Utilização de Data Mining e Deep Learning para Business Intelligence em estrutura integrada de sistema Smart Parking

Lucas Ribeiro Mendes

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Dr. Paulo Alexandre Vara Alves

Prof. Dr. André Pinz Borges

Prof. Dr. Paulo Jorge Pinto Leitão

Prof. Dr. Gleifer Vaz Alves

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2020-2021

Utilização de Data Mining e Deep Learning para Business Intelligence em estrutura integrada de sistema Smart Parking

Lucas Ribeiro Mendes

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Dr. Paulo Alexandre Vara Alves

Prof. Dr. André Pinz Borges

Prof. Dr. Paulo Jorge Pinto Leitão

Prof. Dr. Gleifer Vaz Alves

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2020-2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa

Diretoria de Graduação e Educação Profissional

TERMO DE APROVAÇÃO

Utilização de Data Mining e Deep Learning para Business Intelligence em estrutura integrada de sistema Smart Parking

por

LUCAS RIBEIRO MENDES

Este Trabalho de Conclusão de Curso (TCC) ou esta Monografia ou esta Dissertação foi apresentado(a) em 22 de fevereiro de 2021 como requisito parcial para a obtenção do título de Mestre em Informática. O(a) candidato(a) foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Paulo Alexandre Vara Alves Prof. Orientador IPB	José Eduardo Moreira Fernande Membro titular
André Pinz Borges Prof. Orientador UTFPR	Pedro João Soares Rodrigues Membro titular
Geraldo Ranthum Responsável pelos Trabalhos de Conclusão de Curso	Mauren Louise Sguario Coelho de Andrade Coordenador do Curso UTFPR - Campus Ponta Grossa

Dedicatória

Dedico este trabalho primeiramente aos meus pais Valter Alves Mendes e Cristhiane Ribeiro Mendes que sempre me apoiaram e incentivaram meus estudos e decisões. Também ao meu irmão Johnatan Ribeiro Mendes pelo convívio e auxílio, conversas e apoio que sempre me forneceu.

Agradeço também à minha companheira Anelize Cristina da Luz, que me acompanhou durante todo esse processo, fornecendo apoio em toda minha trajetória e estando do meu lado em todos os momentos.

Agradecimentos

Agradeço aos Professores Dr. Paulo Alves e Dr. André Pinz, que auxiliaram com seu conhecimento durante todo o processo de desenvolvimento e elaboração da dissertação, guiando-me e contribuindo em todas as decisões.

Aos professores Dr. Paulo Leitão e Dr. Gleifer Vaz que me introduziram no projeto do Smart Parking, auxiliando no conhecimento da estrutura e compreensão dos problemas relativos ao projeto, além de acompanharem todo o processo de desenvolvimento.

Agradeço à Universidade Tecnológica Federal do Paraná e ao Instituto Politécnico de Bragança por proporcionarem o programa de dupla diplomação e fornecerem toda a estrutura necessária para minha formação.

Resumo

O avanço tecnológico e o crescimento populacional dos últimos anos trouxe uma alta demanda por soluções inteligentes que pudessem melhorar a qualidade de vida da população. Uma dessas soluções é o Smart Parking (estacionamentos inteligentes). Esse conceito integra diferentes áreas e tem por objetivo reduzir o fluxo de trânsito de cidades por meio da implementação de sistemas inteligentes, focados no controle e gestão de estacionamentos. O presente trabalho integrou o desenvolvimento de um modelo de Smart Parking já estruturado, o qual foi concebido de forma gradual por alunos e professores da UTFPR e IPB. Propôs-se a criação de uma estrutura de dados que integrasse todos os módulos do sistema. Além disso, foi proposto um sistema que pudesse auxiliar na tomada de decisões do produto, utilizando como base o grande volume de dados gerados por esse tipo de aplicação. Com isso, no decorrer do trabalho é apresentado o modelo conceitual utilizado na integração dos módulos, seguido de etapas de mineração e análise de dados. Também é abordada a criação de um modelo para simulação de dados e a implementação de algoritmos de machine learning (K-Means e Random Forest) e deep learning (LSTM) focados na previsão de demanda de estacionamentos. A aplicação dos algoritmos mostrou bons resultados na previsão de demanda, sendo os melhores obtidos pelo Random Forest. Por fim, é apresentada uma ferramenta modular, que integrou processos de mineração e análise de dados, fornecendo aos gestores um sistema para auxiliar na tomada de decisões do produto.

Palavras-chave: Smart Parking, Data Mining, Deep Learning, Business Intelligence.

Abstract

The technological advance and population growth of the last years brought a high demand for intelligent solutions that could improve the population's quality of life. One of these solutions is Smart Parking. This concept integrates different areas and aims to reduce the traffic flow of cities through the implementation of intelligent systems, focused on the control and management of parking lots. The present work integrated the development of an already structured Smart Parking System, which was conceived gradually by students and professors from UTFPR and IPB. It was proposed the creation of a data structure that integrated all the system modules. Moreover, a system that could help in the decision making process of the product was proposed, using as base the large volume of data generated by this kind of application. Consequently, the conceptual model used in the integration of the modules is presented, followed by the data mining and analysis steps. The creation of a model for data simulation and the implementation of machine learning (K-Means and Random Forest) and deep learning (LSTM) algorithms, focused on parking lot demand forecasting are also addressed. The application of the algorithms showed good results in predicting demand, the best results being obtained by Random Forest. Finally, a modular tool is presented, which integrated data mining and analysis processes, providing managers with a system to assist in product decision making.

Keywords: Smart Parking, Data Mining, Deep Learning, Business Intelligence.

Conteúdo

Lista de Tabelas

Lista de Figuras

Siglas

API Application Programming Interface.

BI Business Intelligence.

CeDRI Centro de Investigação em Digitalização e Robótica.

CSV Comma-separated values.

DL Deep Learning.

DM Data Mining.

GRU Gated Recurrent Units.

IA Inteligência Artificial.

IoT Internet of things.

IPB Instituto Politécnico de Bragança.

ITS Intelligent Transportation Systems.

KDD Knowledge Discovery in Databases.

LSTM Long Short-Term Memory.

ML Machine Learning.

NN Neural Networks.

RF Random Forest.

RMSE Root Mean Square Error.

RNN Recurrent Neural Networks.

SaaS Software as a Service.

SKLearn Scikit-Learn.

UML Unified Modeling Language.

UTFPR Universidade Tecnológica Federal do Paraná.

Capítulo 1

Introdução

Este capítulo visa apresentar a estrutura do trabalho e o problema levantado na pesquisa, identificando sua origem e cenário atual, em seguida contextualizando o ambiente no qual a proposta foi desenvolvida e finalizando com uma breve abordagem sobre a solução proposta.

1.1 Enquadramento

Com o vasto crescimento populacional das últimas décadas e o potencial de urbanização futuro, estima-se que até 2050, 86% da população de países mais desenvolvidos viverão em áreas urbanas [11]. Esse crescimento populacional produzirá um grande impacto na distribuição de serviços comuns das cidades, afetando diretamente setores como o transporte.

A consequência do crescimento do tráfego nas cidades é refletida pelo aumento do número e extensão de congestionamentos em grandes centros urbanos, o que reduz o fluxo de toda a região e faz com que a população perca cada vez mais tempo no trânsito.

Um estudo realizado por Guilherme Szczerbacki e Carlos Frickmann [41] apontou que, no Brasil, o tempo médio de locomoção diária da população é de 63,08 minutos e, em grandes centros metropolitanos esse tempo pode chegar a 100 minutos. Além disso, identificaram que as perdas econômicas consequentes de problemas na mobilidade urbana

podem chegar a 1,8% do PIB nacional, o que demonstra ser um problema de grande impacto no país.

Um dos fatores associados ao congestionamento urbano está atrelado aos estacionamentos, pois os motoristas fazem rodízios de forma indefinida buscando vagas disponíveis na sua região de interesse, fator esse que contribui para uma maior aglomeração de carros nas vias públicas, principalmente em regiões centrais [96].

Tendo em vista o problema, uma abordagem comum que vem sendo adotada para mitigá-lo é a criação e adoção de Smart Parkings, em português estacionamentos inteligentes, que podem ser definidos como: "Uma estratégia de estacionamento que combina tecnologia e inovação humana em um esforço para usar o mínimo de recursos possível - como combustível, tempo e espaço - para conseguir estacionamento mais rápido, fácil e denso de veículos"[90].

O Centro de Investigação em Digitalização e Robótica (CeDRI) do IPB possuí um trabalho de desenvolvimento de Smart Parking, realizado por alunos e professores, que acompanha a produção de todos os módulos dessa estrutura, desde seu desenvolvimento de hardware, até a comunicação com dispositivos (smartphones) e o tratamento de dados.

O Smart Parking auxilia na redução do fluxo de trânsito trazendo uma abordagem inteligente de controle e manutenção dos espaços de estacionamentos, permitindo que usuários possam reservar locais com horas ou dias de antecedência, indicando horários de entrada e saída. Dessa forma, evita-se a busca por estacionamentos por meio do rodízio e busca individual por vagas disponíveis, permitindo que os motoristas saiam de sua residência sabendo quando e onde irão estacionar.

1.2 Objetivos

O presente trabalho integra parte do desenvolvimento do produto Smart Parking, tendo por foco seu módulo de ciência de dados, mais especificamente fazendo o uso de tecnologias de Inteligência Artificial (IA), e Data Mining (DM), em português mineração de dados, focados em Business Intelligence (BI), em português inteligência de negócios.

Dessa maneira, o objetivo desse estudo é a criação de uma estrutura para realizar a análise de dados advindos do sistema de Smart Parking e implementar a aplicação de técnicas de Machine Learning (ML), em português aprendizado de máquina, e Deep Learning (DL), em português aprendizado profundo, sobre os dados para trazer uma maior percepção para os donos do produto a fim de os auxiliar no apoio à decisões relativas ao seu negócio.

Assim, esse estudo dispõe-se a elaborar as seguintes atividades no âmbito do projeto Smart Parking:

- Estruturar e integrar os dados utilizados por diferentes módulos em um padrão único para todo o produto.
- Realizar a mineração e tratamento dos dados do Smart Parking, tanto em arquivos quanto na nuvem.
- Desenvolver um modelo de simulação de dados de estacionamentos que contemple as variáveis utilizadas no sistema tendo em vista apoiar os testes e validação do mesmo.
- Elaborar e avaliar desempenho e performance de técnicas de ML e DL para previsão de dados sobre estacionamentos.
- Desenvolver uma plataforma WEB focada em BI que integre módulos de análise e visualização de dados, assim como modelos de ML e DL, visando trazer ao gestor do produto, uma única ferramenta que permita desde a identificação de padrões e características do estacionamento até a aplicação de algoritmos de previsões de demanda futura.

Os resultados obtidos no decorrer do processo são a identificação de visualizações de dados que permitam auxiliar gestores na tomada de decisão de seu produto em diferentes aspectos. Também estão incluídos a aplicação de algoritmos de ML e DL focados no contexto do Smart Parking, trazendo uma solução que abstrai a complexidade desse tipo de aplicação e a torna acessível de ser realizada por gestores do produto.

1.3 Estrutura do Documento

A estrutura do documento é divida em capítulos, focando-se no capítulo 2 o levantamento teórico de soluções atualmente aplicadas a esse problema, o contexto atual e tecnologias utilizadas. No capítulo 3 são abordadas as etapas de definição de um padrão de estrutura de dados que integre todos os módulos do sistema, seguido pela definição de requisitos e estruturação teórica da solução final. O capítulo 4 aborda o processo de desenvolvimento, desde a mineração e tratamento de dados existentes, a simulação de dados, a aplicação de conceitos de IA sobre os dados até o desenvolvimento do sistema integrado de análise de dados focado em BI. Por fim, no capítulo 5 serão apresentadas as conclusões e trabalhos futuros a serem realizados.

Capítulo 2

Estado da Arte

Nesse capítulo é percorrida toda a contextualização teórica acerca dos principais temas que integram a pesquisa, indicando o conceito no qual se enquadra o Smart Parking, suas principais características e suas variadas aplicações, casos de uso reais e estudos recentes sobre o tema. Além disso, aborda-se a utilização de ciência de dados aplicada no contexto específico de Smart Parkings, identificando a maneira como pode ser aplicada e apontando seus benefícios.

2.1 Cidades Inteligentes

O trânsito é um dos instrumentos presentes em todas as cidades, porém, as cidades são compostas por diversos outros componentes de igual importância, como sustentabilidade, distribuição de energia, distribuição de água, iluminação pública, entre outros. Dessa forma, visando tornar cidades mais produtivas, sustentáveis e inteligentes, foi introduzido o conceito de cidades inteligentes, em inglês Smart Cities.

Por se tratar de um contexto abrangente, que integra diversas características, como por exemplo, o controle do tráfego público, da iluminação municipal e distribuição de energia, tornando-se mais amplo com os avanços tecnológicos, as cidades inteligentes não possuem uma definição única, contudo, podem ser descritas de forma abrangente como "transformações digitais convertidas em melhorias de serviços públicos para a população,

para um melhor uso dos recursos com menos impacto no meio ambiente"[58]. Há também definições mais específicas, que consideram cidades inteligentes como um imenso sistema de informação composto por diversos subsistemas, os quais podem comunicar-se entre si tendo, cada um, um objetivo dentro do ecossistema da cidade, possuindo, entre outras, a capacidade de coleta e armazenamento de grande quantidade de dados [93].

A Figura 2.1 destaca os principais pontos e componentes que integram uma cidade inteligente. Esses componentes integram soluções inteligentes para aspectos que afetam o dia a dia de uma cidade, desde o controle do transporte e vias públicas, o controle da distribuição de recursos como água e energia, até aspectos de segurança pública e saúde da população.

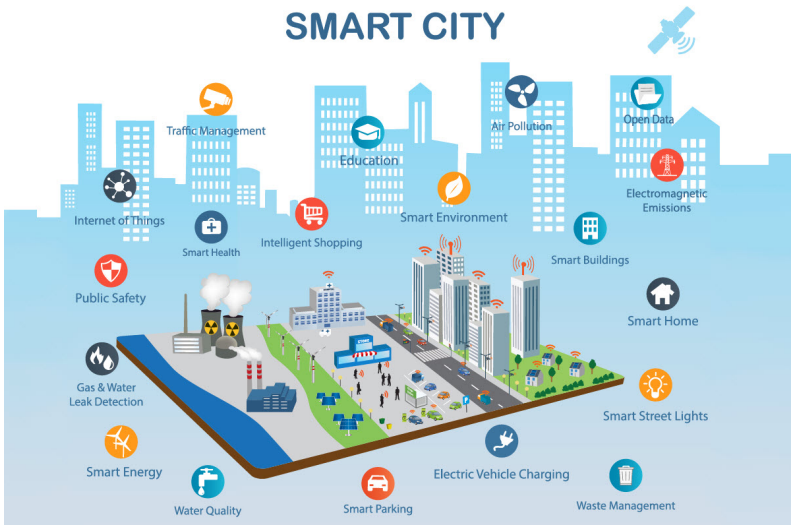


Figura 2.1: Exemplo de Estrutura de uma Smart City (adaptado de Digital Sign [24]).

Existem diferentes abordagens para o desenvolvimento e implementação de cidades inteligentes. Há o uso de arquiteturas baseadas em multicamadas, como o estudo [35] que propõe um modelo multicamadas, sendo executado na nuvem e distribuído para uso como Software as a Service (SaaS), um modelo de distribuição de softwares hospedados geralmente em nuvem. Segundo o mesmo estudo, cada camada é responsável por uma função, que abrange desde a coleta, processamento, integração e compreensão dos dados

emitidos por sensores, até a entrega de serviços customizados em formato de SaaS. Abordagens mais generalistas, como [32] propõem arquiteturas com o uso de um centro de informações integrado que opera por meio de um provedor de serviços IoT. Esse provedor conecta-se aos sistemas da cidade como transporte, saúde e energia, além de integrar-se com módulos auxiliares de nuvem, os quais permitem o acesso a funcionalidades de controle e armazenamento de dados e implementações SaaS.

2.1.1 Internet das Coisas

A inovação que permitiu a disseminação do conceito de cidades inteligentes, assim como da Indústria 4.0, estão intimamente ligadas ao crescimento e popularidade de dispositivos de Internet of things (IoT), em português Internet das Coisas , que nos últimos anos tem se tornado cada vez mais eficientes, compactos e acessíveis.

O termo IoT foi definido por Kevin Ashton em 1999, à época com a introdução de dispositivos RFID. Com o passar dos anos, os dispositivos IoT evoluíram, se tornando cada vez mais acessíveis e abrangendo praticamente qualquer coisa do cotidiano, por isso, definições mais recentes podem ser descritas como "a capacidade de fazer tudo ao nosso redor, desde máquinas, dispositivos, telefone celular e carros, até mesmo cidades e estradas, serem conectados à Internet com um comportamento inteligente e levando em consideração a existência do tipo de autonomia e privacidade."[43].

A quantidade de dados produzida nos dispositivos IoT proporciona extensas bases de dados de informação, as quais muitas vezes criam ambientes de Big Data, que podem ser caracterizados como conjuntos massivos de dados diversificados, podendo ou não ser estruturados chegando a uma velocidade acima do comum [74].

Essa massiva geração de dados desenvolvida nos módulos IoT, permite que haja uma quantidade de dados grande o suficiente a fim de possibilitar a realização de melhorias no sistema utilizando técnicas de mineração de dados e inteligência artificial, assunto em foco nesse trabalho.

No estudo [59] o autor apresenta o desenvolvimento de um sistema de Smart Parking

baseado em design de IoT, que abrange funcionalidades como o agendamento online, criação de tickets sem necessidade de papel, pagamentos digitais, além de um guia para estacionamentos. Em [85] também é apresentado um sistema de Smart Parking baseado em IoT, com o uso de sensores em todas as vagas a fim de identificar para o sistema e usuários sobre a disponibilidade de vagas do estacionamento.

2.1.2 Estacionamentos Inteligentes

Como foi descrito anteriormente, Smart Parkings integram as cidades inteligentes para fornecer todas as melhorias propostas para esse contexto no âmbito do trânsito, agregando dispositivos IoT (tais como sensores e controladores) focados nesse ambiente. Além disso, assim como as Smart Cities, o Smart Parking trata-se de uma complexa estrutura que ser dividida em diferentes módulos, como hardware, aplicação móvel, análises de dados e módulos de tarifação.

O SmartParking implementa parte do conceito de Sistema de Transporte Inteligente, em inglês Intelligent Transportation Systems (ITS), podendo ser descrito como "a combinação de alta tecnologia e melhorias em sistemas de informação, comunicação, sensores, controladores e métodos matemáticos avançados com o mundo convencional da infraestrutura de transporte"[55]. Dentro do contexto de ITS, o Smart Parking tem por foco o controle de estacionamentos, tanto públicos como privados.

Na literatura conceitual, o Smart Parking pode ser segmentado em dois grandes grupos distintos de estacionamentos, sendo eles:

- OnStreet Parking: Tratam-se de estacionamentos abertos e públicos, caracterizados normalmente como uma zona de espaço aberto, na rua, (medido ou não) com acesso direto de uma estrada, destinado a estacionar veículos [30].

- OffStreet Parking: Tratam-se de estacionamentos fechados, caracterizados em geral como um local fora da via, destinado ao estacionamento de veículos, gerido de forma independente e com pontos de acesso adequados e claramente sinalizados (entradas e saídas) [29].

Estacionamentos OnStreet e OffStreet se assemelham muito na estruturação e implementação de seus sistemas tecnológicos. Contudo, devido a algumas características que os diferem, é necessário considerar qual o tipo de estacionamento no qual pretende-se implementar a tecnologia.

2.2 Ciência de Dados

A área de ciência de dados, em inglês data science, assim como a de cidades inteligentes, abrange a união de diversas áreas e conceitos que, em conjunto, formam a estrutura desse estudo. Por se tratar de um conceito relativamente recente, a ciência de dados não possuí uma definição formal única, contudo, a ciência de dados pode ser definida de forma simples como um novo paradigma para a abordagem de problemas com técnicas de análise de dados [26][21].

O crescimento e popularização da ciência de dados foi exponencial nos últimos anos, sendo principalmente alavancada pela recente produção massiva de dados científicos [26] [97]. Não por coincidência, o desenvolvimento de estruturas IoT e de cidades inteligentes criou um ambiente muito propício para esse meio, visto que, a ciência de dados é uma peça chave para utilizar os dados produzidos visando trazer benefícios para o sistema, como a identificação de padrões e aplicação de algoritmos de previsão futura.

A ciência de dados contempla áreas multidisciplinares de estudo, como matemática, estatística, negócios e tecnologia da informação. A figura 2.2 detalha a união que forma esse conceito. Essa união é feita pela integração de conceitos de diferentes áreas, como ciência da computação, matemática e estatística e domínio de negócio. De forma simples, a ciência da computação utiliza-se de métodos matemáticos e estatísticos para criar soluções computacionais que auxiliem em problemas de negócio.

Dentro do contexto do presente trabalho, a área de ciência de dados também foi abordada, principalmente no que tange aos conceitos de mineração e análise de dados e aplicação de algoritmos de IA. Sendo essas, etapas complexas e fundamentais para atingir o objetivo proposto, já que são responsáveis por boa parte do processo desenvolvido para

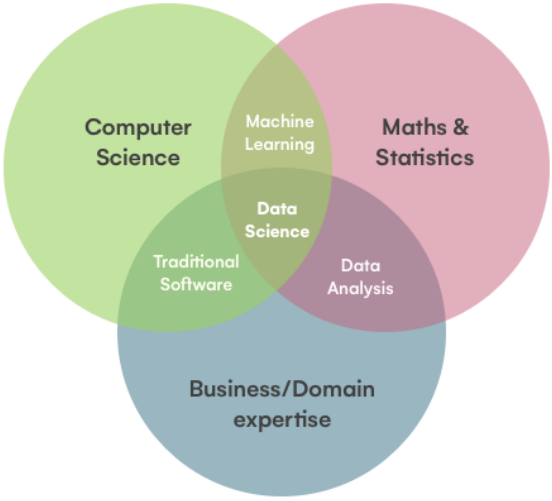


Figura 2.2: Conceito de Ciência de Dados (adaptado de Iron Hack [49]).

alcançar os resultado esperados, desde a obtenção e tratamento dos dados, até a aplicação de análises que produzem os resultados finais.

