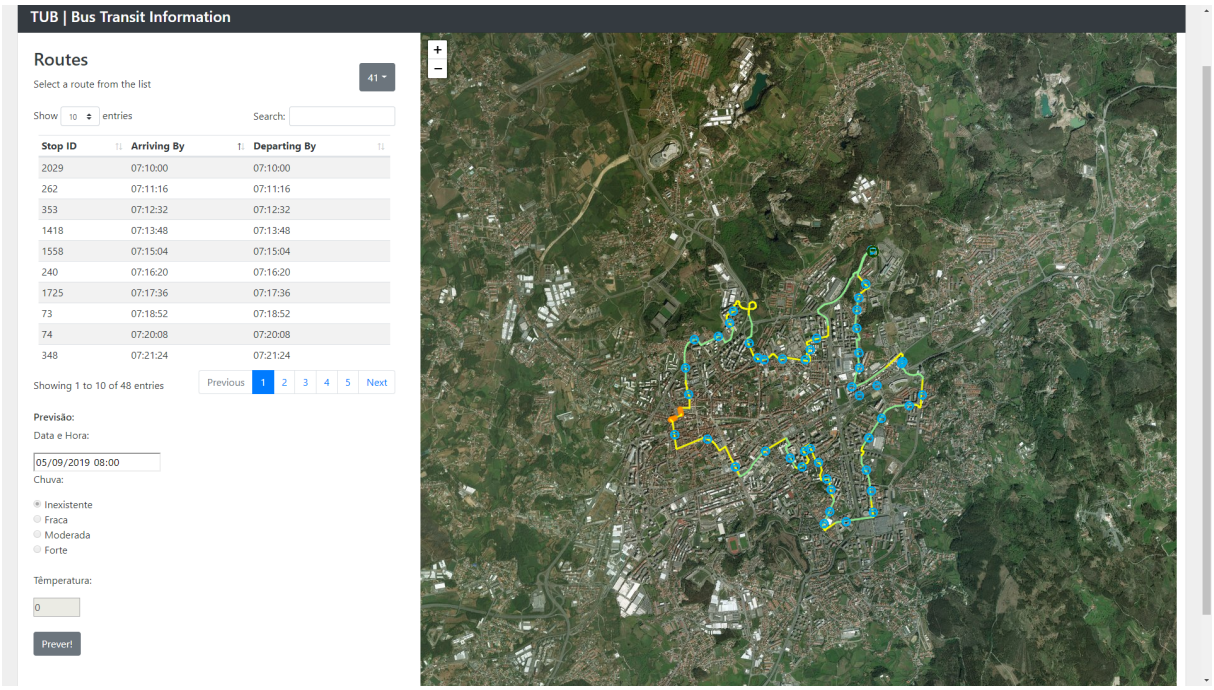


Figura 43: Visualização por rotas.