2.2.1 Data Mining

DM é um módulo presente dentro do escopo da ciência de dados, podendo ser definida como o processo de realizar buscas e pesquisas em grandes volumes de dados a fim de obter informações úteis e relevantes [20] [23]. O processo de mineração de dados compreende desde a captura da informação, o processamento dos dados, a criação de modelos até a visualização e análise dos dados.

A aplicação de mineração de dados pode ser realizada em diversas áreas do conhecimento, como indústrias, e-commerce e governança pública, sendo nesse último, utilizada em aplicações como a melhoria de serviços públicos e a descoberta de problemas que afetam cidadãos [23]. Dessa forma, sistemas de Smart Parking também podem se beneficiar dessa técnica. No estudo [44] o autor propõe um sistema de mineração de dados em nuvem, baseado em modificações de modelos de Regressão Linear e Naïve Bayes, extraindo informações de sensores e de usuários do estacionamento.

Durante a etapa de análise da mineração de dados, são aplicados diversos algoritmos de IA, tanto focados em ML como DL. A IA é um conceito introduzido ainda no início da computação, sendo que sua primeira aplicação é datada por Herbert Simon em 1955. Desde seu surgimento, sua definição foi bastante modificada, abrangendo cada vez mais contextos e áreas, contudo, de forma ampla, a IA é um domínio que se preocupa em desenvolver sistemas que apresentem características que podem ser associadas à inteligência humana [87].

Nas últimas décadas, a aplicação de algoritmos de inteligência artificial é realizada em todos os setores que possam ser beneficiados com essa técnica, incluindo ambientes de Smart Parking, nos quais a IA pode ser utilizada para, entre outros objetivos, identificar vagas livres no estacionamento, utilizando, por exemplo, tecnologias de detecção de imagem [69].

Em [18] os autores comparam diversos algoritmos de IA aplicados em uma base de dados de estacionamentos, a fim de obter previsões de demanda futura mais precisas. A utilização de IA também pode ser aplicada nos sistemas IoT de Smart Parkings, como em [42] no qual foi proposto um sistema IoT com ML para resolver problemas de gerenciamento do estacionamento em tempo real, visando também a previsão de vagas de estacionamentos livres para os usuários finais.

2.2.1.1 Random Forest

O algoritmo florestas aleatórias, em inglês Random Forest (RF), é um algoritmo supervisionado de ML. O termo florestas é utilizado pois o algoritmo baseia-se em outro algoritmo de ML, denominado árvore de decisão, contudo, sua abordagem cria diversas árvores de decisão, com a ideia de que a combinação de diversas árvores pode trazer resultados mais precisos para as previsões finais [25].

O funcionamento da árvore de decisão é bastante simples, partindo de variáveis categóricas dos dados, cria-se uma estrutura de árvore que abrange todas as possibilidades para cada variável. A figura 2.3 apresenta um exemplo de árvore de decisão que considera 3 variáveis do produto.

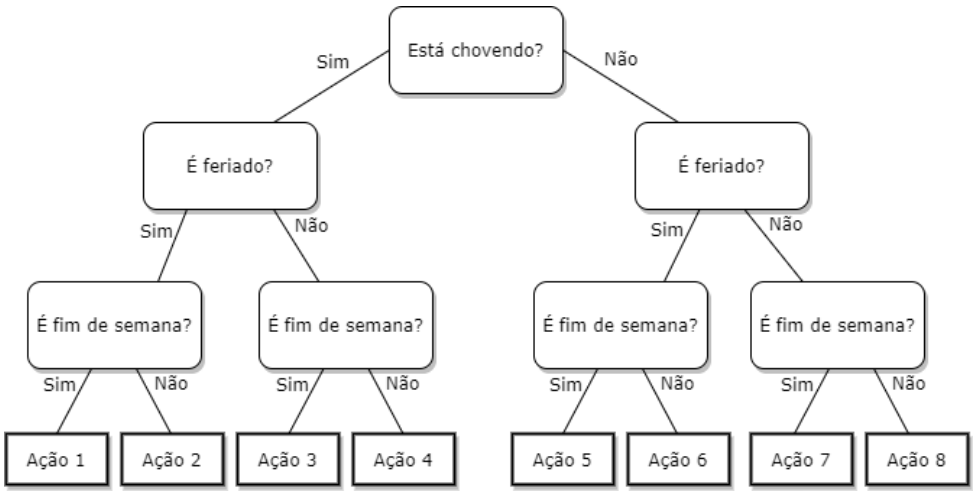


Figura 2.3: Exemplo de funcionamento de árvore de decisão.

A função da árvore de decisão é conseguir separar ao máximo os dados por suas características, a fim de que cada folha da árvore seja o mais distinta possível de outra folha, ao mesmo tempo em que todo conjunto de dados que acabar pertencendo à mesma folha seja o mais similar possível [91]. Assim, após o modelo de árvore ser finalizado, quando um novo dado for inserido, ele irá percorrer toda a lógica da árvore e ao entrar em uma folha, obtém-se a previsão de suas características advindo das características de dados semelhantes da mesma folha.

2.2.1.2 K-Means

O K-Means é um algoritmo de IA não supervisionado bastante popular e de fácil implementação. Algoritmos não supervisionados procuram fazer inferências sobre os dados sem terem recebido instruções ou classificações prévias [33]. A técnica utilizada por esse algoritmo é a clusterização, que trata-se do processo de dividir o conjunto de dados em sub-grupos de dados similares entre si, sendo esses sub-grupos denominados de clusters [77].

A ideia por trás da clusterização está em agrupar os dados semelhantes em diferentes regiões de um gráfico, e conectá-los a partir de clusters, formando por fim um modelo. Dessa maneira, quando um novo dado for recebido, ele é inserido no gráfico do modelo e

será integrado em um cluster, e após ser integrado, é possível deduzir suas características pelos dados semelhantes que integram o mesmo cluster.

2.2.2 Deep Learning - LSTM

Deep Learning (DL) é uma área do presente dentro do escopo de ML, possuindo objetivos semelhantes, porém com uma abordagem bastante distinta. O DL aplica conceitos de redes neurais [45], em inglês Neural Networks (NN), que proporcionam uma estrutura multicamadas, a qual recebe uma entrada que transita entre as camadas, podendo ser alterada no decorrer do processo para, por fim, obter-se uma saída como resposta. Essas camadas são a composição de múltiplas transformações, tanto lineares como não lineares [54].

Um importante fator que torna aplicações de DL interessantes e as diferencia de ML é de que, assim como algoritmos de ML, os modelos de DL desenvolvidos são treinados e aprendem com os dados propostos. Contudo, diferente de modelos de ML onde são necessários uma extensa base de dados supervisionada, as aplicações de DL trabalham com dados não supervisionados, permitindo que o modelo possa aprender e se adaptar aos dados, sem a necessidade prévia de supervisão humana, beneficiando-se de sua estrutura multicamada [46] [57]. Dessa forma, são produzidos modelos com previsões adaptadas para cada tipo de dado, podendo ser mais precisas que aplicações de ML tradicionais.

Como o treinamento do DL baseia-se em estruturas não supervisionadas, a qualidade da precisão de seus resultados está geralmente atrelada à quantidade de dados, sendo melhor quanto maior a quantidade de dados [46]. O DL é aplicado em contextos como processamento de imagem e processamento de linguagem natural. Esses contextos produzem dados de grandes dimensões e de difícil processamento, sendo incapazes de ser realizados por algoritmos de ML [34].

O DL também é utilizado para a previsão de séries temporais [6], nome dado para tipos de dados que variam e tem dependência do tempo. A modelagem de série temporal é uma abordagem popular para fazer previsões em problemas de transporte [51]. Existem

diferentes modelos de DL focados em séries temporais, como por exemplo, Long Short-Term Memory (LSTM) e Gated Recurrent Units (GRU). No estudo [6] o autor conclui que os algoritmos LSTM e GRU produzem previsões de séries temporais muito precisas, sendo o LSTM mais eficiente para conjuntos de dados estáveis.

Para o presente trabalho foi utilizado o algoritmo LSTM na aplicação de DL, um algoritmo baseado em redes neurais recorrentes, em inglês Recurrent Neural Networks (RNN), que por sua vez são estruturas dinâmicas de redes neurais que contam com neurônios e conexões. Segundo Staudemeyer et al, "Isso se deve às conexões circulares entre os neurônios das camadas superior e inferior e às conexões opcionais de feedback automático. Essas conexões de feedback permitem que os RNNs propaguem dados de eventos anteriores para as etapas de processamento atuais. Assim, os RNNs constroem uma memória de eventos de série temporal"[83].

O LSTM por sua vez, ao introduzir modificações na RNN, torna a estrutura mais robusta e versátil, essas mudanças estão principalmente na introdução de controles nas células da rede neural [78], permitindo dessa maneira, um treinamento que alcance melhores resultados.

Em [10] o autor apresenta uma solução baseada em Redes Neurais utilizando métodos de LSTM com o intuito de gerar previsões de vagas disponíveis para os usuários do estacionamento. No estudo [80] é proposta uma solução que utiliza o conceito de ML extremamente profundo na detecção de tráfego rodoviário, no qual os autores relataram uma taxa de precisão de 91% na previsão de vagas futuras do local.

Em um estudo comparativo entre abordagens de IA e DL em Smart Parkings, os autores relatam que "independentemente do tamanho do conjunto de dados, os algoritmos menos complexos, como árvore de decisão, floresta aleatória e KNN superam algoritmos complexos, como multicamadas Perceptron, em termos de maior precisão de previsão, enquanto fornecem informações comparáveis para a previsão de disponibilidade de espaço de estacionamento"[15].

2.3 Business Intelligence

Business Intelligence (BI) é o processo de transformar dados em percepções e informações úteis que são utilizadas para a tomada de decisão [79]. A área de atuação de BI pode muitas vezes ser confundida com a ciência de dados, pois a ciência de dados, por meio da análise, também visa a descoberta de padrões e conhecimentos utilizando dados, contudo, é possível distinguir os dois campos de estudo em diferentes aspectos. A Figura 2.4 detalha o foco de cada área, destacando suas diferenças.

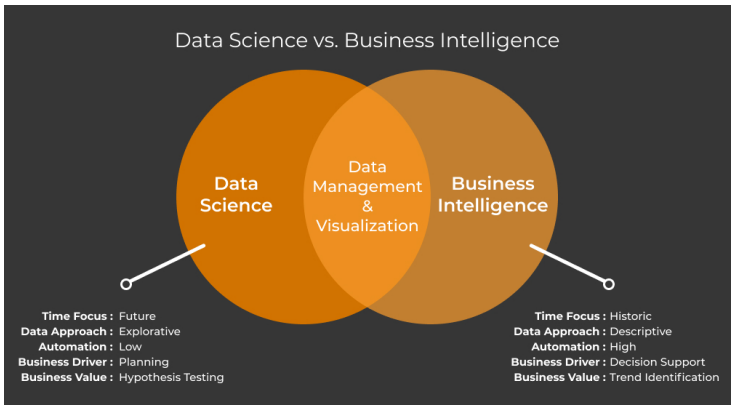


Figura 2.4: Diferenças entre Ciência de Dados e Inteligência de Negócio (adataptado de Intelegain [86]).

O maior destaque a ser considerado em BI é seu foco no suporte para tomada de decisões. Segundo Negash, "inteligência de negócios é usado para entender os recursos disponíveis na empresa; o estado da arte, tendências e direções futuras nos mercados, as tecnologias e o ambiente regulatório em que a empresa compete".

No contexto de Smart Parkings, existem diversas plataformas de BI como o Smarking [81], um sistema para gestão de ativos de estacionamentos, que conta com projeção de receita e taxa de ocupação dos estacionamentos, além de detecção de comportamento de usuários e recomendações de negócio.

Em um catálogo de 2016 da Optimum Parking Management [48], é ressaltada a importância da integração de BI em ambientes de Smart Parking, que ressaltam que profissionais e ferramentas de BI pode trazer para esse contexto benefícios como:

- Automação e rapidez: Captura e gestão de dados automatizada, disponível 24 horas por dia e integrada ao sistema automaticamente.

- Relatórios dinâmicos: Geração de relatórios dinâmicos, adaptados ao modelo e necessidades do negócio.

- Painéis customizados: Disponibilização de painéis interativos, disponíveis em diferentes plataformas, atualizados com informação em tempo real que tragam de forma fácil e simplificada, a visualização de tendências e comportamento do estacionamento.

Capítulo 3

Especificação e Metodologia

Neste capítulo é realizada a descrição do problema de forma detalhada, especificando a metodologia de desenvolvimento do trabalho, identificando as etapas executadas durante todo o processo. Por fim, são especificados os requisitos da solução proposta e sua arquitetura, por meio de casos de uso e diagrama de atividades.

3.1 Problema

Como esclarecido anteriormente, o sistema de Smart Parking é uma estrutura final resultante da conexão e trabalho conjunto de diferentes módulos, cada qual com suas especificidades e objetivos. Dessa forma, o produto Smart Parking desenvolvido no Instituto Politécnico de Bragança (IPB) e na Universidade Tecnológica Federal do Paraná (UTFPR) faz parte desse contexto, sendo que, para seu desenvolvimento, foram agregadas diversas áreas do conhecimento, as quais percorrem desde a estrutura física de sensores e dispositivos IoT até o produto final digital, entregue ao usuário em formato de sistema web ou aplicativo.

A união de diferentes módulos, desenvolvidos utilizando diferentes tecnologias e em momentos distintos, criou um problema de compatibilidade na etapa de integração entre as estruturas, etapa na qual visa-se obter o produto final. Esse problema ocorre em dois pontos de grande relevância, sendo o primeiro deles em como desenvolver a plataforma que

realiza a conexão entre os módulos, e o segundo na estrutura de dados para comunicação, devido à falta de padronização da estrutura de dados.

Assim, o presente trabalho tem por objetivo solucionar problemas na integração da estrutura de dados dos módulos do produto Smart Parking. Além disso, o trabalho ainda propõe uma solução focada em BI , utilizando a estrutura de dados desenvolvida na integração, a fim de fornecer ao sistema uma plataforma web que permita a execução de análises de dados, modelos de previsão de demanda futura, visualização de tendências e identificação de padrões.

Para alcançar o objetivo proposto, as atividades desenvolvidas foram definidas e organizadas em ordem cronológica, sendo descritas na figura 3.1.

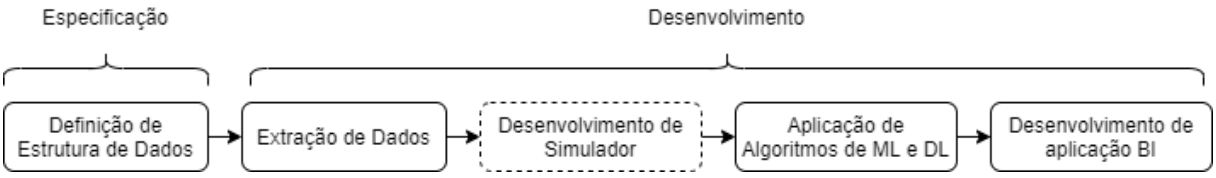


Figura 3.1: Metodologia de Desenvolvimento.

A etapa inicial de extração de dados é fundamental para todo o desenvolvimento futuro da pesquisa e do produto, pois essa etapa integra o estudo de estruturas de dados aplicadas em Smart Parking, verificando variáveis importantes para a organização e controle desse tipo de negócio. Além disso, é realizado também o levantamento de variáveis necessárias para os módulos específicos do produto atual, como variáveis focadas em trazer informações relevantes nas etapas de análise de dados e BI, como o registro do clima.

A aquisição de dados iniciou-se após a definição da estrutura, tendo por objetivo a busca por bases de dados públicas de estacionamentos inteligentes. Essa busca foi necessária pois o módulo a ser desenvolvido necessita de aplicações sobre grandes bases de dados, tanto na aplicação de algoritmos de IA, quanto para visualizações de BI.

A etapa de desenvolvimento de um simulador encontra-se destacada na figura 3.1 pois, inicialmente, não era uma etapa considerada no processo. Contudo, devido à escassez de dados públicos focados especificamente no contexto de Smart Parkings e principalmente,

pela regra de negócio muito específica do produto, não haviam dados que abrangessem todas as variáveis necessárias. Dessa maneira, criou-se uma etapa para o desenvolvimento de um sistema que visa simular registros de estacionamentos, considerando nessa simulação, as variáveis de Smart Parkings tradicionais e as variáveis específicas do produto.

A aplicação de algoritmos de ML e DL nos dados do sistema foi realizada após a obtenção de bases de dados públicas e bases de dados simuladas.

Essa etapa teve por foco testar aplicações de ML e DL sobre os dados, com o objetivo de obter previsões futuras de demanda baseadas no uso de cada estacionamento.

A última etapa, a de desenvolvimento da aplicação BI tem por objetivo a criação de uma estrutura que permita a leitura de dados de diversos estacionamentos, fornecendo ao gestor do produto variadas páginas para visualização e análise de dados. Além disso, a aplicação integra os algoritmos de ML e DL desenvolvidos na etapa anterior, a fim de permitir que o gestor possa aplicar esses algoritmos em cada estacionamento, obtendo assim previsões de demanda futura.

3.2 Integração de Módulos

Dada a complexidade e ampla gama de módulos do produto, considerando também que cada módulo foi desenvolvido por diferentes alunos e professores , utilizando ferramentas e técnicas distintas, em variados períodos nos últimos anos, ocorreu a necessidade de integração entre esses módulos. Essa integração foi necessária pois as características distintas de cada módulo não convergiam de forma padrão para uma única estrutura centralizada, que pode ser considerada como o produto final. Por esse motivo, foi necessário a elaboração de um modelo para a integração das estruturas.

Dessa forma, haviam dois problemas em relação à integração dos módulos do produto, sendo um deles em relação à estrutura de comunicação e outro em relação à estrutura de dados utilizada.

3.2.1 Integração Estrutural

O presente trabalho não tem por objetivo direto a integração estrutural do sistema, sendo o seu foco na integração de dados. Contudo, o processo estrutural define a tecnologia a ser utilizada na integração de módulos. Essa escolha interfere diretamente na definição do banco de dados utilizado no processo de integração de dados, pois o mesmo deve obrigatoriamente ser compatível com a estrutura integradora.

A integração estrutural trata-se da integração dos módulos em um mesmo ambiente, no qual consigam ter acesso uns aos outros e possam se comunicar utilizando um ambiente em comum. Assim, a abordagem mais comum e moderna para a criação de conexões entre diferentes componentes eletrônicos e digitais trata-se da internet , havendo por esse meio, diferentes ferramentas que podem atender os requisitos, como servidores privados e serviços de nuvem.

3.2.1.1 Modelo Atual

O produto Smart Parking desenvolvido no IPB no qual esse trabalho é baseado, tem por foco apenas estacionamentos do tipo OffStreet, que são estacionamentos com acessos de entrada e saída, sendo geralmente locais fechados. A Figura 3.2 ilustra os módulos necessários para o funcionamento de um sistema de Smart Parking, utilizando diferentes cores para identificar cada módulo e suas respectivas funções. A seguir serão descritos brevemente cada módulo presente na estrutura, onde:

- CPS: trata-se do sistema ciber-físico do produto, focado em dispositivos IoT e sensores. Permitem, por exemplo, a liberação e bloqueio de travas de estacionamentos e a contagem de número de vagas preenchidas utilizando sensores. Esse módulo utiliza sistemas multiagentes.

- Gerenciamento: esse módulo tem por objetivo fornecer aplicações de alto nível para permitir acesso do cliente ao produto Smart Parking com a disponibilização de aplicativos, além de interfaces de controle para gestores.

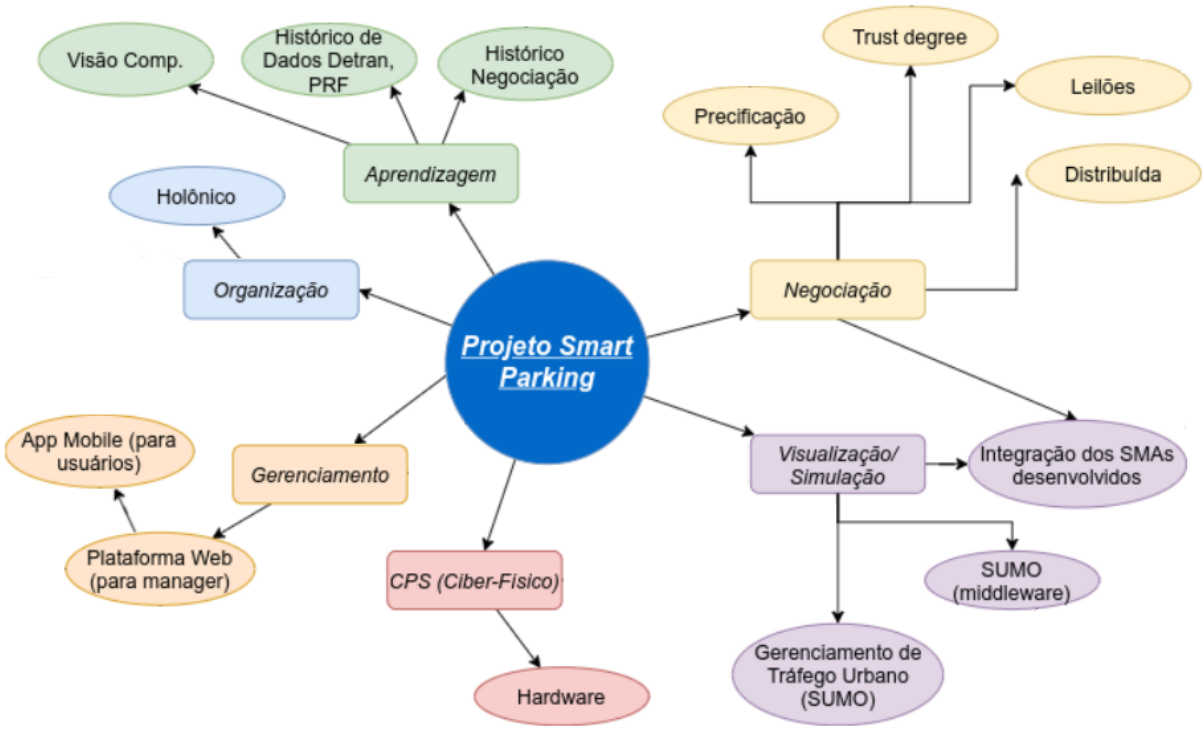


Figura 3.2: Estrutura do Produto Smart Parking.

- Aprendizagem: esse módulo tem como sua característica principal a implementação de estruturas e modelos de ML e DL no produto.

- Organização: compõe modelos de estruturação de manufatura do produto, se beneficiando da estrutura de multiagentes do módulo CPS. Um exemplo é o modelo Holônico, que visa a criação de uma estrutura de produção distribuída, flexível, inteligente e autônoma, que desenvolva de maneira rápida lotes de pequenas quantidades, reduzindo custos e tornando as organizações mais competitivas [70].

- Negociação: o módulo de negociação abrange algoritmos e estruturas de comunicação, disputa e concorrência entre os multiagentes, visando proporcionar um ambiente no qual possa, por exemplo, ser escolhido os melhores preços para cada situação de estacionamento. Essa situação é identificada levando em conta variáveis como o tipo de vaga, fluxo do local, variáveis ambientais, horário do estacionamento, entre

outros.

- Visualização e Simulação: permite desenvolver e empregar estruturas que simulem diferentes aspectos de um estacionamento, tendo por objetivo fornecer dados de forma simulados que tenham o máximo de fidelidade com estacionamentos reais. Isso permite que seja possível executar melhorias e testes em versões simuladas antes de serem introduzidas em ambiente de produção.

3.2.1.2 Modelo de Integração

Levando em conta a complexidade e principalmente a diversidade dos componentes que precisam ser interligados, uma abordagem eficiente utilizada é a de computação em nuvem. A computação em nuvem oferece uma estrutura que contempla diversos serviços em um único ambiente, como por exemplo, proporcionar a comunicação de dispositivos IoT e aplicativos móveis. Além disso, a nuvem conta com o acesso de recursos sob demanda, escaláveis e com soluções de software, armazenamento e infraestrutura [39].

O uso da nuvem para integração de aplicações pode trazer benefícios como "oferecer escalabilidade para permitir expansão futura em termos de número de usuários, número de aplicativos ou ambos. Também reduz a dependência de TI e integra SaaS aplicativos mais rápido [60].

Outro importante fator que o uso da nuvem proporciona é a segurança, pois a estrutura possuí protocolos de segurança para todos os tipos de conexões que recebe, tanto em conexões diretamente via hardware, quanto via software. Além disso, a nuvem integra funcionalidades que possibilitam uma fácil conexão com smartphones [52], criando uma ponte que conecta as ações executadas pelo usuário em seus aplicativos diretamente com sensores e dispositivos IoT. Esse processo é feito por meio de API, um padrão de comunicação entre dispositivos que permite a comunicação entre diferentes tipos de aplicação pela internet, permitindo que dispositivos de diferentes tipos e sistemas possam trocar informação de forma simples e compatível.

Existem diferentes serviços de computação em nuvem no mercado, sendo os principais deles a Amazon AWS [2], Microsoft Azure [5] e Google Cloud [3]. Esses serviços representaram, em 2020, um percentual de mercado de 32%, 19% e 7% respectivamente [82], totalizando cerca de 58% de todo o mercado de serviços de computação em nuvem. Ambas possuem uma estrutura que garante a conexão de todos os módulos presentes no produto Smart Parking, a escolha pautou-se pela preferência da equipe, sendo o Google Cloud escolhido como o serviço para realizar a integração estrutural do produto.

3.2.2 Integração de Dados

A integração da estrutura de dados foi o primeiro passo importante na pesquisa, essa integração tem por objetivo definir quais serão as estruturas e o modelo de dados que será utilizado na comunicação e armazenamento dos diferentes módulos do produto.

Sabendo que as características e variáveis usadas no produto podem variar entre os módulos, a padronização usando um modelo de dados relacional tornaria o processo mais complexo, devido à sua baixa flexibilidade para mudanças estruturais. Por esse motivo, optou-se pela utilização de bases de dados não relacionais, que são flexíveis e possuem extenso suporte na estrutura de nuvem.

3.2.2.1 Modelo Fiware

Visando basear-se em um modelo de dados amplamente conhecido e consolidado, optou-se por utilizar o modelo e estrutura de dados da Fiware [31]. A Fiware caracteriza-se como uma plataforma de código aberto que define padrões e conceitos de soluções inteligentes. A plataforma conta com estruturas de dados para soluções inteligentes em domínios de cidades, controle ambiental, sensores, agricultura e alimentação, energia, entre outros.

Em 2018, o modelo da Fiware para Smart Cities, o qual abrange variados módulos de cidades inteligentes, foi adotado pelo Mecanismo de Conexão da Europa (CEF), um programa da união europeia que visa integrar infraestruturas a fim de promover o crescimento de oportunidades de trabalho e competitividade regional [27]. O modelo de dados

inteligente para Smart Parking integra o modelo de Smart Cities da plataforma.

O modelo de dados para Smart Parking contém estruturas de dados e variáveis para estacionamentos OnStreet e OffStreet, além de abranger outros componentes que pertencem ao estacionamento [31], sendo eles:

- ParkingAccess: um ponto de acesso a um estacionamento fora da rua, geralmente em estacionamentos OffStreet.
- ParkingSpot: vaga de estacionamento monitorada de forma individual por um sensor ou dispositivo IoT.
- ParkingGroup: agrupamento de diferentes vagas de estacionamento. O nível de abrangência do grupo pode ser variável, podendo ser um piso de um estacionamento, ou uma área específica pertencente a um grande estacionamento ou somente um conjunto de vagas.

A figura 3.3 apresenta a hierarquia básica de um Smart Parking do tipo offStreet utilizando a estrutura de dados inteligente Fiware. As setas indicam a conexão entre um elemento superior ao outro, como o *"OnStreetParking"* contém um conjunto de *"ParkingGroups"* e esses contém um conjunto de *"ParkingSpots"*. As ligações pontilhadas entre elementos e *"ParkingAccess"* definem que esse tipo de estrutura é opcional no estacionamento.

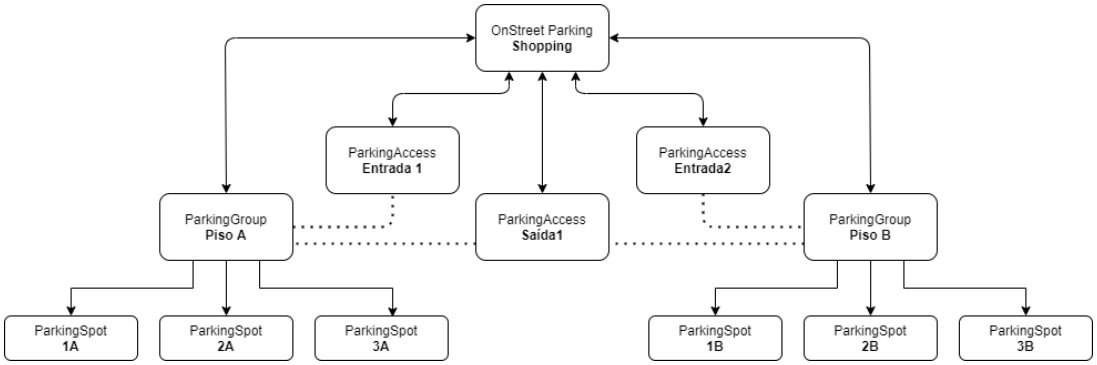


Figura 3.3: Exemplo da estrutura Fiware para estacionamento offStreet.

3.2.2.2 Modelo de Dados

A regra de negócio do produto Smart Parking desenvolvido teve de ser incorporada à estrutura de dados proposta, baseada no modelo Fiware, para que essa pudesse se adaptar as características do produto. Dessa maneira, diversas características individuais do produto tiveram de ser levantadas para integrar o modelo de dados proposto para o sistema, sendo as principais características descritas a seguir:

- Clima: o clima foi adicionado como uma variável categórica, inicialmente identificando-se no momento da realização do estacionamento estava ou não chovendo, por exemplo;
- Feriado: indica se o dia em que o estacionamento foi realizado trata-se de um feriado ou não, sendo também uma variável categórica; e
- Fim de semana: também categórica, essa variável indica se no dia em que realizou-se o estacionamento era um final de semana ou não.

A escolha por essas variáveis foram baseadas em tendencias de fluxo observadas no estacionamento da universidade. Todos os três fatores elencados modificavam o fluxo do estacionamento, como um acréscimo de carros em dias de chuva, ou a redução deles em finais de semana e feriados, dias em que menos profissionais trabalham na universidade. Dada a complexidade que poderia haver em tratar as três variáveis, optou-se por restringir apenas à esse escopo as regras específicas do modelo.

3.2.2.3 Modelo Proposto

Considerando o modelo Fiware, em conjunto com as características individuais do produto, propôs-se um modelo de dados para integrar a estrutura do sistema, utilizado para a comunicação entre todos os módulos. A comunicação abrangeria desde os dispositivos de hardware, até aplicações de usuários e aplicações de análise de dados, sendo os dados transmitidos dentro da nuvem.

Além disso, o modelo proposto permite também maximar futuras análises de dados e aplicações de BI sobre o produto, além de permitir uma estrutura adequada, na qual houvesse as informações necessárias para todos os módulos realizarem a comunicação, mas também uma estrutura que pudesse maximar futuras análises de dados e aplicações de BI sobre o produto.

As tabelas 3.1 e 3.2 apresentam, respectivamente, o modelo proposto de estrutura de dados para regiões de estacionamento e para ações de estacionamento. O modelo de regiões de estacionamento é baseado na estrutura de *OnStreetParking* e o modelo de ações é baseado nas estruturas de *ParkingGroup* e *ParkingSpots*, citadas na seção 3.2.2.1.

As variáveis da tabela 3.1 visam identificar o estacionamento como um todo, sendo criada uma estrutura do tipo para cada estacionamento. Estas contém informações que auxiliam sua identificação física como nome e endereço, além de características do local como o tempo máximo de estacionamento e tipos de veículos que ele permite.

As variáveis da tabela 3.2 tem por objetivo identificar a ação de estacionar, sendo criado um registro para cada estacionamento realizado em qualquer local. As variáveis abrangem o estacionamento no qual foi realizado, dados de entrada e saída do local e as variáveis de negócio, incluindo clima, feriado e fim de semana.

3.3 Análise de Requisitos

Tendo em vista que a solução final proposta trata-se de um software, a análise de requisitos foi aplicada para a definição do escopo da solução, a fim de identificar a estrutura e as funcionalidades que devem ser atendidas pelo sistema.

O conceito utilizado foi o de Linguagem de Modelagem Unificada, em inglês Unified Modeling Language (UML), que trata-se de um padrão para a modelagem de software, sendo utilizado para aprimorar a visualização e compreensão da estrutura e comportamento relativos ao sistema [8]. A aplicação de UML é feita em diversas etapas, dependendo da complexidade do projeto e da compreensão de seus requisitos. Dessa maneira, podem

Variável	Tipo	Descrição
address	Texto	Endereço completo do estacionamento.
coordinates	Vetor	Vetor contendo os valores de latitude e longitude do respectivo local.
description	Texto	Descrição básica do estacionamento.
image	Texto	Endereço de URL para uma imagem que descreve o local.
maxDuration	Numérico	Variável que define o máximo de horas contínuas que usuários podem estacionar no local.
name	Texto	Nome do estacionamento.
statistics	Objeto	Armazena estatísticas do estacionamento que são produzidas pela ferramenta de BI.
vehiclesAllowed	Vetor	Identifica quais tipos de veículos podem estacionar no local (carros, motos, bicicletas).

Tabela 3.1: Estrutura de dados de estacionamento (local).

existir projetos com mais ou menos etapas de desenvolvimento, como por exemplo, modelos de prototipação, baseados em criações rápidas que visam compreender os requisitos ou métodos interativos com desenvolvimentos cíclicos. Nesse trabalho foi aplicado o uso de prototipação, pois essa metodologia permite identificar de forma rápida se os protótipos atendem os requisitos da solução proposta.

Visando levantar e identificar os requisitos propostos pela solução final, foram desenvolvidos diagramas de casos de uso e diagrama de atividades, para identificar o comportamento do sistema, seus participantes, fluxo de funcionamento e funcionalidades.

3.3.1 Casos de Uso

Os casos de uso fazem parte do contexto da UML, sendo utilizados para documentar os requisitos do sistema. Sua aplicação é importante para a análise comportamental do sistema, pois seu objetivo é relacionar a comunicação de todas as pessoas e objetos que interagem com a aplicação, elencando quais ações que cada um desses atores pode exercer na mesma [53].

Variável	Tipo	Descrição
parking	Texto	Identificador do local onde realizou-se o estacionamento.
parkingName	Texto	Nome do estacionamento.
region	Texto	Identificador único da região de estacionamento.
regionName	Texto	Nome da região onde foi realizado o estacionamento.
distanceRange	Numérico	Distância máxima que o usuário aceita uma vaga advinda do sistema.
maxPrice	Numérico	Preço máximo que o usuário definiu na requisição da vaga.
locationWeight	Numérico	Percentual de valorização da localização escolhida pelo usuário.
spotWanted	Numérico	Número da vaga que o usuário solicitou na requisição.
timeFrom	Data e Hora	Data e hora da entrada no estacionamento.
timeTo	Data e Hora	Data e hora da saída do estacionamento.
priceWeight	Numérico	Percentual de valorização do preço escolhido pelo usuário.
spotWanted	Numérico	Vaga que o usuário requisitou.
spotWon	Numérico	Vaga que o sistema atribuiu ao usuário.
priceWon	Numérico	Preço que o sistema atribuiu ao usuário.
userId	Texto	Identificador único do usuário.
vehicleId	Texto	Identificador único do veículo do usuário.
isRainy	Booleana	Identifica o clima na região do estacionamento no horário de entrada do usuário.
isHoliday	Booleana	Identifica se o dia em que o estacionamento foi realizado é feriado ou não na cidade do estacionamento.
isWeekday	Booleana	Identifica se o dia em que o estacionamento foi realizado é um dia de semana ou não.

Tabela 3.2: Estrutura de dados de estacionamento (ação de estacionar).

A figura 3.4 apresenta o caso de uso geral do sistema, identificando quais atores interagem com o mesmo e quais funcionalidades são proporcionadas, abstraindo-se um conjunto

de funcionalidades em grandes escopos gerais, que posteriormente serão desmembrados e detalhados individualmente, sendo apresentados nas figuras 3.5, 3.6 e 3.7.

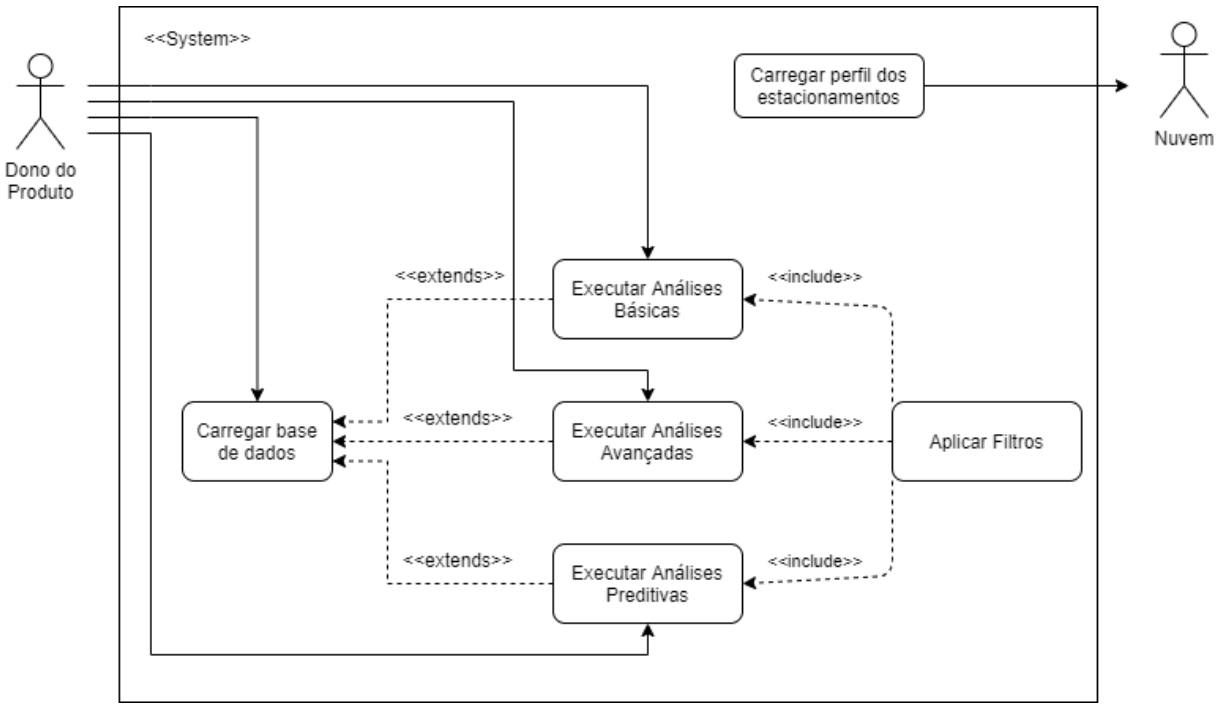


Figura 3.4: Caso de uso geral da solução proposta.

Como é especificado nos casos de uso, o sistema conta com diversas funcionalidades, focadas principalmente na visualização e análise de dados. Por se tratar de uma ferramenta de BI, a apresentação de relatórios de forma gráfica é um requisito fundamental, pois é uma maneira simples de apresentar os dados, permitindo que possam ser visualizados de forma rápida e compreendidos com facilidade por gestores do produto. Além disso, a execução de algoritmos de IA focados na previsão de demanda, requisito do sistema, integra o escopo de análises preditivas.

O sistema possuí como atores somente o gestor do produto e a nuvem, não havendo terceiros ou clientes acessando o produto. Isso ocorre porque trata-se de uma aplicação para gerenciamento de negócio. O gestor tem acesso a maior parte das ações do sistema, enquanto que, a nuvem é utilizada para trazer dados do produto ao sistema e também

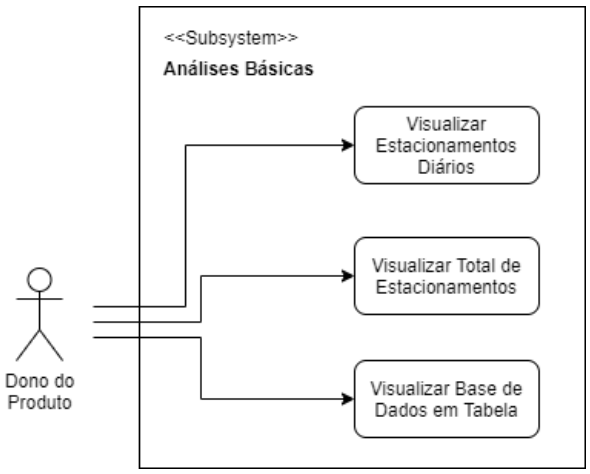


Figura 3.5: Caso de uso para análises básicas.

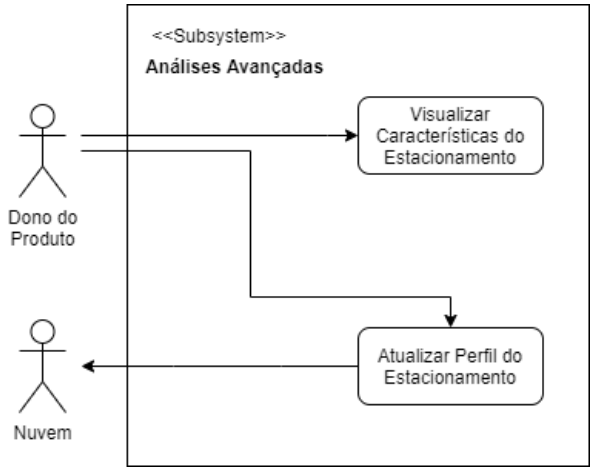


Figura 3.6: Caso de uso para análises avançadas.

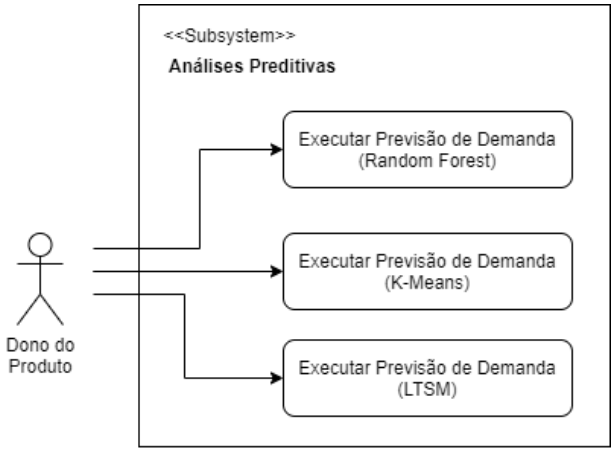


Figura 3.7: Caso de uso para funcionalidade de previsão de demanda.

para executar uma atualização de dados quando requisitada pelo gestor.

A escolha pelo carregamento da base de dados ser feita pelo gestor foi decidida para que não houvesse a necessidade da captura e atualização de dados em tempo real pelo sistema, fator que poderia sobrecarregar a nuvem com requisições constantes. Considerando que características comportamentais de estacionamentos podem levar bastante tempo para serem alteradas, devido à necessidade de um volume razoável de dados para que o todo sofra alterações significativas. Essa abordagem permite que as análises possam ser realizadas em diferentes períodos de tempo, sem que houvesse uma perda significativa

de informação.

Com isso, o passo inicial, antes de iniciar a aplicação, é alimentá-la com uma base de dados de estacionamentos retirados do sistema. Além disso, uma segunda base de dados é necessária para a aplicação de algoritmos de ML e DL, contudo essa base de dados é opcional e somente necessária quando pretende-se utilizar as aplicações de previsão de demanda.

3.3.2 Diagramas de Atividade

A etapa de diagrama de atividade é um processo da UML utilizado na visualização de casos de uso individuais, sendo geralmente executado para obter-se uma visualização de alta abstração, focando-se em uma perspectiva orientada a processos [8].