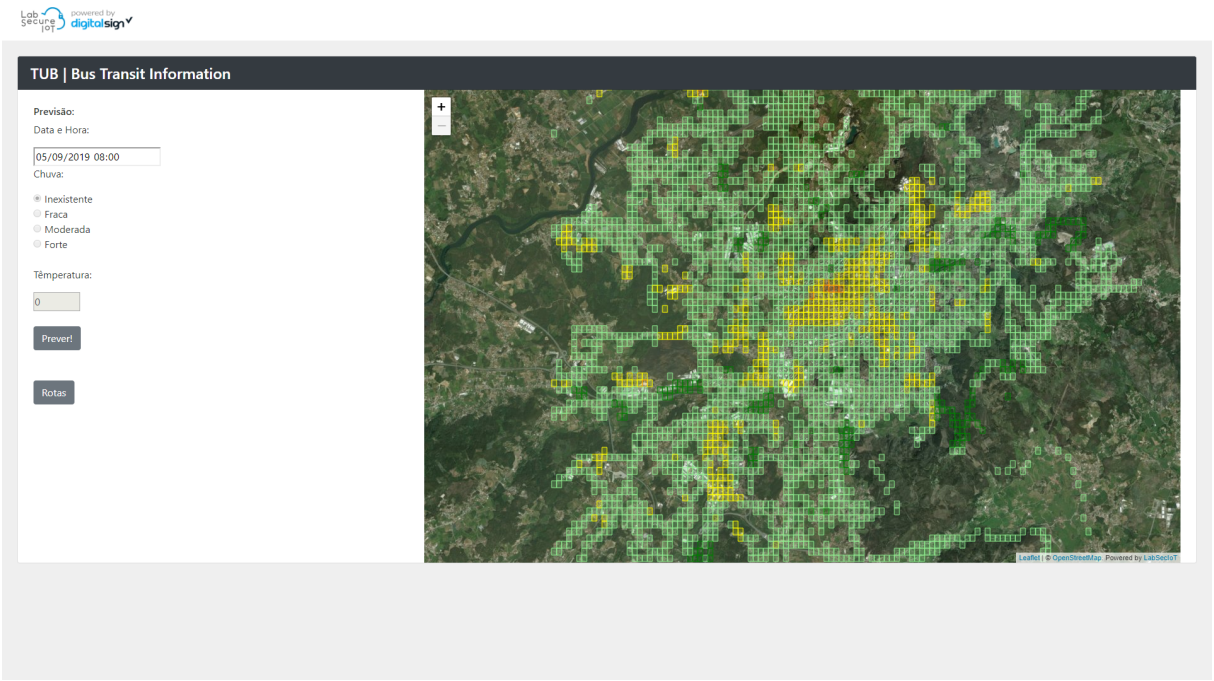


Figura 44: Visualização por mapa de calor.

5

Conclusão

Neste capítulo será apresentada uma síntese do trabalho realizado, serão discutidas as limitações em que este trabalho incorreu e serão também apresentadas indicações para o trabalho a ser realizado no futuro.

5.1 Síntese do Trabalho Efetuado

Neste projeto de dissertação foi proposto que recorrendo a técnicas de DM fosse treinado um modelo utilizando dados fornecidos pelos TUB e outros dados complementares, com o objetivo de prever o trânsito na cidade de Braga, fornecendo assim informação para auxiliar a tomada de decisões. Para a realização deste projeto foram definidos os enquadramentos e motivações, os objetivos do mesmo e que abordagens e estratégias seriam utilizadas no desenvolvimento da mesma. A seguir foi feito um levantamento do estado da arte, entre conceitos relevantes e projetos das áreas de previsão e trânsito e obtidas informações essenciais para ajudar na realização desta dissertação. Foram definidas as tecnologias e linguagens a utilizar e feito um estudo inicial dos dados existentes e necessários de obter.

Utilizando o conhecimento obtido nas fases anteriores procedeu-se com o desenvolvimento da componente prática do projeto em questão. Iniciou-se pela compreensão do negócio em estudo que forneceu uma perspetiva sobre o funcionamento da área que se está a tentar modelar e prosseguiu-se pela compreensão dos dados reunidos, tarefa facilitada pelo estudo prévio dos mesmos e definiu-se, que informação seria utilizada para a modelação. Após isso seguiu-se o tratamento dos dados convertendo os seus formatos para versões mais indicadas para o trabalho e foram associados os dados complementares, assim como foram também criados novos dados provenientes do cruzamento dos dados originais.

Tendo os dados prontos para utilização estes foram importados para um ambiente R onde foram fornecidos a diversos algoritmos padrão na indústria para treino e teste, gerando modelos capazes de

realizar a previsão da velocidade nas várias zonas da cidade de Braga e retirou-se várias métricas. Cada modelo foi avaliado individualmente e comparado com os outros, tendo-se concluído que o modelo que apresentou os melhores resultados preditivos foi o Random Forest. Este modelo foi testado com uma validação adicional, onde se preservou a ordem temporal, sendo por isso treinado com dados mais antigos e testado em dados mais recentes. Esta validação adicional permitiu aferir que o modelo mantém uma capacidade preditiva similar à obtida na experimentação inicial de divisão aleatória de exemplos em treino e teste.