A figura 3.8 apresenta o diagrama de atividades do caso de uso geral, visando identificar o fluxo de funcionamento das atividades e as ações executadas por cada ator no processo. Esse diagrama apresenta todos os fluxos presentes na aplicação, abrangendo todos os casos de uso citados anteriormente.

Como pode ser visto na figura 3.8, o fluxo do sistema não é linear, permitindo que o gestor possa realizar ações em qualquer ordem. Dessa forma, é possível executar apenas análises diárias, ou apenas prever demandas utilizando um único algoritmo de previsão, fornecendo ao gestor o controle de quais ações pretende executar a qualquer momento. Contudo, a ação de carregamento do banco de dados é realizada antes de iniciar a aplicação, e caso não seja feita, a aplicação não é iniciada e finaliza-se todo o ciclo de execução. Caso a base de dados seja carregada corretamente, o fluxo do sistema é executado normalmente.

Com isso, após a definição das etapas da pesquisa, a definição da estrutura de dados do projeto e a arquitetura e estruturação da solução final proposta, foi possível iniciar o processos de desenvolvimento.

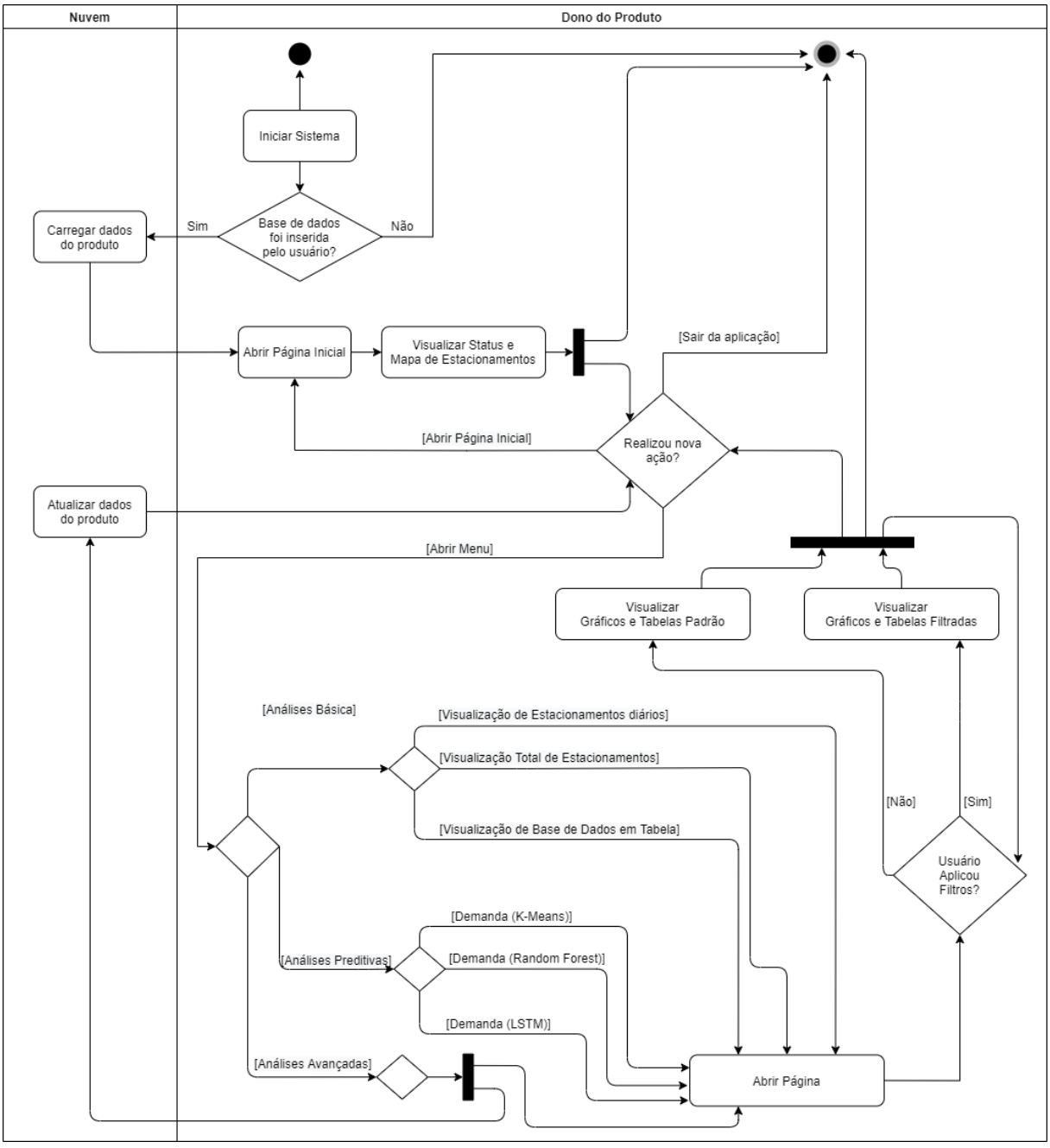


Figura 3.8: Diagrama de atividades da solução proposta.

Capítulo 4

Desenvolvimento

Neste capítulo é abordado todo o processo de desenvolvimento do trabalho, identificando cada passo em ordem cronológica de realização, desde a seleção das ferramentas até o desenvolvimento da plataforma de BI proposta como solução para os problemas identificados.

4.1 Ferramentas

Nessa seção são apresentadas as ferramentas, linguagens de programação e bibliotecas utilizadas para o desenvolvimento da pesquisa. A escolha das ferramentas foi feita no início do projeto, baseando-se em ferramentas amplamente utilizadas, com grande número de produções de código aberto e que garantam uma fácil manutenção futura das soluções desenvolvidas.

A linguagem de programação utilizada em todas as etapas foi o Python, que surgiu nos anos 80, sendo uma linguagem de proposito geral, a qual permite o desenvolvimento de qualquer tipo de algoritmo. A escolha pela linguagem se deu pela sua extensa versatilidade e facilidade de uso. A versatilidade permitiu o reaproveitamento de diversos códigos e algoritmos desenvolvidos durante as várias etapas de desenvolvimento, integrando-os de forma simples na solução final proposta.

Além disso, o Python conta com um robusto número de bibliotecas focadas na área de

ciência de dados e Machine Learning, sendo a linguagem líder em número de aplicações nesse meio, como afirma Raschka: "Python continua a ser a linguagem preferida para computação científica, ciência de dados e aprendizado de máquina, aumentando o desempenho e a produtividade ao permitir o uso de bibliotecas de baixo nível e APIs de alto nível limpas"[72].

Em uma pesquisa realizada em 2018 [1], pela Kaggle, uma comunidade focada no contexto de IA, relatou que a linguagem Python era a mais popular, sendo utilizada por 83% dos desenvolvedores da comunidade, enquanto que a linguagem de programação mais próxima era o R, em terceiro lugar, utilizado por 36% dos entrevistados, logo atrás de SQL, segundo lugar, utilizada por 44% dos usuários. O domínio da linguagem nesse contexto, possibilita que aplicações desenvolvidas com essa tecnologia possuam uma manutenção mais duradoura, devido à sua popularidade e vasto número de usuários. Com isso, a probabilidade de escalar aplicações de código aberto focadas em ciência de dados e IA é maior, quando comparado com linguagens menos populares.

A seguir serão apresentadas as principais bibliotecas do Python que foram utilizadas durante o progresso do trabalho, sendo exploradas suas principais funcionalidades e as vantagens que as tornam essenciais para as etapas do desenvolvimento.

4.1.1 Pandas

A biblioteca Pandas é a segunda biblioteca mais utilizada para manipulação e análise de dados no Python [50] . A funcionalidade mais importante da biblioteca está na introdução de DataFrames [1], que são estruturas de dados próprias em um formato de duas dimensões, muito semelhantes às planilhas, contendo linhas e colunas. Os DataFrames permitem realizar dezenas de operações diretamente em suas tabelas, como soma, média e contagem, tornando muito simples a manipulação de dados.

Além disso, é possível realizar o processamento e filtragem de dados, desde a manipulação de colunas e linhas, até a remoção de valores inválidos ou que não cumpram uma

[1]https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

determinada regra pré-estabelecida como, por exemplo, a busca por uma data específica na tabela de dados.

4.1.2 Matplotlib

O MatPlotlib é uma biblioteca para produção de gráficos, sendo definida em seu site oficial como: "uma biblioteca abrangente para a criação de visualizações estáticas, animadas e interativas em Python"[4]. Essa biblioteca possuí um vasto conjunto de gráficos disponibilizados de forma simples e intuitiva, que tornam mais acessível a criação de gráficos.

A biblioteca ainda conta com uma ampla compatibilidade com o Pandas, e, quando usada em conjunto, permite a criação de gráficos como dispersão, barras, linhas e histogramas, baseados em dados das tabelas (DataFrames) processadas pelo Pandas. Dessa forma, é possível realizar a criação de gráficos complexos e robustos advindos de uma fonte de dados já previamente processada e analisada.

No estudo [84] é apresentado a importância de visualizações gráficas de dados, e considera-se que os gráficos são a forma mais influente usada na comunicação científica, sendo essenciais na tomada de decisão da aceitação de um manuscrito científico. Por esse motivo, a plataforma de BI desenvolvida ao fim do projeto, utiliza-se de gráficos na maioria de suas funcionalidades, abstraindo bases de dados massivas em um formato visual simples e agradável. Possibilitando ao usuário visualizar o desempenho do seu estacionamento e identificar os desvios de padrão que ocorrem no processo.

4.1.3 Scikit-Learn

O Scikit-Learn (SKLearn) é uma biblioteca gratuita focada em aplicações de algoritmos de IA, amplamente utilizada no meio de desenvolvimento Python, sendo a biblioteca mais comum de IA nesse contexto. A biblioteca abrange dezenas de algoritmos e funcionalidades, como K-Means[1] e Random Forest[2], que tornam mais simples e intuitivo a aplicação de abordagens baseadas em IA.

[1]https://scikit-learn.org/stable/modules/clustering.htmlk-means
[2]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

No contexto do projeto, o SKLearn foi essencial para as aplicações de algoritmos de ML nos dados de Smart Parkings, sendo utilizada para a aplicação de K-Means e Random Forest na previsão de demanda futura. Seu uso permitiu a aplicação dos algoritmos de previsão de forma simples.

Vale ressaltar que o SKLearn tem seu principal foco em aplicações de algoritmos de ML, mesmo que possua funcionalidades para aplicações utilizando DL, esse não é o ponto forte dessa biblioteca. Por esse motivo, para aplicações de DL foi utilizada a biblioteca Keras, descrita de seguida, mais apropriada para DL.

4.1.4 Keras

O Keras é uma API de alto nível para DL, desenvolvida sobre a biblioteca TensorFlow, com o intuito de permitir a modelagem e execução de redes neurais (NN) de forma simplificada [14]. Seu funcionamento baseia-se em abstrair a complexidade de NN por meio da disponibilização de algoritmos pré-estabelecidos de DL, como LSTM e GRU.

Dessa maneira, a estrutura, métricas e modelos de otimização aplicados aos algoritmos de NN já estão previamente desenvolvidos, cabendo ao usuário, identificar e preparar a estrutura de dados na qual será executada o algoritmo. Contudo, caso o usuário queira, antes de iniciar o treinamento da NN é possível aplicar métricas e otimizações personalizadas no modelo. A precisão dos resultados está ligada à quantidade de treinos, sendo em geral incrementada enquanto maior o número de etapas de treinamento [92].

4.1.5 Dash

O Dash é um framework Python criado no topo de diferentes bibliotecas, como Flask, Plotly e React, focado na construção de aplicativos web [62]. A união dessas bibliotecas garantem ao Dash uma estrutura completa, trazendo ao desenvolvedor uma estrutura de sistema totalmente programada em Python, que integra um servidor para execução de algoritmos e uma interface web cliente que permite usuários acessá-lo por meio da internet.

Como algoritmos e aplicações desenvolvidas no decorrer do trabalho estão todos em Python, a escolha por um sistema web compatível com a linguagem permitiu a integração das soluções de forma rápida e eficiente. Assim, o Dash permitiu executar um servidor Python, que executa algoritmos de mineração, processamento, análise e filtragem de dados, além de algoritmos de ML e DL. Fornecendo também uma interface web, conectada a esse servidor, a qual permite visualizar os dados de forma intuitiva e de fácil consumo para gestores do sistema Smart Parking.

Um grande diferencial do Dash está em sua interface de cliente, pois ele permite a criação de componentes de interface web, como botões, menus e campos de texto utilizando Python. Quando a interface é compilada, converte-se a solução do cliente para um modelo React baseado em JavaScript, que executa na grande maioria dos navegadores de internet, incluindo dispositivos móveis. Isso torna as soluções desenvolvidas em Dash mais amplas a compatíveis com os dispositivos atuais. Além disso, os componentes de interface são manipuláveis e permitem que o usuário execute ações como filtragem e seleção de dados, trazendo um aspecto mais dinâmico ao sistema.

4.1.6 Plotly

O Plotly é uma biblioteca para criação de visualizações de dados focada em ML e DL, possuindo como diferencial, a apresentação de gráficos em modelos interativos e compatíveis com aplicações web. O foco em usabilidade e interface torna o Plotly uma ótima opção para aplicações desenvolvidas em Python, por esse motivo, é a biblioteca padrão de gráficos utilizada pelo framework Dash.

O seu uso é recomendado para aplicações que pretendem fornecer uma interface com visualizações gráficas que interajam com as ações do usuário. A biblioteca disponibiliza gráficos que permitem funcionalidades como zoom, recorte de área, download de gráficos e legendas interativas para cada informação.

A biblioteca assemelha-se com o Matplotlib em seu quesito de finalidade, contudo, enquanto o Matplotlib é mais simples de usar, sendo utilizada no desenvolver do projeto

em visualizações rápidas e menos elaboradas, focado no ganho de produtividade durante o desenvolvimento. O Plotly é focado em sua interface de interação de usuário, necessitando de mais tempo para aprendizado e um desenvolvimento mais elaborado. Por esse motivo, o Plotly é utilizado como a biblioteca padrão para representações gráficas na aplicação final, pois a mesma deve garantir aos usuários interfaces interativas e agradáveis, focadas em usabilidade e robustez, agregando assim, uma maior gama de recursos ao produto final.

4.2 Mineração de Dados

O processo de mineração de dados como explicada na seção 2.2.1, é um processo essencial para adquirir, preparar, processar e analisar os dados que serão utilizados pelo sistema [28]. As etapas do processo de mineração de dados aplicado no trabalho foram separadas nas seguintes atividades:

1. Extração de dados: essa etapa visa realizar a descoberta de conhecimento, sendo a etapa inicial do processo. Nessa etapa são focados aspectos desde a obtenção e preparação dos dados, até a visualização e descoberta de conhecimentos sobre a informação.

2. Simulação de dados: trata-se de uma etapa adicional realizada no processo, que foi necessária devido às conclusões obtidas na etapa de extração de dados. Essa etapa apresenta uma solução para simular dados e permitir a criação de bases de dados para aplicações de algoritmos de ML e DL.

3. Análise de dados: nessa etapa é realizado a análise de dados por meio de algoritmos de ML e DL com o intuito de obter previsões futuras a partir dos dados de estacionamentos. Para isso, são aplicados diferentes algoritmos e é realizado uma comparação entre seus resultados.

4.2.1 Extração de Dados

Essa etapa foi iniciada após a definição da estrutura de dados que serviria de base para o sistema Smart Parking e a escolha das ferramentas. Após esses processos, foi possível identificar quais tipos de dados devem ser extraídos, focando em suas variáveis e semelhanças ao modelo proposto na seção 3.2.2.3.

O processo de extração de dados foi focado em Knowledge Discovery in Databases (KDD), em português descoberta de conhecimentos em bases de dados. Esse processo tem por objetivo identificar padrões e características de interesse em grandes bases de dados, sendo que a mineração de dados faz parte do processo [47].

Considerando que o produto Smart Parking ainda estava em desenvolvimento, não haviam dados reais sendo produzidos, visto que trata-se de um protótipo que não foi aplicado em um ambiente real de estacionamento. Essa inexistência de dados próprios do sistema, pôde ser inicialmente suprida por meio de um levantamento e mineração de dados de estacionamentos presentes na internet. Esse método foi aplicado a fim de dispor de bases de dados reais que pudessem ser utilizadas para o teste e validação dos algoritmos desenvolvidos no decorrer do trabalho.

A extração pode ser feita por sistemas que obtém dados de diversas fontes e os agregam em uma única estrutura realizando processos de carregamento, extração e transformação [38]. Além disso, essas etapas podem ser feitas sem o auxílio de um sistema único, de forma individual. Esse processo é importante pois os resultados das análises de BI estão ligados ao desempenho e qualidade do processo de extração de dados [67].

A metodologia de extração de dados utilizada no desenvolvimento tratou-se de um processo mais simplificado, no qual as etapas foram feitas de forma individual e com diferentes ferramentas, citadas na seção 4.1. Essa escolha se deu pela baixa quantidade de dados que preenchiam os requisitos do negócio. Isso ocorreu devido à falta de dados reais do produto e limitação das bases de dados disponíveis na internet, as quais são limitadas devido ao tipo de dado ser muito específico (registros de estacionamentos).

4.2.1.1 Obtenção

Essa etapa tem por objetivo adquirir dados que contenham informações relevantes para o negócio, sendo focada na compreensão do negócio e dos dados [56]. Os dados obtidos devem conter variáveis importantes para a descoberta de conhecimento e compatíveis com o negócio, sendo exclusivamente dados de estacionamentos.

A etapa iniciou-se com a busca por bases de dados públicas disponíveis na internet, que são geralmente disponibilizadas por entidades governamentais. Esse tipo de dado é denominado *open data*, em português dados abertos, e refere-se a dados coletados e compartilhados publicamente para terceiros, sem restrição de uso ou direitos legais [16].

O uso de dados abertos é uma realidade que cresceu nos últimos anos, principalmente devido à grande produção de dados que são gerados em todos os meios de produção da sociedade, tanto por pessoas, como organizações e equipamentos inteligentes. Os benefícios desse tipo de informação abrangem características como: maior economia de tempo e dinheiro com a utilização de dados existentes e evita replicação de dados. Além disso, com uma maior quantidade de dados disponíveis, a tomada de decisão de organizações pode ser aprimorada [16].

Em [17] é destacada a importância de dados abertos. O autor afirma que o compartilhamento de dados abertos de forma bruta e limpa realizado por jornalistas, foi um fator fundamental para a quebra do monopólio de interpretação de dados que o governo possuí. Identificando que esse compartilhamento de dados brutos dá ao usuário a possibilidade de visualizar e interpretar a informação à sua maneira, sem qualquer tipo de direcionamento pré-estabelecido pela entidade que detém a informação. Além disso, o uso de dados abertos é uma iniciativa adotada por cidades inteligentes, tendo maior impacto nas áreas de governança, economia e transporte e mobilidade [63].

A busca por dados foi realizada em diversas bases disponíveis na internet, tanto em plataformas de dados como a Kaggle[1] , que abrange uma extensa e diversificada base de dados de *datasets* (conjuntos de informação) dos mais diversos campos do conhecimento,

[1]https://www.kaggle.com/datasets

como em sites governamentais, de diferentes países e cidades do mundo.

Um fator fundamental durante a escolha do dataset foi o seu formato de exportação, que define a dificuldade para manuseio em que encontram-se os dados. O formato que o trabalho pautou-se foi Comma-separated values (CSV), que trata-se de um formato simples no qual os dados são separados, geralmente, por vírgula e linhas, produzindo uma tabela de duas dimensões. A escolha por esse formato se deu por tratar-se de um formato universal e de fácil manipulação, o qual pode ser acessado pela grande maioria dos softwares de análise e visualização de dados, além de bibliotecas de programação como Pandas. Dessa maneira, dados exibidos em formatos menos acessíveis, como PDF e imagem, foram descartados, pois além de serem incomuns encontrar, quando comparados ao formato CSV, exigem métodos complexos para convertê-los em bases de dados legíveis por softwares de dados.

Outro fator que pautou a escolha de datasets foi a similaridade com o conceito estrutural proposto na etapa de integração de dados. Devido a isso, foi identificado que a grande maioria dos datasets encontrados não conseguiam preencher algumas regras de negócio do Smart Parking proposto. Assim, alguns datasets possuíam variáveis relevantes para uma etapa do projeto e outros para outras etapas, onde alguns continham apenas informação de entrada e saída, enquanto outros apenas a taxa de ocupação do estacionamento por hora. As buscas foram realizadas em sites governamentais e em bases de datasets. Contudo, não foi encontrado nenhum dataset que possuísse dados de estacionamento juntamente com as variáveis necessárias para o sistema, como o clima registrado no momento do estacionamento.

Com isso, foi considerado o uso de um dataset com o maior número de dados possível, que contivesse ao menos o registro de horário de entrada e saída de cada estacionamento realizado. Isso se deu pois como citado na seção 2.2.1, o tamanho da base de dados é um fator fundamental na aplicação de análises. Com essas informações, ainda que de forma limitada, já seria possível executar algoritmos para previsão de demanda, que futuramente, poderiam ser incrementados com variáveis como o clima e feriados.

Dessa maneira, o dataset selecionado para servir de base no desenvolvimento inicial

do trabalho foi o dataset de transportes [40] disponibilizado pelo ACT Government Open Data Portal, plataforma de disponibilização de dados da capital do território da Austrália. O dataset trata-se dos registros de um sistema de Smart Parking, chamado ParkCBR, que foi implementado na cidade, especificamente na região comercial de Manuka. Esse sistema visa reduzir o tráfego da região com o uso de dados em tempo real do sensor da baia, que indica para os motoristas as regiões com maior número de vagas disponíveis.

Em [95] o autor utiliza-se do mesmo dataset para identificar características de fluxo da área do estacionamento. Além disso, o autor identifica que existem faixas de horários com maior fluxo e que feriados possuem fluxo diferenciado do padrão, sendo em geral mais elevados. Para o presente trabalho, o uso do dataset foi aplicado na identificação dessas catarísticas e outras mais específicas, como por exemplo: identificar a média de estacionamentos diária, semanal, por horário e dias da semana, realizando agrupamentos de dados.

A tabela 4.1 apresenta detalhes do dataset utilizado, identificando seus atributos.

Variável	**Tipo**	**Descrição**
SectorName	Numérico	Tipo do setor (on/off street).
SectorCode	Texto	Código do setor.
LotName	Texto	Identificação do lote de estacionamento.
BayNumber	Numérico	Identificador da vaga.
BayName	Texto	Identificador secundário da vaga.
Arrived	Data e Hora	Data e Hora de início do estacionamento.
Departed	Data e Hora	Data e Hora do fim do estacionamento.
Street	Texto	Identificação da rua.

Tabela 4.1: Estrutura de dataset de estacionamentos público (adatptado de ACT [40]).

Como pode ser visto na tabela 4.1, o dataset não possui o registro de características como o clima, contudo, ele possuí algumas variáveis muito semelhantes ao modelo estrutural proposto, sendo as principais: *Arrived* e *Departed*. Essas variáveis permitem realizar análises de fluxo do estacionamento, tanto para dias, quanto para meses e anos. Além disso campos como *SectorName* permitem filtrar análises para onStreet e offStreet,

e *LotName* permite analisar o comportamento do estacionamento por regiões específicas.

Assim, considerando as semelhanças estruturais do dataset com o modelo proposto e sua quantidade de dados, ele foi definido como base para os levantamentos e algoritmos iniciais da pesquisa, sendo necessário após isso, realizar a preparação dos dados.

4.2.1.2 Preparação dos dados

A etapa de preparação, também conhecida como pré-processamento, é primordial no processo de DM, pois é nela que realiza-se a limpeza dos dados. Nessa etapa, os dados brutos encontram-se, em geral, desorganizados e incompletos, e passam por diversas etapas que têm por função limpar e organizar os dados, para após isso, tornarem-se dados compreensíveis para análise [7].

Um dos passos iniciais para começar a preparação dos dados é identificar o tipo de cada variável. Os autores [65] [71] dividem os tipos de variáveis em:

- Categórica: variáveis que representam rótulos que são bem definidos e servem para identificar um grupo como, por exemplo, sexo, profissão e cor dos cabelo.
- Quantitativa: variáveis desse tipo expressam valores, em sua maioria numéricos, como altura, peso, idade, salário.

A grande maioria dos algoritmos baseados em métodos estatísticos tem por foco dados quantitativos enquanto que, para dados categóricos existem em menor quantidade [65]. As variáveis categóricas permitem a aplicação de filtros a fim de obter características de um dado grupo, como por exemplo, obter apenas uma região de estacionamento do dataset para realizar uma análise em um grupo específico.

A principal característica buscada nos dados foi identificar o padrão de fluxo dos estacionamentos. Para isso, as variáveis categóricas *Street* e *SectorName* foram utilizadas para fragmentar o dataset em vários datasets menores, separados por tipo de estacionamento e região.

Tendo em vista que o dataset registra a data e hora nas mesmas colunas, realizou-se a separação desses dados em duas colunas distintas, permitindo dessa maneira, que fosse

possível realizar agrupamentos diversificados, como agrupamentos apenas por data ou hora semelhantes.

O agrupamento de dados foi essencial para permitir análises baseadas em faixas temporais curtas, médias e longas, pois a partir disso, criou-se uma nova coluna, quantitativa, denominada *TotalParkings* que contém a quantidade de estacionamentos que foram agrupados. Assim, ao agrupar estacionamentos por data e hora, obteve-se o total de estacionamentos realizados por hora em toda a faixa temporal do dataset. Além disso, caso o agrupamento seja feito por data, obtém-se o total de estacionamentos realizados por dia durante toda a faixa temporal do dataset.

Um importante fator a ser considerado na etapa de processamento é o tratamento de valores vazios e nulos, que sempre estão presentes em grandes volumes de dados, sendo muitas vezes causado pela falta de informações do dataset. Existem diferentes formas de tratar dados desse tipo, contudo, para esse trabalho, dados inexistentes foram descartados. Essa medida foi escolhida pois o dataset manteve-se muito extenso, mesmo após a remoção de dados inválidos.

4.2.1.3 Visualização

A etapa de visualização de dados teve por objetivo identificar padrões e informações relevantes acerca de estacionamentos a partir de agrupamentos e visualizações de dados. Dessa maneira, essa etapa focou-se na descoberta de formatos de exibição e agrupamentos que pudessem agregar valor ao produto, principalmente quando o retorno permite auxiliar na tomada de decisão dos gestores. Essas informações fazem parte do processo de KDD, citados na seção 4.2.1 e seus resultados refletem no processo de BI citado na seção 2.3.

Com o foco da solução em BI, o levantamento de questões de negócio e a descoberta de formas que possam solucioná-las, principalmente utilizando gráficos de fácil interpretação, são fundamentais para facilitar as análises [68]. Por isso, diferentes perguntas foram elencadas e vários testes foram realizados, muitas vezes de forma experimental, para identificar quais elementos deviam estar presentes e quais tipos de gráficos eram mais relevantes para responder cada pergunta.

As perguntas de negócio levantadas devem ser interessantes para os gestores, principalmente no auxilio da tomada de decisões à cerca de características que afetam o estacionamento [66] [12] [95], como por exemplo:

- Como traçar o perfil do estacionamento?
- Como identificar padrões de fluxo no local?
- Como identificar o comportamento do local em dias chuvosos?
- Em finais de semana ou feriados a demanda é reduzida ou incrementada?

Uma informação básica que deve ser questionada é a identificação dos horários de pico do estacionamento, que pode ser realizada com o agrupamento de estacionamentos por hora e a aplicação de uma média sobre o resultado. Essa informação permite por exemplo, identificar para os usuários do estacionamento os horários de maior ocupação do local, que por consequência terão maior fluxo e congestionamento.

A figura 4.1 apresenta a média de estacionamentos realizados em um estacionamento por faixa horária.

A importância de diferentes visualizações de um mesmo dado, incluindo diferentes características no gráfico e permitindo que filtros possam ser utilizados, podem tornar as curvas e pontos do gráfico mais característicos, permitindo um ganho relevante de informação [88] . As figuras 4.2 e 4.3 exibem a visualização de estacionamentos realizados por horário de entrada e saída. Ambas possuem o mesmo tipo de gráfico e são aplicadas no mesmo conjunto de dados, contudo, a primeira exibe os dados para todo o conjunto, enquanto que a segunda os filtra por apenas uma faixa horária de entrada. Com isso, é possível identificar de forma clara que com uma visualização mais próxima e em um escopo reduzido (filtrado), as características são mais evidentes.

O agrupamento de estacionamentos permite que a visualização seja muito mais simples e destaca o perfil do estacionamento. Dessa maneira, é possível aplicar esse agrupamento de forma cada vez mais ampla, a fim de identificar aspectos diários, semanais, mensais e

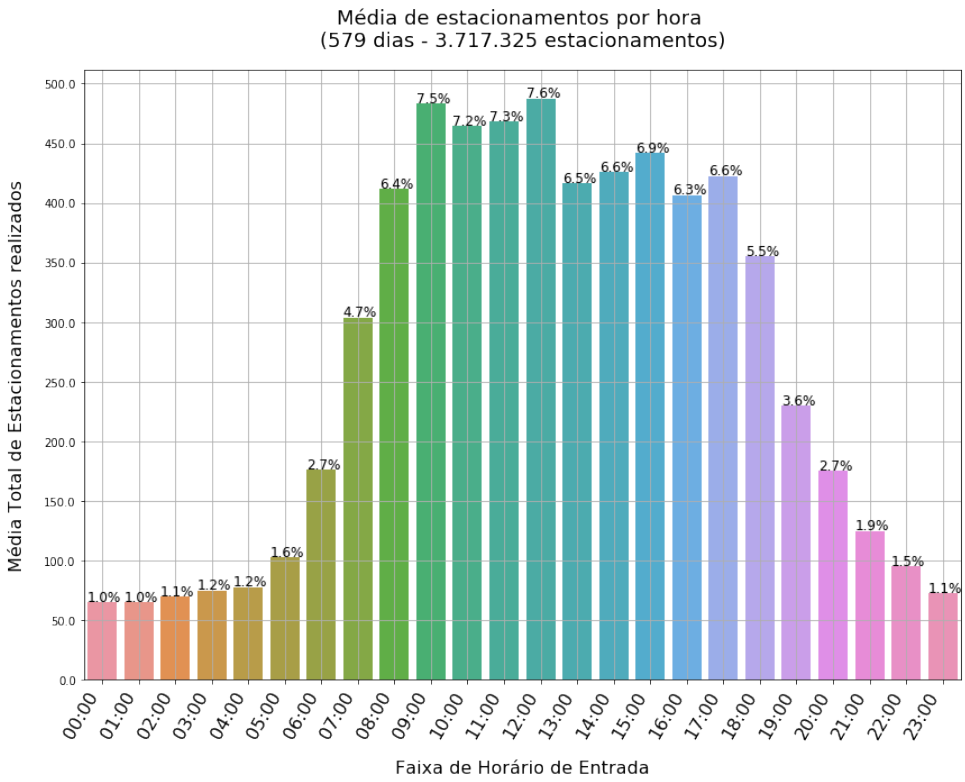


Figura 4.1: Média de estacionamentos realizados por faixa de horário.

até mesmo anuais. A figura 4.4 apresenta a visualização de dados agrupados por hora e a figura 4.5 agrupados por dia.

Na figura 4.4 é possível visualizar que as horas seguem uma tendência padrão crescente e decrescente em todos os dias, enquanto que na figura 4.5 verifica-se que os meses definem o limite máximo e mínimo dessa tendência diária, havendo meses com maior fluxo e outros com menor fluxo. Além disso, na figura 4.5, a aplicação de um filtro de legenda para identificar finais de semana demonstra que os menores fluxos de cada semana ocorrem justamente nos finais de semana. Esse tipo de característica auxilia na identificação do perfil do estacionamento, pois identifica o padrão de fluxo semanal do mesmo.

Considerando que as regras de negócio aplicadas ao produto devem verificar o comportamento para finais de semana, clima e feriados, e sabendo que o dataset não conta com dados sobre feriados e clima, foi possível apenas testar o comportamento em finais de semana. Para isso utilizou-se todo o grupo de dados filtrados por dias da semana,

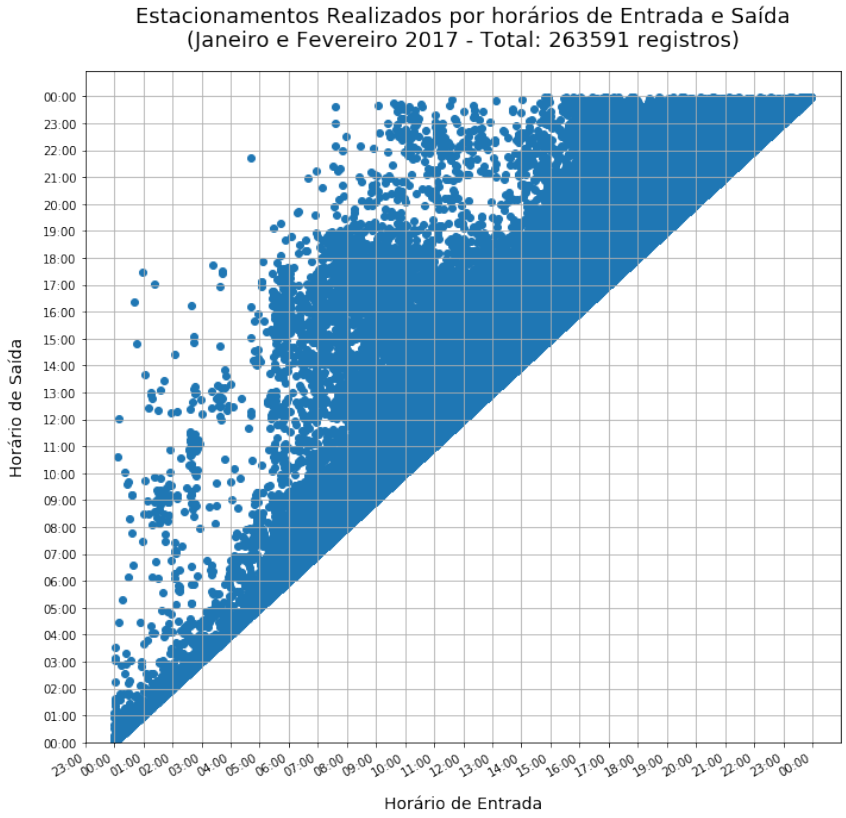


Figura 4.2: Estacionamentos realizados por horário de entrada e saída (sem filtro).

obtendo-se assim uma média dos estacionamentos realizados para cada dia da semana, que pode ser vista na figura 4.6. Assim como na figura 4.5, na figura 4.6 é possível identificar que o fluxo em finais de semana tem uma redução quando comparado aos dias da semana, sendo essa diferença detalhada em valores percentuais.

4.2.1.4 Avaliação e Resultados

A etapa de avaliação consistiu em identificar a qualidade das análises desenvolvidas. Levando em conta que o foco é BI, procurou-se identificar quais benefícios os agrupamentos realizados e as visualizações produzidas podem auxiliar em cada ponto do produto.

Com isso, na figura 4.1 é possível visualizar de forma acessível que o perfil do estacionamento é, por exemplo, voltado para horários comerciais, sendo seu fluxo de entrada muito elevado nesses horários. Além disso, durante 1/4 do dia, no período da madrugada,

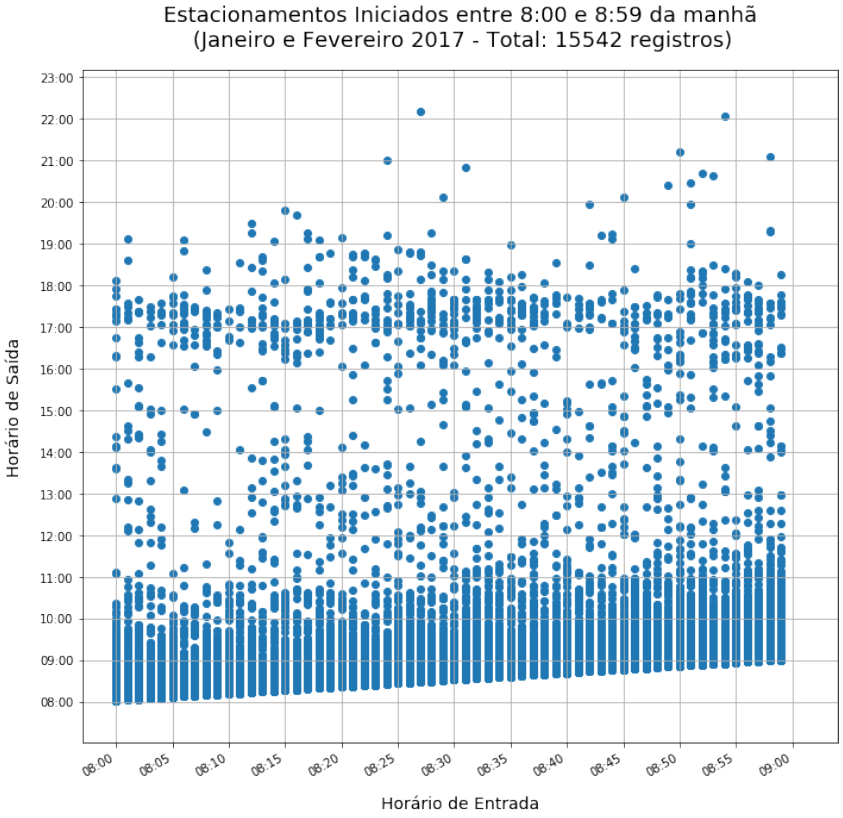


Figura 4.3: Estacionamentos realizados por horário de entrada e saída (com filtro).

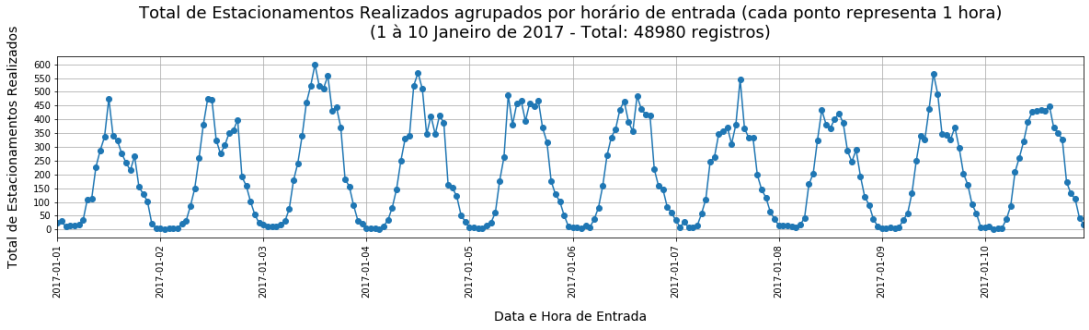


Figura 4.4: Total de estacionamentos realizados por horário de entrada (10 dias).

a soma de estacionamentos realizados não chega ao total de estacionamentos realizados em 1 hora de um horário de pico, como 12:00. Essa informação auxilia os gestores na identificação de horários de maior fluxo, no qual podem, por exemplo, aumentar o custo do estacionamento com tarifação dinâmica.

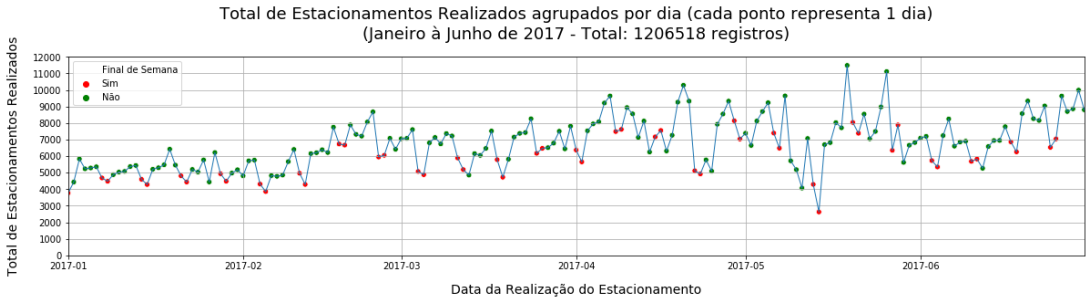


Figura 4.5: Total de estacionamentos realizados por dia (6 meses).

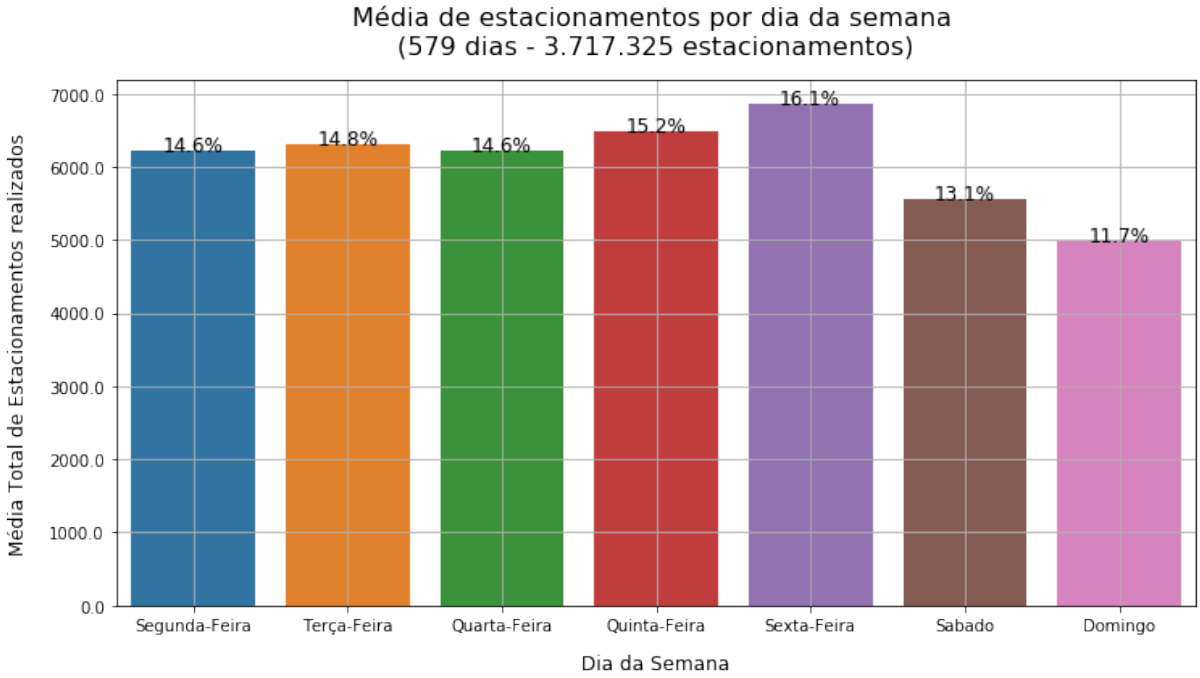


Figura 4.6: Média de estacionamentos realizados por dia da semana.

A figura 4.2 permite uma visualização ampla para controle de entrada e saída nos estacionamentos, contudo, devido à quantidade de registros, é difícil identificar padrões nesse meio. A figura 4.3 trata-se de um zoom ou filtro realizado nos mesmos dados da figura 4.2. Ao reduzir o escopo de registros, o gestor pode visualizar com clareza que indivíduos que entram no estacionamento na faixa das 8h às 9h da manhã, tendem a trabalhar até o horário de almoço, saindo meio dia, ou ficando até o horário de fechamento do comércio local entre 17h e 18h. Esse tipo de informação ajuda a segmentar o perfil

de usuários do estacionamento, além de mostrar o quão importante é entregar ao gestor gráficos dinâmicos que permitam funcionalidades de filtragem e zoom.

A figura 4.4 demonstra que é possível identificar um padrão para estacionamentos realizados quando agrupados por horário de entrada. Essa informação é muito importante para a aplicação de algoritmos de IA com foco na previsão de demanda, visto que padrões tornam as previsões mais precisas. A figura 4.5 demostra que a variação do total de estacionamentos realizados em um longo período de tempo, mantém um padrão semanal que eleva e decai, e com o passar dos meses, torna-se em média maior ou menor, o que ajuda a identificar o perfil do estacionamento. Essa informação é bastante relevante pois saber o tráfego médio de cada dia da semana e a variação média em cada mês do ano, permite ao gestor ter um controle melhor na previsão de demanda e previsão de rentabilidade do estacionamento para dias ou meses futuros.