Ao longo do trabalho foram ainda desenvolvidos um servidor e uma página de visualização, que foram utilizados para tornar um modelo baseado em previsões numéricas numa ferramenta de apoio à decisão fácil de utilizar. Por fim, discutiu-se a potencial utilidade dos modelos preditivos desenvolvidos para a empresa TUB. Espera-se também que este trabalho possa servir de base para pesquisas futuras na área do Smart Mobility, sendo este um tópico que se torna a cada dia mais relevante.

No âmbito deste projeto foi submetido um artigo para a convenção "WorldCIST" intitulado de "Traffic Flow Prediction using Public Transport and Weather Data: A Medium Sized City Case Study", que aborda as temáticas discutidas neste projeto de dissertação.

5.2 Limitações

Ao longo do decorrer deste projeto de dissertação foram encontradas várias limitações que impediram certos caminhos de ser tomados. A maioria destas limitações foi descoberta numa fase muito inicial do projeto, ainda durante a preparação do mesmo, mas outras apenas foram aparecendo durante fases mais finais do mesmo. Logo de início foi rapidamente descoberto que o projeto não poderia avançar com tantos dados como inicialmente fora antecipado. Apesar de existirem muitos dados abertos, falta um arquivo no que toca a informações como estradas cortadas e acidentes, e os dados que existem sobre outras informações importantes por muitas vezes encontram-se em formatos como PDF, maioritariamente como imagens que não se destinam a leituras "simples" por parte de computadores. Os dados dos TUB têm também a particularidade de serem relatados da perspetiva de um autocarro podendo haver certas generalizações que não se traduzam na perfeição para o trânsito esperado, o que pode ser visto como uma limitação ou como uma especificidade do negócio. Estes dados também utilizaram um identificador de rotas não normalizado que causou opções de trabalho a serem fechadas. Desta forma pode-se dizer que houve limitações nos dados que foram possíveis de serem obtidos.

Por outro lado foram encontradas limitações relacionadas com o poder computacional disponível para lidar com um projeto desta complexidade. Com apenas 16 GB de RAM, não foi possível desenvolver todas as vertentes deste projeto utilizando esta memória rápida tendo que se optar pela utilização de memória mais lenta na forma de memória swap. Sendo estas máquinas limitadas por vários aspetos físicos, os tempos de processamento não foram tão bons quanto poderiam ter sido dado uma máquina com um poder computacional superior e a ausência de um UPS (Uninterruptible Power Supply) colocou vários processos em risco de terminarem abruptamente, o que aconteceu em algumas instâncias executadas, sendo a penalização destas o reinicio do processo que poderia já estar em estágios finais. Houve desta forma limitações causada pelo hardware disponível.

A nível de software existiram também limitações na forma de programas que na sua versão gratuita não permitiam a utilização de grandes quantidades de dados. Por exemplo, no caso peculiar do módulo SciToolKit para Python, a partir de um determinado ponto este deixou de produzir modelos que realizassem previsões diferentes. Foram utilizados vários métodos de debugging e testada a mesma script utilizada noutras máquinas, as quais correram-na sem problemas o que indica um problema particular na instalação do módulo na máquina em questão, levando ao abandonamento desta ferramenta por inteiro tendo em conta que existia uma excelente alternativa na forma do R. Por último existiu outra limitação sendo esta o tempo disponível para a realização desta dissertação.

5.3 Perspetivas para Trabalho Futuro

Durante a execução deste projeto de dissertação foram exploradas diversas opções para a realização deste, muitas das quais foram descartadas por motivos de tempo. Seria de interesse em projetos futuros voltarem a olhar para estas escolhas e explorar se seria possível obter mais valor dos caminhos previamente descartados. Serão deixadas nesta secção referências a possíveis caminhos a realizar.

Uma das primeiras decisões tomadas neste projeto foi descartar a análise do trânsito por rotas, pois seria dispendioso conseguir compreender o seu significado utilizando os recursos disponíveis publicamente sobre os seus códigos. A análise por rotas poderia fornecer uma nova perspetiva e seria interessante criar cenários de experiência com estes dados, podendo trazer um valor acrescentado no poder decisivo dos TUB. Tendo em conta a perspetiva de trabalho futuro seria também importante a utilização de mais dados, caso disponíveis numa data posterior, como por exemplo o de cortes e obras nas estradas. De facto, o futuro para o qual as cidades estão a mover-se é um futuro Smart City, assim existem grandes possibilidades de disponibilização de novas fontes para uso num projeto como o apresentado nesta disser-

tação. Por fim, seria interessante utilizar mais dados para o treino dos modelos que ganhariam com isto possivelmente um maior valor preditivo. Num possível futuro, em que existam mais veículos autónomos a circular nas estradas pode também ser feita uma reavaliação dos modelos, pois sem a componente humana deverá existir uma maior facilidade na previsão de trânsito.

Tendo sido selecionados os cinco modelos testados nesta dissertação não houve muito espaço para a experimentação com outros modelos diferentes, variantes dos já existentes e restrições mais otimizadas. Um projeto futuro poderia revelar novos algoritmos capazes de previsões mais precisas e rápidas. Existirá sempre espaço para uma nova abordagem ao problema sendo que espera-se que esta possa ser ainda melhor do que a apresentada neste projeto de dissertação.

rpy2. URL https://pypi.org/project/rpy2/.

Imtiaz Adam. Kdnuggets, 2018. URL https://www.kdnuggets.com/2018/11/an-introduction-ai.html.

Administrator. Next, 2018. URL http://its.armis.pt/produtos/next/.

A. Agarwal, A. Aggarwal, and A. Agarwal. An approach for augmenting selection operators of sql queries using skyline and fuzzy-logic operators. In Procedia Computer Science, volume 115, pages 14–21, 2017. Cited By :1.

R. Agrawal and R. Srikant. Privacy-preserving data mining. SIGMOD Record (ACM Special Interest Group on Management of Data), 29(2):439–450, 2000. Cited By :1855.

Manuel Alfonseca. ¿ basta la prueba de turing para definir la "inteligencia artificial"? Scientia et Fides, 2 (2):129–134, 2014.

A. Azevedo and M. F. Santos. Kdd, semma and crisp-dm: A parallel overview. In MCCSIS'08 - IADIS Multi Conference on Computer Science and Information Systems; Proceedings of Informatics 2008 and Data Mining 2008, pages 182–185, 2008. Cited By :85.

M. R. Berthold. From patterns to discoveries, volume 9783642280474 of Journeys to Data Mining: Experiences from 15 Renowned Researchers, pages 43–49. 2012.

A. Bezuglov and G. Comert. Short-term freeway traffic parameter prediction: Application of grey system theory models. Expert Systems with Applications, 62:284–292, 2016. Cited By :29.

Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. Neurocomputing, 300:17–33, 2018.

A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. Science, 356(6334):183–186, 2017. Cited By :81.

Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. Smart cities in europe. Journal of urban technology, 18(2):65–82, 2011.

Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. Crisp-dm 1.0 step-by-step data mining guide. 2000.

M. Chen, S. Mao, and Y. Liu. Big data: A survey. Mobile Networks and Applications, 19(2):171–209, 2014. Cited By :876.

K. . Chu, Y. . Huang, C. . Tseng, H. . Huang, C. . Wang, and H. . Tai. Reliability and validity of ds-adhd: A decision support system on attention deficit hyperactivity disorders. Computer methods and programs in biomedicine, 140:241–248, 2017.

Byung-Tae Chun and Seong-Hoon Lee. Review on its in smart city. Advanced Science and Technology Letters, 98:52–54, 2015.

Y. Cong, J. Wang, and X. Li. Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm. In Procedia Engineering, volume 137, pages 59–68, 2016. Cited By :26.

CRAN. Contributed packages. URL https://cran.r-project.org/web/packages/.

J. G. Davis and D. Sundaram. Petaps: A prototype decision support system for consumer product marketing and promotion. European Journal of Operational Research, 87(2):247–256, 1995. Cited By :1.

J. Demšar, T. Curk, A. Erjavec, C. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan. Orange: Data mining toolbox in python. Journal of Machine Learning Research, 14:2349–2353, 2013. Cited By :271.

M. S. Dougherty and M. R. Cobbett. Short-term inter-urban traffic forecasts using neural networks. International Journal of Forecasting, 13(1):21–31, 1997. Cited By :181.