A figura 4.6, ao apresentar a média de estacionamentos por dia da semana, mostra que o perfil do estacionamento pode ser, por exemplo, de área comercial, pois tem maior movimento em dias úteis no qual o comércio funciona sem restrições. Essa informação é relevante para, por exemplo, aplicação de tarifação dinâmica, que possa reduzir o preço do estacionamento aos finais de semana, visando incentivar usuários à utilizarem mais o estacionamentos nesse período e por consequência aumentar a rentabilidade.

Por fim, a avaliação das análises foi essencial para identificar que, mesmo respondendo a diversas perguntas relevantes de negócio, não era possível, utilizando dados de terceiros, alcançar todas as regras de negócio estipuladas no produto. Com isso, após a avaliação, foi necessário o desenvolvimento de um simulador de dados, que pudesse gerar dados de estacionamentos com padrões semelhantes ao de um dataset real. Além disso, a simulação teria de abranger todas as variáveis necessárias para o sistema, principalmente o clima e feriados. Dessa maneira, considerou-se que apenas com um dataset simulado seria possível a aplicação de análises que não foram possíveis com o uso de dados públicos.

4.2.2 Simulação de Dados

Como explicado anteriormente, para avançar o contexto do trabalho e os requisitos para as aplicações de ML e DL, houve a necessidade de um grande dataset de estacionamentos, que fosse extenso em volume de dados e principalmente, que abrangesse todas as variáveis do produto. Nesse contexto estão inclusas variáveis importantes utilizadas no dataset da seção 4.2.1.2, como o nome do estacionamento e o horário de entrada e saída, informações utilizadas para o agrupamento e visualização dos dados.

A simulação de dados pode ser um processo importante para o treinamento de algoritmos que precisam de grandes volumes de dados dos quais muitas vezes não possuímos [37]. Por esse motivo, foi desenvolvida uma ferramenta que pudesse simular dados de estacionamento, que conseguisse produzir padrões similares aos reais.

A ferramenta foi desenvolvida em Python e contou com 5 etapas de desenvolvimento, a de especificação (1), estruturação (2) desenvolvimento (3), resultados (4) e validação (5).

4.2.2.1 Especificação do sistema

Inicialmente foi necessário identificar quais os tipos de dados relacionados com estacionamentos, que como definido na seção 2.2.2 pode caracterizar-se como dados de séries temporais. Essa informação pode ser reforçada pelos resultados da seção 4.2.1.3, na qual horários, dias e meses são variáveis que podem aumentar ou reduzir o fluxo do estacionamento.

O simulador deve produzir dados de forma a assemelhar-se à um padrão real de estacionamento, seguindo características de série temporal. Assim, deve-se considerar que estacionamentos possuem diferentes fluxos de movimento baseados em variáveis temporais comuns como dias da semana, horários do dia, meses do ano. Além disso, foi considerado também, variáveis específicas do negócio, citadas na seção 3.2.2.3, como simular a alteração de fluxo baseado no clima atual do local e simular um fluxo diferenciado para feriados e finais de semana.

Com foco na flexibilidade da simulação, o simulador teve de permitir o ajuste de variáveis, por esse motivo, o simulador dispõe de pesos percentuais que podem ser alterados para cada variável. Dessa maneira, torna-se possível gerar dados para os mais variados tipos e padrões de estacionamentos como universidades, restaurantes, parques de diversões, entre outros.

Levando em consideração as diretrizes para Smart Parking, especificamente o padrão Fiware que é utilizado no produto, os estacionamentos podem conter diferentes grupos, como citado na seção 3.2.2.1. Com isso, o sistema foi estruturado para gerar dados de forma escalável para estacionamentos com qualquer quantidade de regiões (grupos), assim é possível em uma única execução, gerar dados para N estacionamentos contendo M regiões em cada estacionamento. Além disso, cada estacionamento contém variáveis abrangentes como o clima aplicável para aquele local, e cada região contém variáveis locais, como tempo máximo de permanência e capacidade máxima do estacionamento.

4.2.2.2 Estruturação

A estrutura a ser simulada baseou-se em dois grandes níveis, o de estacionamentos (Parking) e o de grupos (Regions). A seguir serão descritas as estruturas de simulação:

- Parking: trata-se da estrutura ampla, com informações como o nome do estacionamento e clima local, sendo baseadas na estrutura de Parking da figura 3.3 .As variáveis do Parking são compartilhadas por todas suas regiões, o que permite por exemplo, filtrar os dados por estacionamento para análises mais generalistas. A tabela 4.2 apresenta a estrutura de um parking;

- Regions: trata-se da estrutura presente dentro de um parking, como um setor ou piso do estacionamento, baseado na estrutura do ParkingGroup da figura 3.3. Cada região contém valores próprios que irão definir o fluxo de estacionamentos da região, permitindo assim, que um único estacionamento possua diversas regiões cada qual com um padrão de fluxo próprio. A tabela 4.3 apresenta a estrutura de uma região.

Variável	Tipo	Valor Padrão	Descrição	Exemplo de Entrada
Parking	Texto	-	Nome do estacionamento	IPB UTFPR
Weather	Texto	Vancouver	Nome da cidade que o estacionamento irá simular a condição climática	Vancouver, Portland, Montreal
Regions	Matriz	-	Contém dados de cada região do estacionamento	[{região1...}, {região2...}]

Tabela 4.2: Estrutura de dados do simulador (parking).

Cada variável presente nas estruturas tem uma determinada importância ao ser considerada na criação dos dados, podendo mudar de forma drástica o fluxo final do estacionamento. Essa flexibilidade das variáveis e a mudança que podem causar nos resultados é o que permite a criação de diversos padrões de estacionamentos em uma única simulação. Para suportar as variáveis de negócio descritas na seção 3.2.2.3, o simulador possui abordagens específicas para cada uma dessas variáveis, sendo elas:

Variação Climática: sabendo que a simulação climática trata-se de um problema muito complexo, para inserir variáveis climáticas com precisão real, o simulador utilizou-se de um dataset aberto, disponibilizado por David Beniaguev [19]. O dataset contém diversas cidades e armazena o tipo de clima registrado em cada cidade a cada hora, durante os anos de 2012 a 2017. Isso possibilitou o uso de climas reais nas simulações. Dessa forma, a simulação de horários de chuva e sol não seguem padrões aleatórios, cabendo ao usuário, escolher uma cidade no dataset que contenha mais ou menos registros de chuva. O nome da cidade é atribuído à variável *"Weather"* de Parking. Além disso, dentro de cada região é possível atribuir na variável *"Rainy_flow"* o comportamento que o estacionamento irá ter em horários de chuva, se o fluxo sofrerá aumento ou redução.

Variação para Feriados: esse tipo de variação é totalmente dependente do estacionamento local, por isso, cabe ao usuário definir quais são os dias "especiais"do ano para

Variável	Tipo	Descrição
Name	Texto	Nome da região de estacionamento
Open_time	Tempo	Horário de abertura da região
Close_time	Tempo	Horário de fechamento da região
Max_slots	Inteiro	Número máximo de vagas da região
Parking_time_ min	Inteiro (minutos)	Tempo mínimo de estacionamento
Parking_time_ average	Inteiro (minutos)	Tempo médio de estacionamento
Parking_time_ max	Inteiro (minutos)	Tempo máximo de estacionamento
Weekday_ flow	Número Flutuante	Variação do número de estacionamento para dias de semana
Weekend_ flow	Número Flutuante	Variação do número de estacionamento para finais de semana
Rainy_ flow	Número Flutuante	Variação do número de estacionamento para horários de chuva.
Year_increase_ flow_max	Número Flutuante	Crescimento máximo do número médio de estacionamentos por ano.
Year_decrease_ flow_max	Número Flutuante	Redução máxima do número médio de estacionamentos por ano
Rush_range	Matriz	Define faixas de horário que terão acréscimo ou redução de fluxo no estacionamento, podendo haver de 0 à N faixas de horário. Cada faixa contém seu fluxo próprio
Months_flow	Vetor	Define o fluxo médio de estacionamentos para cada mês do ano, podendo haver meses com mais ou menos estacionamentos em média, gerando assim uma sazonalidade.
Holidays_flow	Matriz	Define os feriados e dias especiais que afetarão o número de estacionamentos (flow) para aquela data. Podem ser inseridos de 0 à N dias especiais, cada qual com um fluxo próprio de estacionamento, podendo aumentar ou reduzir o número médio de estacionamentos para toda a data especificada.

Tabela 4.3: Estrutura de dados do simulador (region).

aquele local. Esses dias terão, inevitavelmente, um impacto diferenciado no estacionamento, sendo necessária a atribuição da variação percentual de fluxo para cada um desses dias. Esse dado compõe uma matriz e é atribuído à variável *"Holidays_flow"* de Region. A adição de variações para dias especiais torna a simulação mais dinâmica e permite aproximá-la ainda mais de dados reais.

Variação de dias úteis e finais de semana: considerando que alguns estacionamentos podem possuir um fluxo comum em dias úteis e finais de semana, é possível balancear de forma individual a variação de fluxo para cada um deles. Assim, é possível simular estacionamentos com alto fluxo em dias uteis e pouco fluxo em finais de semana, ou vice-versa, além de ser possível simular estacionamentos com um fluxo constante durante toda a semana. A variável que controla o fluxo para dias de semana é a *"Weekday_flow"* e para finais de semana é a *"Weekend_flow"*, sendo que ambas integram a estrutura de Region.

4.2.2.3 Desenvolvimento

Após a definição da estrutura, os algoritmos aplicados para o desenvolvimento da simulação basearam-se em laços de repetição que produziam dados para estacionamentos individuais, alocando-os em seus respectivos parking e region. Além da repetição, os dados gerados baseiam-se nas regras estipuladas para cada região e estacionamento, respeitando o horário máximo de permanência, horário de abertura e fechamento do estacionamento, limite máximo do local, entre outros.

A geração de dados inicia e finaliza com base em datas de início e fim estipuladas pelo usuário ao inicio da execução. Os registros de estacionamentos iniciam a partir da data de início e são gerados de forma gradual até que o último estacionamento alcance a data final. As datas iniciais e finais podem variar entre poucos dias ou muitos anos, dependendo da quantidade de dados que o usuário pretenda simular.

O simulador inicia obtendo uma estrutura de variáveis preenchidas que contém dados de ao menos 1 parking e 1 region, podendo haver qualquer quantidade acima desses valores. Essa estrutura é a base para que o algoritmo possa definir as variáveis sobre o processo

de simulação e possa gerar estacionamentos com o perfil que corresponde as variáveis de entrada.

A criação dos dados é feita por um processo cíclico que realiza a verificação de diversos pontos, acompanhando diversas regras lógicas. Essas regras lógicas definem quantos dados serão gerados em cada ciclo, sendo as principais:

- clima: verifica-se o clima atual correspondente à data e horário atual da simulação, após isso, aplica-se o coeficiente climático sobre o total de vagas.
- feriados: realiza-se uma verificação para identificar se a data é um feriado ou dia comum, aplicando o coeficiente de feriados sobre o total de vagas.
- final de semana: identifica-se qual o tipo de dia da data atual da simulação, aplicando-se o coeficiente de final de semana ou dia da semana sobre o total de vagas.
- horário de pico: o sistema verifica quais são os horários de pico do estacionamento e se o horário atual da simulação faz parte desse horário, caso faça, aplica-se o coeficiente do horário de pico específico sobre o total de vagas.

Depois que o dado passa por todas as regras, define-se quantos estacionamentos serão gerados nesse ciclo, podendo ser nenhum ou até dezenas. Os horários de início do estacionamento são definidos como a data e hora atual do simulador e o horário de fim é baseado em uma função aleatória que determina uma duração entre o mínimo e máximo, procurando, na maior parte dos casos, estar próximo da média. Quando um ciclo chega ao fim, adiciona-se uma quantidade aleatória de minutos entre 1 e 5 à data atual do simulador e um novo ciclo é iniciado. Assim os ciclos são executados até que a data do simulador seja maior do que a data final estipulada para simulação.

Dessa maneira, o desenvolvimento foi baseado em metodologias lógicas, que visam criar um padrão para o fluxo de estacionamentos, permitindo sua diferenciação em diferentes climas, dias da semana e horários. Além disso, durante todos os processos existem etapas aleatórias, que produzem ruídos nos dados a fim de tornar os padrões mais dinâmicos e inesperados. O uso de metodologias lógicas que geram padrões e a aleatóriedade aplicada

em determinadas funções, permitiu que o sistema pudesse gerar padrões lógicos e com uma variação indefinida, características que foram vistas nos datasets reais.

4.2.2.4 Testes e Resultados

Para demonstrar a abrangência e capacidade da simulação de dados foram simulados dois estacionamentos, que contém características únicas e fluxos diferenciados. Para isso, foram definidos valores distintos para as variáveis disponíveis no simulador, as quais definem o padrão de fluxo do estacionamento.

A tabela 4.4 apresenta características de cada um dos estacionamentos simulados e identificam como as variáveis serão aplicadas em cada um dos casos

Características	**Estacionamento 1 (E1)**	**Estacionamento 2 (E2)**
Nome	Restaurante Comunitário em região central (E1)	Parque de Diversões e Lazer
Horário de Funcionamento	Abertura: 10:30 Fechamento: 21:00	Abertura: 07:30 Fechamento: 22:00
Fluxo Diário	Maior nos horários entre: 12:00∼13:00 e 18:00∼19:00.	Maior nos horários entre: 08:30∼9:30 e 13:30∼14:30
Fluxo Semanal	Maior em dias de semana e menor em finais de semana	Menor em dias de semana e maior em finais de semana
Fluxo Mensal	Semelhante durante todo o ano	Pico nos meses de Dezembro e Abril
Fluxo em horários de Chuva	Aumento considerável	Queda considerável
Fluxo em feriados	Redução considerável	Aumento drástico

Tabela 4.4: Características de estacionamentos simulados.

A faixa de dados simulados foi de 4 anos, iniciando-se em 01/01/2013 e finalizando em 31/12/2016. Esses valores foram escolhidos pois a faixa temporal de 4 anos permite a criação de um volume suficiente de dados , além disso, o dataset de dados clmáticos citado na seção 4.2.2.2 possuí dados para toda essa faixa. Foi considerado que ambos estacionamentos encontram-se localizados na mesma cidade, ambos são estruturas do

tipo Region alocadas no mesmo Parking. Assim, a variação climática e os feriados anuais são os mesmos em ambos os casos.

A simulação de dados resultou em dois datasets, o primeiro referente ao $E1$ contendo um total de 925.047 registros e o segundo referente ao $E2$ contendo um total de 866.761 registros. As figuras 4.7 e 4.8 apresentam a visualização do dataset gerado para $E1$ e $E2$ respectivamente.

	parking	region	timeFrom	timeTo	spotWanted	spotWon	isWeekday	isRain	isHoliday
0	Porto	E1	01-01-2013 10:30	01-01-2013 11:25	164	164	1	0	1
1	Porto	E1	01-01-2013 10:36	01-01-2013 10:51	248	248	1	0	1
2	Porto	E1	01-01-2013 10:41	01-01-2013 11:08	91	91	1	0	1
3	Porto	E1	01-01-2013 10:46	01-01-2013 11:04	197	197	1	0	1
4	Porto	E1	01-01-2013 10:49	01-01-2013 11:25	187	187	1	0	1
...	...	...	...	...	...	...	...	...	...
925042	Porto	E1	31-12-2016 20:39	31-12-2016 20:47	174	174	0	1	0
925043	Porto	E1	31-12-2016 20:39	31-12-2016 20:49	0	0	0	1	0
925044	Porto	E1	31-12-2016 20:39	31-12-2016 20:45	185	85	0	1	0
925045	Porto	E1	31-12-2016 20:40	31-12-2016 20:58	193	193	0	1	0
925046	Porto	E1	31-12-2016 20:40	31-12-2016 20:59	178	178	0	1	0

925047 rows × 9 columns

Figura 4.7: Dataset gerado na simulação do E1.

	parking	region	timeFrom	timeTo	spotWanted	spotWon	isWeekday	isRain	isHoliday
0	Porto	E2	01-01-2013 07:30	01-01-2013 08:34	4	4	1	0	1
1	Porto	E2	01-01-2013 07:30	01-01-2013 07:39	58	58	1	0	1
2	Porto	E2	01-01-2013 07:30	01-01-2013 07:43	153	153	1	0	1
3	Porto	E2	01-01-2013 07:32	01-01-2013 07:41	106	106	1	0	1
4	Porto	E2	01-01-2013 07:32	01-01-2013 07:41	62	62	1	0	1
...	...	...	...	...	...	...	...	...	...
866756	Porto	E2	31-12-2016 19:38	31-12-2016 19:48	9	9	0	1	0
866757	Porto	E2	31-12-2016 19:38	31-12-2016 19:50	1	1	0	1	0
866758	Porto	E2	31-12-2016 19:38	31-12-2016 19:55	235	235	0	1	0
866759	Porto	E2	31-12-2016 19:42	31-12-2016 20:00	109	109	0	1	0
866760	Porto	E2	31-12-2016 19:42	31-12-2016 19:59	179	179	0	1	0

866761 rows × 9 columns

Figura 4.8: Dataset gerado na simulação do E2.

Para realizar a comparação entre ambos, visando apresentar a variação de cada estacionamento em características específicas como climática, feriados e finais de semana, foram desenvolvidos 4 gráficos, que representam respectivamente totais de estacionamento:

1. Agrupados por dia do ano.

2. Agrupados por horário, com linhas destacando a diferença entre dias comuns (1) e feriados (0).

3. Agrupados por horário, com linhas destacando a diferença entre dias de semana (1) e finais de semana (0).

4. Agrupados por horário, com linhas destacando a diferença entre dias chuvosos (1) e ensolarados (0).

As figuras 4.9 e 4.10 apresentam os conjuntos gráficos de visualização dos estacionamentos simulados, representando $E1$ e $E2$ respectivamente. Os padrões de horários identificados nos gráficos têm maior fluxo nos horários definidos em suas respectivas configurações, na faixa das 12h em $E1$ e na faixa das 9h em $E2$. Dessa maneira, seguem as características individuais descritas na tabela 4.4. Além disso, é possível identificar também que, para o $E1$, dias comuns, dias de semana e dias de sol tem em média maior fluxo, enquanto que, para $E2$ é exatamente o contrário.

Os registros gerados para ambos os casos apresentam padrões e conseguem se adaptar a diferentes tipos de estacionamentos. Dessa forma, foi possível gerar dados para os variados ambientes de Smart Parking, garantindo a construção de um dataset sólido, o qual contém variação suficiente para a execução de métodos de ML e DL que visam treinar sobre dados diversificados e em grandes quantidades. Sendo necessário, por fim, validar um dataset simulado comparando-o com um dataset real.

4.2.2.5 Validação

Visando a validação da eficácia e precisão do simulador de dados, a etapa de validação utilizou o mesmo dataset real apresentado na seção 4.2.1.1. Dessa maneira, buscou-se realizar uma comparação entre os padrões de estacionamentos de um dataset real já apresentado e um dataset produzido pelo simulador.

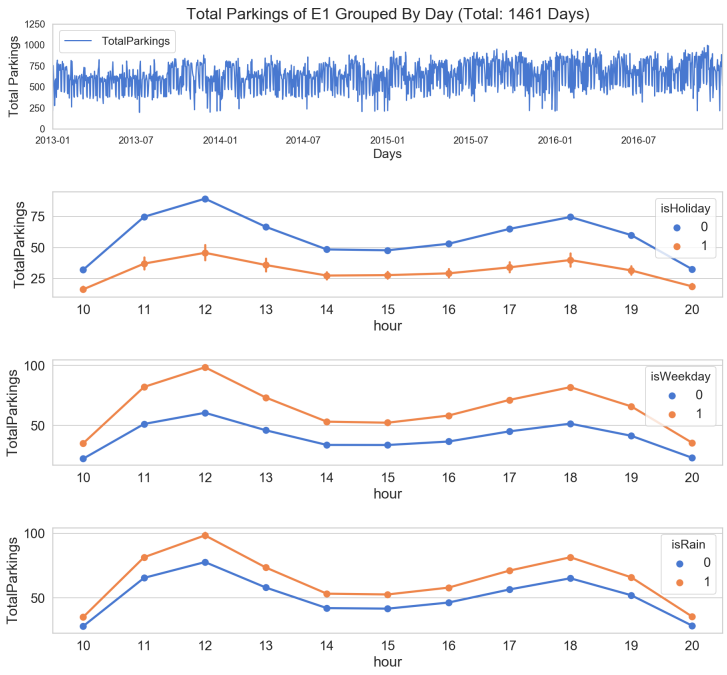


Figura 4.9: Visualização Gráfica do E1.

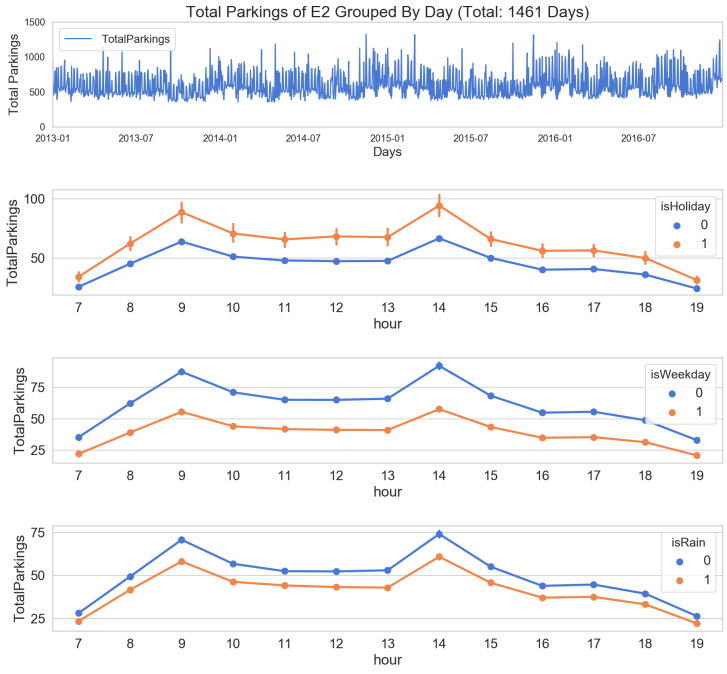


Figura 4.10: Visualização Gráfica do E2.