B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline. Machine learning for medical imaging. Radiographics, 37(2):505–515, 2017. Cited By :48.

Ericlicoding. How to choose algorithms for azure machine learning studio, 2018. URL https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice.

Alex Fabrikant. Predicting bus delays with machine learning, Jun 2019. URL https://ai.googleblog.com/2019/06/predicting-bus-delays-with-machine.html.

Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2):654–669, 2018.

G. Fusco, C. Colombaroni, and N. Isaenko. Short-term speed predictions exploiting big data on large urban road networks. Transportation Research Part C: Emerging Technologies, 73:183–201, 2016. Cited By :24.

A. Gandomi and M. Haider. Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management, 35(2):137–144, 2015. Cited By :617.

Chiara Garau, Francesca Masala, and Francesco Pinna. Cagliari and smart urban mobility: Analysis and comparison. Cities, 56:35–46, 2016.

Group Gartner. What is big data? - gartner it glossary - big data, Dec 2016. URL https://www.gartner.com/it-glossary/big-data/.

C. Goves, R. North, R. Johnston, and G. Fletcher. Short term traffic prediction on the uk motorway network using neural networks. In Transportation Research Procedia, volume 13, pages 184–195, 2016. Cited By :13.

N. W. Grady. Kdd meets big data. In Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016, pages 1603–1608, 2016. URL www.scopus.com. Cited By :5.

J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7):1645–1660, 2013. Cited By :3161.

I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47:98–115, 2015. Cited By :654.

Y. T. IÇ and M. Yurdakul. Development of a quick credibility scoring decision support system using fuzzy topsis. Expert Systems with Applications, 37(1):567–574, 2010. Cited By :43.

M. Kantardzic. Data Mining: Concepts, Models, Methods, and Algorithms: Second Edition. Data Mining: Concepts, Models, Methods, and Algorithms: Second Edition. 2011. Cited By :343.

James Le. The 10 algorithms machine learning engineers need to know (kdnuggets), Aug 2016. URL `https://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html`.

H. Liu, R. He, K. Zhang, and J. Li. A neural network model for travel time prediction. In Proceedings - 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009, volume 1, pages 752–756, 2009. Cited By :6.

X. Ma, H. Yu, Y. Wang, and Y. Wang. Large-scale transportation network congestion ,evolution prediction using deep learning theory. PLoS ONE, 10(3), 2015. Cited By :157.

Marvin Minsky. Steps toward artificial intelligence. Proceedings of the IRE, 49(1):8–30, 1961.

Mazin Abed Mohammed, Mohd Khanapi Abd Ghani, N Arunkumar, Raed Ibraheem Hamed, Mohamad Khir Abdullah, and MA Burhanuddin. A real time computer aided object detection of nasopharyngeal carcinoma using genetic algorithm and artificial neural network based on haar feature fear. Future Generation Computer Systems, 89:539–547, 2018.

S. Moro, R. M. S. Laureano, and P. Cortez. Using data mining for bank direct marketing: An application of the crisp-dm methodology. In ESM 2011 - 2011 European Simulation and Modelling Conference: Modelling and Simulation 2011, pages 117–121, 2011. Cited By :62.

A. Nantes, D. Ngoduy, A. Bhaskar, M. Miska, and E. Chung. Real-time traffic state estimation in urban corridors from heterogeneous data. Transportation Research Part C: Emerging Technologies, 66:99–118, 2016. Cited By :31.

Elsa Negre, Camille Rosenthal-Sabroux, and Mila Gascó. Introduction to smart cities and smart city government minitrack. In System Sciences (HICSS), 2015 48th Hawaii International Conference on, pages 2316–2316. IEEE, 2015.

Zhaolong Ning, Feng Xia, Noor Ullah, Xiangjie Kong, and Xiping Hu. Vehicular social networks: Enabling smart mobility. IEEE Communications Magazine, 55(5):16–55, 2017.

T. E. Oliphant. Python for scientific computing. Computing in Science and Engineering, 9(3):10–20, 2007. Cited By :1071.

Jeff Z Pan. Closing some doors for the open semantic web. In Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, page 2. ACM, 2012.

Kasey Panetta. 5 trends emerge in the gartner hype cycle for emerging technologies, 2018, 2018. URL `https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/`.

A. Pashkevich, K. Shubenkova, I. Makarova, and D. Sabirzyanov. Decision Support System to Improve Delivery of Large and Heavy Goods by Road Transport, volume 844 of Advances in Intelligent Systems and Computing. 2019.

N. G. Polson and V. O. Sokolov. Deep learning for short-term traffic flow prediction. Transportation Research Part C: Emerging Technologies, 79:1–17, 2017. Cited By :66.

R. Porter, D. Hush, N. Harvey, and J. Theiler. Toward interactive search in remote sensing imagery. In Proceedings of SPIE - The International Society for Optical Engineering, volume 7709, 2010. Cited By :4.

PyPI. Pypi – the python package index. URL `https://pypi.org/`.

Google reCAPTCHA. recaptcha, 2018. URL `https://www.google.com/recaptcha/intro/v3.html`.

S. H. Rubin, J. Boerke, and R. J. Rush Jr. The intelligent web is coming. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, volume 4, page 2552, 2001. Cited By :1.

A. Sankar, P. D. Bharathi, M. Midhun, K. Vijay, and T. S. Kumar. A conjectural study on machine learning algorithms, volume 397 of Advances in Intelligent Systems and Computing. 2016. Cited By :6.

J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun. A lightweight multi-layer authentication protocol for wireless body area networks. Future Generation Computer Systems, 78:956–963, 2018. Cited By :49.

S. Shirgaonkar, S. Rathi, and T. Rajkumar. Overview of real time decision support system. In ICWET 2010 - International Conference and Workshop on Emerging Trends in Technology 2010, Conference Proceedings, pages 179–181, 2010. Cited By :1.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. Nature, 550(7676):354, 2017.

P. Smyth. Data mining: Data analysis on a grand scale? Statistical methods in medical research, 9(4): 309–327, 2000. Cited By :31.

C. Stergiou, K. E. Psannis, B. . Kim, and B. Gupta. Secure integration of iot and cloud computing. Future Generation Computer Systems, 78:964–975, 2018. Cited By :62.

X. K. Sun. The development and research of data mining technology, volume 602-605 of Applied Mechanics and Materials. 2014.

R. Tkachenko, A. Doroshenko, I. Izonin, Y. Tsymbal, and B. Havrysh. Imbalance data classification via neural-like structures of geometric transformations model: Local and global approaches, volume 754 of Advances in Intelligent Systems and Computing. 2019. Cited By :7.

Google Trends, 2018. URL `https://trends.google.pt/trends/`.

J. S. Van Der Veen, B. Van Der Waaij, and R. J. Meijer. Sensor data storage performance: Sql or nosql, physical or virtual. In Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, CLOUD 2012, pages 431–438, 2012. Cited By :84.

K. M. van Hee, L. J. Somers, and M. Voorhoeve. A modeling environment for decision support systems. Decision Support Systems, 7(3):241–251, 1991. Cited By :9.

J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou. Mobile crowd sensing for traffic prediction in internet of vehicles. Sensors (Switzerland), 16(1), 2016. Cited By :56.

Kevin Warwick and Huma Shah. Can machines think? a report on turing test experiments at the royal society. Journal of experimental & Theoretical artificial Intelligence, 28(6):989–1007, 2016.

X. Wu, X. Zhu, G. . Wu, and W. Ding. Data mining with big data. IEEE Transactions on Knowledge and Data Engineering, 26(1):97–107, 2014. Cited By :976.

Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. IEEE Internet of Things journal, 1(1):22–32, 2014.

P. Zaraté. The process of designing a dss: A case study in planning management. European Journal of Operational Research, 55(3):394–402, 1991. Cited By :7.

Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu. Lstm network: A deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems, 11(2):68–75, 2017. Cited By :137.

M. Zhou, X. Qu, and X. Li. A recurrent neural network based microscopic car following model to predict traffic oscillation. Transportation Research Part C: Emerging Technologies, 84:245–264, 2017. Cited By :18.