Para realizar a validação, no dataset com dados reais, filtrou-se apenas um local do dataset, escolhendo-se aquele que possuía o maior número de dias preenchidos e sem

dados incompletos. Além disso, filtrou-se o dataset para apenas um ano de extensão, sendo escolhido o ano de 2016, pois era o ano com maior presença de dados. Após realizar a filtragem do dataset com dados reais, obteve-se um total de 106.855 registros em um ano de estacionamentos realizados, com média de 292 estacionamentos por dia.

O dataset simulado baseou-se no estacionamento $E1$, com algumas modificações em suas variáveis, visando adequar-se ao padrão identificado no dataset real, detalhado na secão 4.2.1.3. Utilizou-se também o filtro para apenas o ano de 2016. Assim, a comparação final teve uma faixa temporal idêntica em ambos os casos. Após realizar a filtragem do dataset simulado, obteve-se um total de 249.243 registros em um ano de estacionamentos realizados, com média de 680 estacionamentos por dia.

A validação baseou-se no agrupamento dos estacionamentos em diferentes variáveis temporais, mantendo o padrão de perfil descrito na seção 4.2.1.3, sendo eles:

1. Agrupamento de estacionamentos por horas diárias, com um total de 24 agrupamentos por dia e cerca de 8784 em um ano.

2. Agrupamento de estacionamentos por dias do ano, com 366 registros para o ano de 2016 (bissexto).

3. Agrupamento de estacionamento por meses, contando com 12 registros em um ano.

4. Agrupamento da média de estacionamento por dias da semana, contando com 7 registros.

Com isso, foram gerados gráficos para cada um dos estacionamentos em cada um dos agrupamentos realizados, buscando identificar seus padrões diários, semanais e mensais. As figuras 4.11 e 4.12 apresentam, respectivamente, os resultados da análise gráfica para o dataset de dados reais e o dataset de dados simulados.

Por meio de observações foi possível identificar que o estacionamento simulado conseguiu desenvolver padrões de fluxo para dias da semana e padrões de aumento e redução durante cada dia. Esse padrão pode ser identificado no segundo gráfico das figuras 4.11 e 4.12, nas quais é possível identificar uma tendência de aumento nos 5 dias da semana

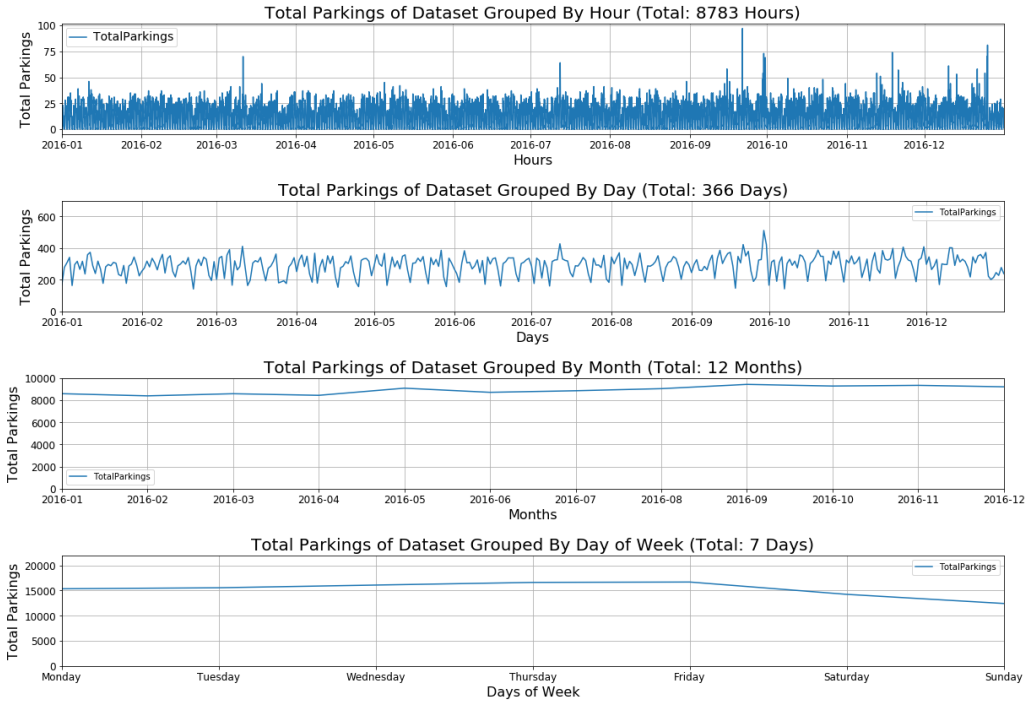


Figura 4.11: Visualização detalhada de dataset de estacionamentos reais.

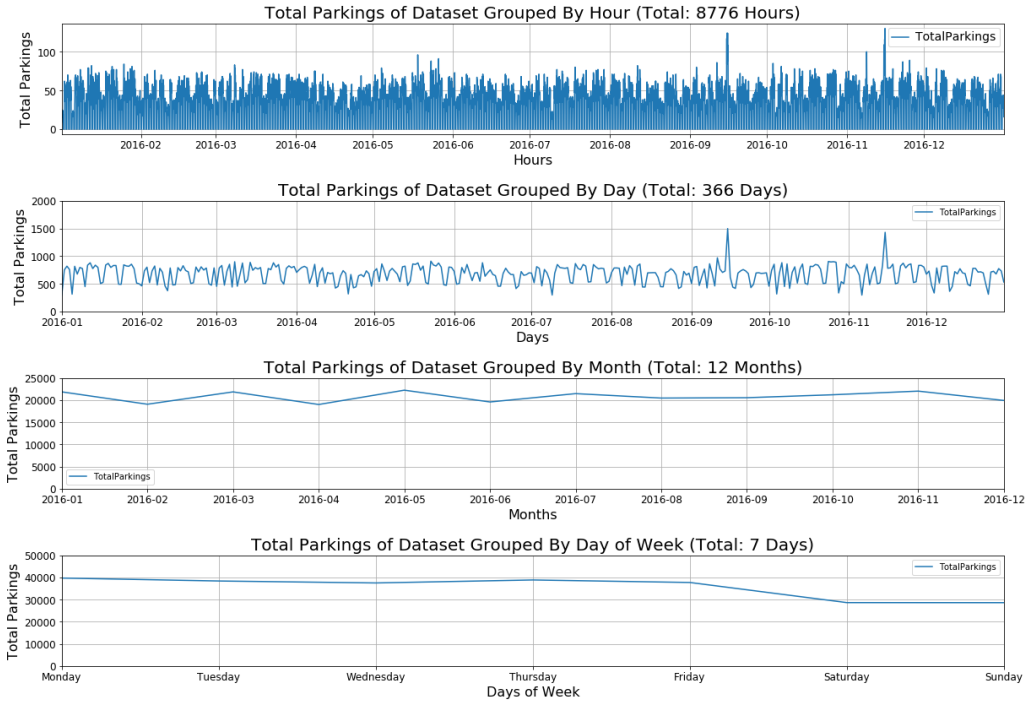


Figura 4.12: Visualização detalhada de dataset de estacionamentos simulados.

e uma queda nos finais de semana, assim como identifica o quarto gráfico de ambas as figuras, com fluxo abaixo da média nos finais de semana. Com isso, a simulação mostrou-se capaz de alcançar padrões que conseguem assemelhar-se ao de estacionamentos reais, podendo produzir dados em grande quantidade e para extensas faixas temporais.

Por fim, a simulação mostrou-se importante pois possibilitou a criação de bases de dados, que por sua vez tornam possível o início da etapa de análise de dados, que conta com a aplicação dos algoritmos de ML e DL nos diferentes ambientes de estacionamento.

4.2.3 Análise de Dados

Após as etapas de extração e simulação de dados, os conjuntos de dados obtidos foram suficientes para iniciar a etapa de aplicações de análise de dados. Essa etapa consiste na aplicação de algoritmos de ML e DL sobre os dados de estacionamento a fim de produzir uma previsão de demanda futura. A previsão de demanda permite ao gestor tomar decisões de forma antecipada, baseado em previsões estatísticas, como mudança de preço para dias com previsão de maior fluxo ou cálculo de rentabilidade futura do estacionamento. Dessa maneira, ao auxiliar na tomada de decisão dos gestores, aplicações de IA com esse foco também podem integrar o contexto de BI.

Os dados utilizados nessa etapa basearam-se exclusivamente em modelos de dados simulados. Essa escolha foi feita pois como a simulação pôde alcançar padrões próximos aos reais e a extensão dos dados simulados pode possuir milhões de registros, o uso de dados simulados foi mais interessante. Os resultados de modelos de previsão são mais consistentes e estáveis quando treinados em bases com maior quantidade de dados [9].

Dessa forma, foi realizada a simulação de um dataset de 5 anos, contendo dados de uma região de estacionamento. Essa região simula o estacionamento da cantina da universidade, que funciona em faixas horárias reduzidas, somente entre 10h e 20h, possuindo um maior fluxo em dias de semana e menor em finais de semana.

Foram escolhidas 3 diferentes abordagens para realizar as previsões, descritas na seção 2.2. A escolha por mais de uma abordagem foi feita a fim de proporcionar a realização

de um comparativo entre os resultados e a precisão de cada algoritmo aplicado.

Todas as aplicações focaram-se na previsão de demanda total do estacionamento, sendo focadas na previsão da demanda diária. Para isso, agrupou-se todos os estacionamentos realizados no mesmo dia, a fim de obter-se o total de estacionamentos realizados a cada dia presente no dataset.

A metodologia aplicada para todos os algoritmos foi a separação de 70% do dataset para o treinamento e 30% do dataset para teste e validação do modelo. Esse tipo de separação é muito comum [61] e estudos empíricos mostram que esse tipo de separação apresenta melhores resultados [36].

Essa separação dividiu o dataset do estacionamento em um total de 1278 dias registrados para treinamento e 548 dias registradas para teste. Com isso, o conjunto de dados para treinamento é extenso o suficiente e contém a variação e dinâmica pertencentes ao perfil do estacionamento, enquanto que, os 30% restantes são utilizados para testar a demanda prevista pelo modelo. Sendo esse último conjunto utilizado na comparação da demanda real registrada e a demanda prevista pelo modelo para esse mesmo conjunto.

4.2.3.1 K-Means

Como detalhado na seção 2.2.1.2, o K-Means é um algoritmo de ML baseado no uso de clusters para desenvolver suas previsões, sendo amplamente utilizado no meio científico e industrial [64].

Para esse trabalho, a função escolhida para prever o número de estacionamentos foi a média dos elementos do cluster. Dessa forma, quando um novo dado é introduzido no modelo, identifica-se o cluster com maior semelhança ao dado, e a previsão de demanda para esse dado é definida com base na média dos elementos presentes no cluster identificado.

Para o desenvolvimento do modelo, foram testados diferentes quantidades de clusters, a fim de encontrar o número que melhor se adequasse à estrutura e as variáveis do produto. Além disso, o tempo de execução do K-Means está relacionado com a quantidade de clusters, sendo mais elevado quanto maior o número de clusters.

A escolha pelo número inicial de clusters foi feita com dois critérios: ser um número

primo, o que evita empate entre clusters e ser maior do que 1, visto que a métrica utilizada para comparar os resultados de cada cluster necessita de ao menos 2 clusters para ser aplicada. Com isso, iniciou-se os testes com 3 clusters e esse total foi incrementando 2 unidades a cada execução.

A métrica utilizada para encontrar o melhor número de clusters foi o coeficiente de silhueta médio de todas as amostras. Esse coeficiente produz um valor entre 1 e -1, sendo 1 o melhor caso e -1 o pior caso. A tabela a seguir identifica o valor do coeficiente para cada quantidade de cluster testada.

Quantidade de Clusters	**Coeficiente de silhueta médio**
3	0.5709740053927154
5	0.5372799922367114
7	0.5266602287322558

Tabela 4.5: Comparativo de coeficiente de silhueta de clusters.

Assim, considerou-se o número ideal de clusters como 3, devido ao seu coeficiente ser o maior entre os clusters testados e por manter-se em um valor ideal de quantidade de clusters. A figura 4.13 apresenta a previsão de demanda para um mês produzida com o K-Means.

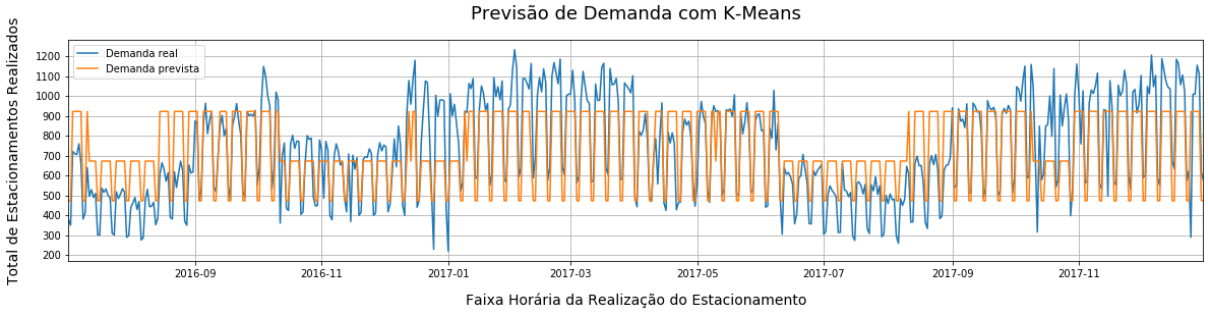


Figura 4.13: Resultado da previsão de demanda utilizando K-Means.

Como pode ser visto na figura 4.13, por basear-se na média dos elementos do respectivo cluster, o valores tendem a criar padrões lineares. Além disso, é possível visualizar

que a previsão conseguiu identificar o padrão de crescimento (dias da semana) e redução (finais de semana) semanal do gráfico, contudo, não conseguiu se ajustar as curvas individualmente.

4.2.3.2 Random Forest

O algoritmo de Random Forest (RF) visa a classificação de dados, sendo adaptado para previsão de variáveis categóricas, como descrito na seção 2.2.1.1. Contudo, sabendo que o dado a ser previsto trata-se de uma variável quantitativa (total de estacionamentos realizados), foi necessário utilizar o modelo de RF baseado em regressão.

A regressão é um método estatístico que visa deduzir relações entre variáveis dependentes e independentes, sendo utilizado para previsão de séries temporais [73], que como dito na seção 2.2.2, os dados de trânsito fazem parte. A representação básica de regressão é mostrada na equação 4.1, onde Y representa a variável dependente e a função f representa a variável independente. Com isso, a regressão ao identificar a semelhança entre variáveis, permite prever possíveis valores futuros a partir da modificação da variável independente [76].

$$Y_i = f(X_i, \beta) + \epsilon_i \tag{4.1}$$

O RF baseado em regressão obtém suas previsões a partir da média de previsão obtida por cada árvore do modelo. Assim, após o modelo ser construído, quando um dado é inserido, ele percorre cada árvore e gera uma previsão única baseada em regressão. E após passar por todo o conjunto, o valor previsto é dado como a média das previsões geradas por todas as árvores.

Dessa forma, aplicou-se a técnica de RF baseada em regressão sobre o dataset, identificando os dados previstos pelo modelo e dados reais do grupo de testes do dataset. A figura 4.14 apresenta os resultados da previsão de demanda utilizando a abordagem de RF.

A métrica utilizada para avaliar o modelo foi o coeficiente de determinação, conhecido

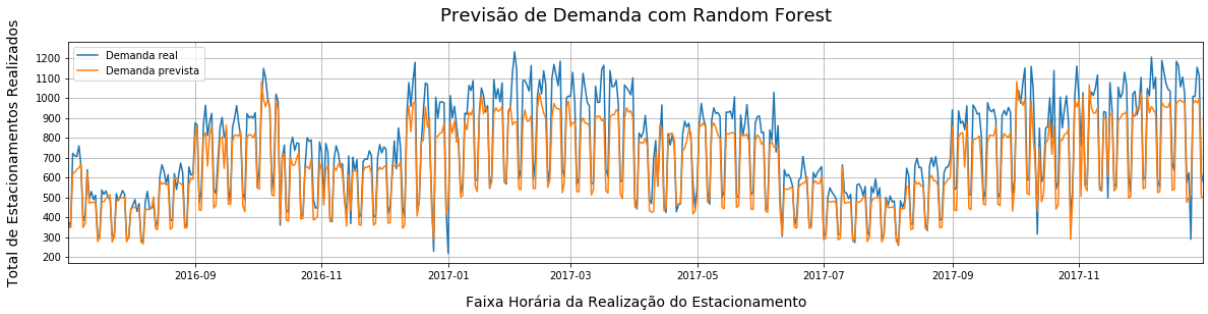


Figura 4.14: Resultado da previsão de demanda utilizando Random-Forest.

como *R squared* ou R2. O R2 é usado em regressão linear e tem por objetivo medir a proporção da variação que ocorre na variável dependente presente no modelo [94]. Geralmente o coeficiente obtém valores entre 0 e 1, sendo 1 o melhor caso. O modelo desenvolvido obteve um R2 de 0.84514, o qual demonstra que houve uma taxa considerável de ajuste no modelo obtida pelos valores previstos.

Como pode ser visto na figura 4.14, o modelo de Random Forest conseguiu identificar os padrões de fluxo do estacionamento, se ajustando nos pontos de crescimento e queda. Além disso, o modelo conseguiu realizar um ajuste de curvas superior ao do K-Means.

4.2.3.3 Long-Short-Term-Memory (LSTM)

A aplicação do LSTM no dataset seguiu o mesmo procedimento inicial realizado nas outras abordagens de IA, contudo, por se tratar de um algoritmo de DL, a sua complexidade é mais acentuada e seu modelo de treinamento é diferente. O LSTM, devido à motivos como o uso de funções de aleatoriedade presentes na etapa de treinamento, desenvolve um modelo diferente em cada execução.

A escolha pelas características de treinamento ideal do modelo foi baseada no uso da métrica Root Mean Square Error (RMSE), em português raiz do erro quadrático médio. A métrica RMSE é amplamente adotada na etapa de avaliação, para medir a diferença entre os valores previstos pelo modelo e os valores atuais [89] [22]. Além disso, o RMSE é uma métrica de propósito geral excelente para previsões numéricas [13]. Por se tratar de

uma métrica de erro, valores menores indicam um melhor ajuste do modelo. A métrica pode ser visualizada na equação 4.2.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2} \tag{4.2}$$

A quantidade de etapas de treinamento realizada no modelo foi baseada no menor valor obtido para a métrica RMSE durante o treinamento. Essa variação pode ser vista na figura 4.15. Com isso, a quantidade ideal de etapas de treinamento escolhida foi 75, pois obteve o menor coeficiente.

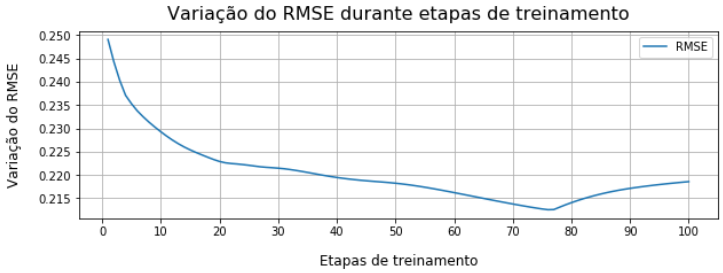


Figura 4.15: Variação do RMSE durante treinamento do modelo.

Após a definição das etapas, o modelo pôde ser treinado e testado. A figura 4.16 apresenta os resultados da previsão de demanda do modelo LSTM no mesmo conjunto de dados de teste das abordagens anteriores.

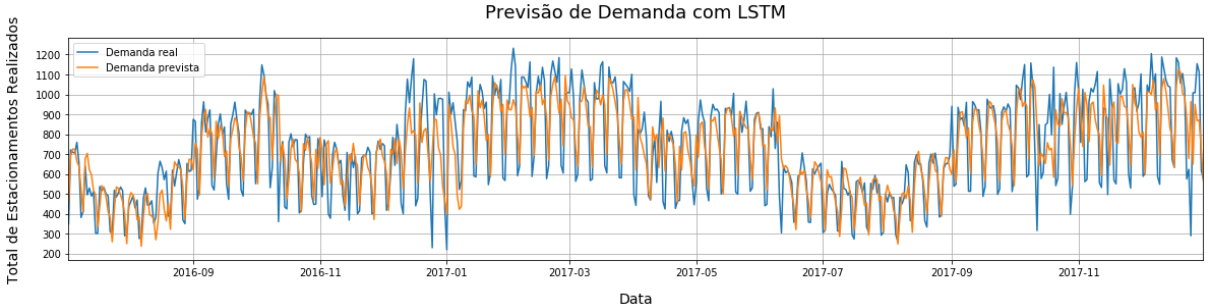


Figura 4.16: Resultado da previsão de demanda utilizando LSTM.

Por fim, para identificar um coeficiente comum de RMSE do modelo, foram desenvolvidos 10 modelos com a mesma configuração. Cada modelo pode produzir uma variação

da métrica de RMSE, devido à processos de aleatoriedade aplicados no treinamento. Com isso, o valor padrão do RMSE para o modelo foi baseado na média do valor de RMSE dos modelos gerados. Assim, o padrão médio de RMSE foi 180.36, sendo o menor registrado de 148.66 e o maior registrado de 207.79.

4.2.3.4 Comparativo de Resultados

O comparativo de resultados trata-se da análise das previsões realizadas por cada modelo a fim de compará-las em diversos aspectos para realizar a avaliação de qual modelo foi mais preciso e obteve melhores resultados. A escolha pela aplicação de todos os algoritmos no mesmo dataset foi fundamental para que os resultados finais pudessem ser comparados de forma igualitária.

A comparação dos resultados foi feita baseada na quantidade de valores previstos pelo modelo que atingiram um determinado percentual de precisão, quando comparados ao valor real registrado. Para isso, elaborou-se uma equação própria que visa identificar o nível de proximidade do valor previsto para o valor real.

O cálculo realizado para identificar esse percentual denominado nível de proximidade é detalhado na equação 4.3, no qual VR significa Valor Real e corresponde ao total real de estacionamentos realizados para um determinado dado, enquanto que VP significa Valor Previsto e corresponde ao total de estacionamentos previstos pelo algoritmo para o mesmo dado. Essa equação produz um valor igual ou inferior a 1 quando o valor de VP não for uma previsão extremamente incorreta, equivalente a um valor positivo igual ou superior ao dobro do VR. Como nos gráficos nenhum dado previsto equivale a essa diferença, a aplicação gerou valores entre 1 e 0 em todos os casos. Em termos percentuais, o nível de proximidade igual a 1 corresponde à 100%, indicando nesse caso, que o valor previsto é idêntico ao valor real, logo, valores próximos de 1 representam melhores resultados.

$$NP \ = \ 1 \ - \ \left(\frac{|\ VR \ - \ VP\ |}{VR} \right) \tag{4.3}$$

Para facilitar a visualização dos resultados, foram criados 4 níveis de faixas percentuais que correspondem ao nível percentual de proximidade encontrado para cada item do dataset. Assim, utilizou-se um algoritmo para identificar a faixa percentual de cada registro por meio da aplicação da equação em cada estacionamento do dataset. Após obter o nível de proximidade de cada registro, o dataset foi agrupado por níveis de proximidade percentuais, obtendo quantos estacionamentos percentem a cada nível. Os 4 níveis compreendem: 90% à 100% (1), 80% à 90% (2), 70% à 80% (3) e menor que 70% (4). Para exemplificar a metodologia, caso o valor real seja 10 e os valores previstos sejam 5, 8.2, 9.8, 10.9, esses valores irão integrar, respectivamente, os níveis, 4, 2, 1, 1.

A quantidade de dados utilizadas na previsão corresponde ao dataset de teste, que não integrou o treinamento da solução e permite a comparação entre os valores reais e os valores previstos gerados pelos modelos. A tabela 4.6 apresenta o comparativo entre a precisão obtida em cada modelo, identificando o algoritmo e a quantidade de registros cuja previsão produzida pelo modelo equivale ao nível de proximidade. A tabela conta ao todo com 547 registros, dessa forma, a soma das linhas das colunas com *"Total"* será sempre 547 registros e a soma das linhas das colunas de *"Total (%)"* será de 100%.

Nível de Proximidade	**KM Total**	**RF Total**	**LSTM Total**	**KM Total(%)**	**RF Total(%)**	**LSTM Total(%)**
90% < x <= 100%	180	279	242	32.91%	51.0%	44.24%
80% < x <= 90%	169	215	143	30.89%	39.31%	26.14%
70% < x <= 80%	79	33	67	14.44%	6.03%	12.25%
x < 70%	119	20	95	21.76%	3.66%	17.37%
Total	547	547	547	100%	100%	100%

Tabela 4.6: Comparativo de nível de proximidade alcançado pelos modelos.

Os resultados obtidos na tabela 4.6 demonstram que o algoritmo RF obteve a maior quantidade de registros de previsão com alto nível de proximidade, seguido do algoritmo LSTM e por último o K-Means. Um importante ponto a ser considerado é de que a aplicação do K-Means centralizou todos os resultados de forma média e linear. Isso faz com que na maioria dos casos o resultado fique próximo do valor total, pois faz parte

da média do conjunto. Dessa maneira, mesmo que os percentuais do K-Means sejam relevantes, os resultados obtidos pelo RF e LSTM conseguiram ajustar-se às curvas do gráfico, produzindo resultados mais específicos e não lineares, sendo por isso, considerados mais relevante e próximos da realidade.

4.3 Plataforma de BI

A plataforma de BI trata-se da solução final proposta para a dissertação. Essa plataforma consiste na união de todos os conhecimentos produzidos pelas etapas anteriores em uma única estrutura. Como detalhado nos diagramas da seção 3.3.1, a solução abrange desde a leitura do dados incluindo a limpeza e filtro sobre os mesmos, a disponibilização de visualizações de análises que permitam auxiliar na tomada de decisão. Além disso, a solução disponibiliza interfaces para aplicações de ML e DL para a criação e execução de modelos focados na previsão de demanda do estacionamento.

A solução foi concebida como uma aplicação web, pois seguindo tendências modernas e tendo em vista que a aplicação deverá comunicar-se com a nuvem, esse tipo de aplicação é muito utilizada para sistemas que interagem com IoT e nuvem [75]. As vantagens dessa plataforma estão na disponibilidade e compatibilidade que possuem, visto que podem ser acessados por qualquer dispositivo computacional com acesso à internet, incluindo computadores e smartphones. Tendo em vista que todas as soluções desenvolvidas no decorrer do trabalho foram implementadas utilizando Python, visando manter a compatibilidade, a plataforma web também baseou-se em Python. Para isso, utilizou a biblioteca Dash, detalhada na seção 4.1.5, a qual permite a criação de aplicações web com Python. Dessa forma, o reaproveitamento de algoritmos desenvolvidos nas etapas anteriores foi de 100% na plataforma BI.

Além disso, o Dash conta com a biblioteca Plotly, descrita na seção 4.1.6, que permite a criação de gráficos dinâmicos, dessa maneira, todos os gráficos gerados pela plataforma são dinâmicos e todos contam com funcionalidades em comum, entre elas:

- Zoom: o usuário pode aplicar zoom em qualquer região do gráfico, para obter uma

visualização mais precisa para qualquer tipo de gráfico. Trata-se de uma funcionalidade muito importante, como demonstrado nas figuras 4.2 e 4.3, o zoom em um gráfico muito denso permite encontrar padrões que não eram destacados com a visualização comum.

- Exportar para Imagem: é possível gerar uma exportação da exata visualização do gráfico em formato de imagem. Essa função auxilia o gestor na criação de relatórios e na exportação de análises para documentos. A função exporta todo o conteúdo do gráfico, incluindo seu título, legenda e descrição. Por esse motivo, todos os gráficos gerados pela plataforma possuem todas as informações personalizadas, a fim de permitir uma exportação que traga os dados da análise da forma mais detalhada e precisa possível.

- Legendas Dinâmicas: essa funcionalidade permite que o gestor possa controlar quais elementos serão destacados ou omitidos no gráfico. Esse procedimento é realizado por meio do clique sobre o nome da legenda, fazendo com que a linha/barra/pontos que correspondem aquela legenda sejam omitidas do gráfico ou sejam novamente destacadas. Essa função permite ao gestor um maior controle da informação contida no gráfico, auxiliando em análises ao garantir, por exemplo, que linhas de um gráfico que estão muito próximas e até se sobrepondo, possam ser visualizadas individualmente.

Outro importante fator presente na maioria dos gráficos é a descrição de itens. Essa funcionalidade trata-se de uma das mais relevantes para a análise de dados e compreensão dos gráficos, sendo uma informação totalmente personalizada para cada gráfico da aplicação. A função é ativada ao sobrepor o mouse sobre um ponto, barra ou linha do gráfico, criando um bloco que apresenta os dados específicos da informação presente naquele exato ponto do gráfico. Esse bloco pode ser visto na cor laranja na figura 4.20. Essa funcionalidade permite ao gestor identificar informações sobre o dado que podem não estar presentes de forma visual no mesmo. Contendo dados como: se o estacionamento

que corresponde ao ponto selecionado foi feito em um dia de chuva, feriado ou final de semana, seu horário exato de entrada e saída, entre outros.

A plataforma foi divida em diversas páginas, cada uma contendo um foco específico e abrangendo algoritmos de visualização, análise e/ou IA, como especificado nos casos de uso da seção 3.3.1. A divisão de páginas pode ser vista na estruturação dos arquivos da plataforma, enquanto que, as funcionalidades de cada página integram algum dos três grandes grupos descritos nos diagramas 3.5, 3.6 e 3.7.

4.3.1 Estrutura de Arquivos

A estrutura de arquivos visou a separação dos arquivos em pastas, de acordo com seus objetivos, a fim de obter uma estrutura mais organizada. A figura 4.17 apresenta a estrutura de dados da aplicação, sendo ela detalhada a seguir:

- Assets: contém os arquivos que tratam da estilização da interface, incluindo arquivos do tipo Cascading Style Sheets (CSS) e imagens utilizadas na interface, como o logotipo da aplicação

- Configs: contém arquivos de configuração da aplicação, incluindo dados sigilosos utilizados para realizar a autenticação com a nuvem.

- InputData: contém os arquivos csv obrigatórios, inseridos pelo usuário antes de iniciar a aplicação, sendo utilizados como a base para todas as análises desenvolvidas pela plataforma. A pasta conta com o arquivo *"parkings.csv"* que corresponde ao dataset de estacionamentos reais, contendo em geral, uma extensa base de dados que conta com todos os registros de estacionamentos. O segundo arquivo csv obrigatório é chamado *"parkings-forecasts.csv"* e esse arquivo é preenchido com estacionamentos futuros, que não existem no dataset "parkings.csv". Esse último contém o registro das datas nas quais pretende-se gerar previsões de demanda nas páginas de aplicações de ML e DL.

- Pages: contém as páginas da aplicação, que são acessadas a partir do menu. As páginas presentes nessa pasta podem ser identificadas pelo seu grupo por meio do prefixo utilizado em seu nome como *"basic_"* e *"predict_"*.

- Raiz: os arquivos presentes na pasta raiz são utilizados para a construção da interface da aplicação, o menu principal, o carregamento do dataset e a inicialização do sistema.

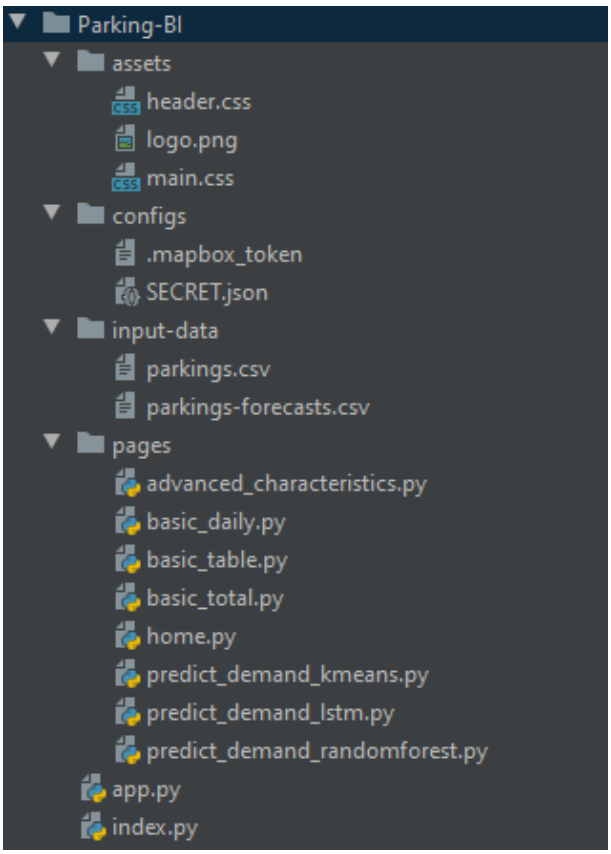


Figura 4.17: Estrutura de arquivos.

Ao todo a estrutura conta com 7 páginas, que podem ser acessadas por meio do menu superior da aplicação. O menu superior é a principal ferramenta utilizada para navegabilidade, sendo a única estrutura que permite ao usuário navegar para qualquer página da plataforma. Além disso, o menu está presente em todas as interfaces do sistema e está dividido em duas partes. Na esquerda do menu encontram-se o logotipo e nome

da aplicação e à sua direita estão os botões de ação, em sua maioria do tipo suspenso (*drop-down*) os quais garantem a navegabilidade para outras interfaces. A figura 4.18 apresenta o menu da aplicação.

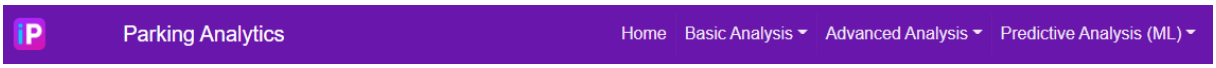


Figura 4.18: Menu da aplicação.

4.3.2 Página Inicial

A página inicial conta com duas estruturas principais, que são separadas em um bloco superior denominado *Current Status* e um bloco inferior denominado *World View* que podem ser vistos na figura 4.19.

O bloco superior conta com uma visualização de gráfico de barras do fluxo de um dia do estacionamento, agrupando o total de registros por hora de entrada. Além disso, conta com dois filtros que permitem a seleção da região (estacionamento) do dataset e a data que pretende-se realizar a visualização gráfica. Ao lado do gráfico encontra-se uma tabela dinâmica, que é atualizada juntamente com o gráfico e contém uma análise básica do gráfico, identificando o total de horas registradas para a data selecionada e a média de estacionamentos realizados por hora na mesma data.

O bloco inferior possuí um gráfico em escala mundial, que obtém informações atuais dos estacionamentos presentes na nuvem e identifica informações importantes sobre cada um, como localização, nome e seu estado de funcionamento atual. O bloco também conta com filtros que permitem modificar o agrupamento e a legenda do gráfico, exibindo por exemplo, estacionamentos atualmente abertos em cor diferente daqueles que encontram-se fechados no momento.

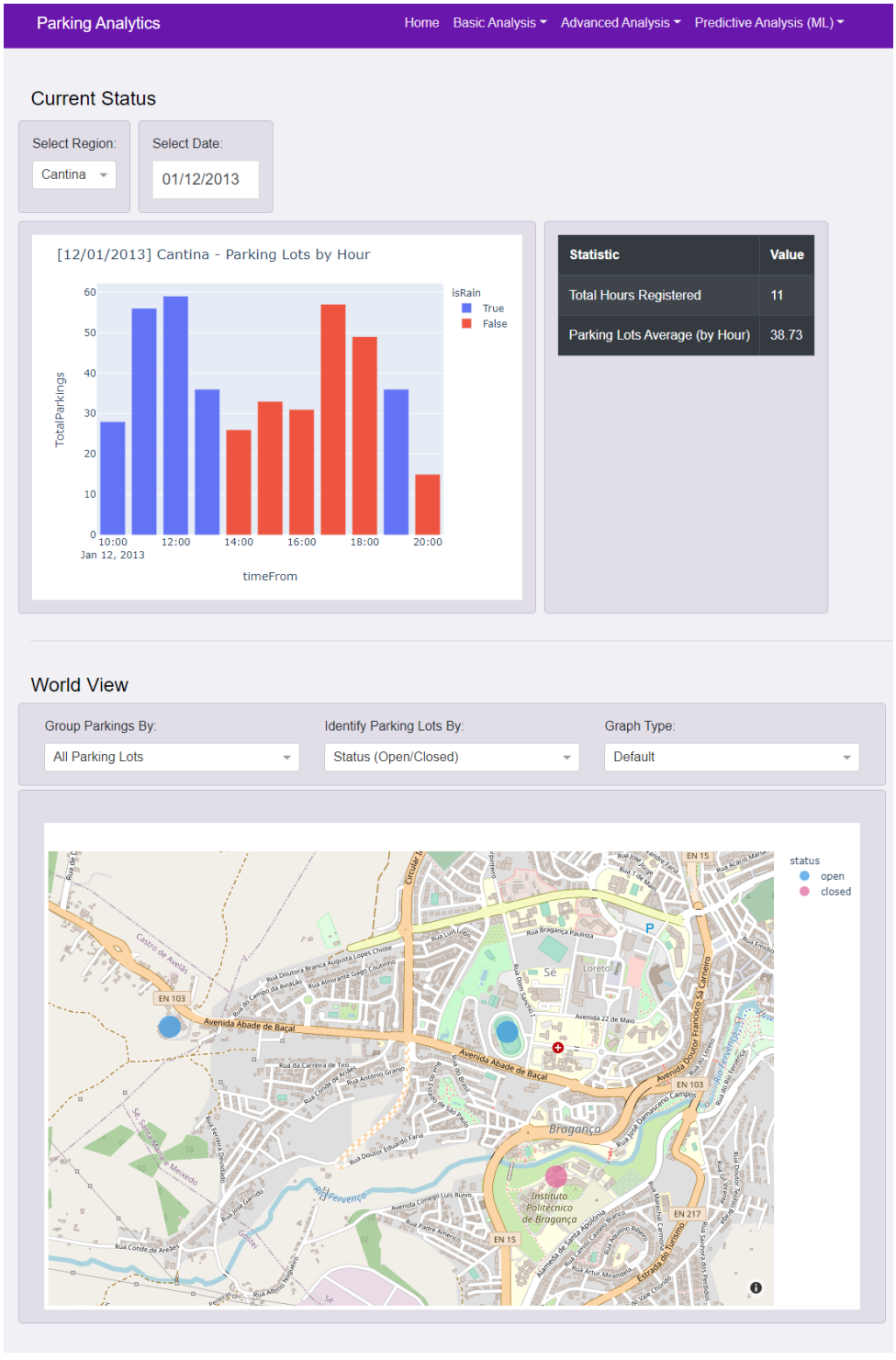


Figura 4.19: Visualização da página inicial.

4.3.3 Páginas de Análises Básicas

O menu de análises básicas é composto por três páginas que contemplam tarefas mais simples, focadas na visualização de dados e estatísticas básicas, as páginas representam a solução destacada nos diagramas de caso de uso da figura 3.5. O acesso a todas as páginas desse contexto é feito pelo item *"Basic Analysis"* do menu da figura 4.18.

4.3.3.1 Estacionamentos Diários

A página de estacionamentos diários conta com um único bloco que contém filtros em sua parte superior, um gráfico centralizado e uma tabela dinâmica à direita. A página pode ser visualizada na figura 4.20.

O gráfico abrange todos os estacionamentos individuais realizados no estacionamento escolhido e na faixa de tempo inicial e final definidas pelo gestor nos filtros superiores. O gráfico contém no eixo X, o horário de entrada no estacionamento e no eixo Y, o horário de saída. Além disso, os filtros permitem que o gráfico destaque cada estacionamento realizado por sua característica, sendo elas: chuva, feriado ou final de semana. O bloco de informações contém todas as variáveis de negócio de cada estacionamento realizado.

A tabela que encontra-se ao lado direito do gráfico agrupa informações à cerca dos estacionamentos filtrados. Sua função é identificar o total de estacionamentos, a média diária de estacionamentos realizados, o total de registros com o clima definido como chuva e o total de registros com o clima definido como sol.

4.3.3.2 Total de Estacionamentos

A página de total de estacionamentos contém um bloco único, dividido em filtros na parte superior, um gráfico centralizado e à direita contém uma tabela dinâmica e um botão de download. A figura 4.21 demonstra a interface completa referente à página.

Essa página tem por objetivo a visualização de agrupamentos totais de estacionamento, permitindo por meio de filtros, o agrupamento de estacionamentos realizados por hora, por dia ou por mês. Além disso, os filtros permitem a definição de uma data inicial e uma data

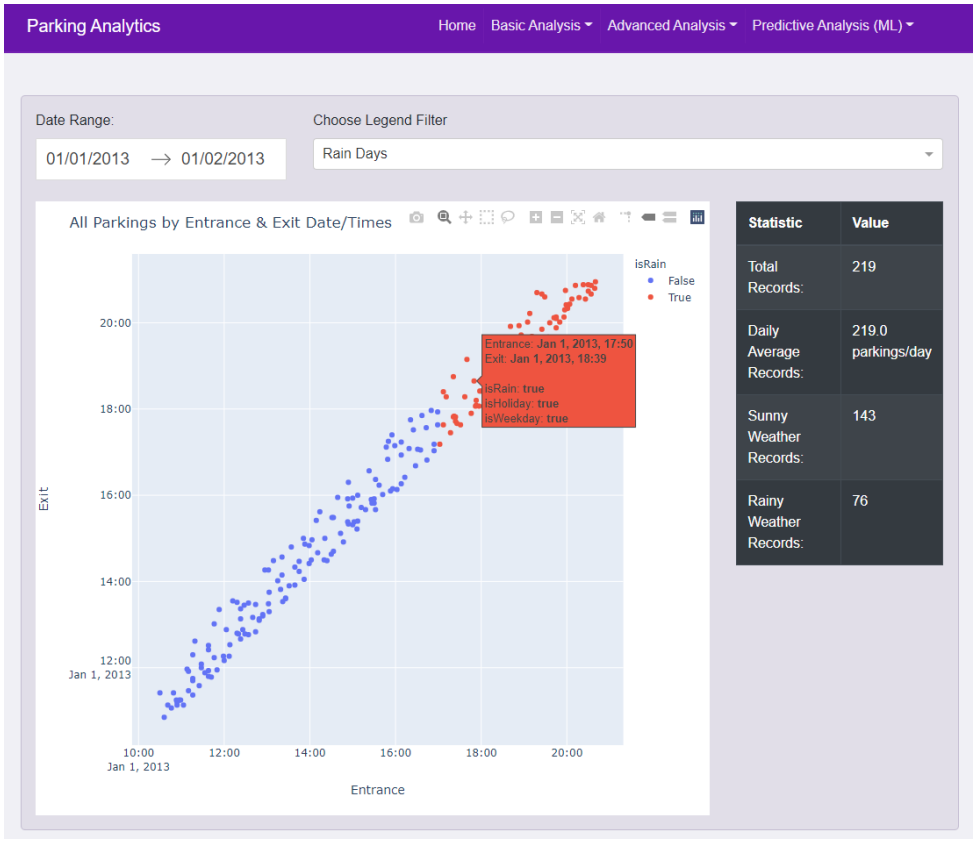


Figura 4.20: Página de Estacionamentos Diários.

final para realizar a análise e visualização. A tabela dinâmica demonstra quantos registros restaram no dataset após a aplicação dos filtros e a média de estacionamentos realizados por unidade nesse conjunto filtrado. Os dados da tabela podem ser exemplificados como: caso o filtro de estacionamentos seja de 7 dias, se o agrupamento for feito por hora, haverão um total de 7 (dias) vezes 11 (total de horas diárias de funcionamento do estacionamento) que equivale a 77 registros. Além disso, caso o mesmo número de dias seja agrupado por dias da semana, haverão apenas 7 registros no gráfico (1 para cada dia da semana). A tabela por ser dinâmica, gera uma nova análise sempre que algum filtro é alterado.

Uma importante funcionalidade presente na página está no botão "Download CSV". Esse botão permite que, após o usuário realizar qualquer filtragem e agrupamento no dataset principal, ele possa exportar um subgrupo do dataset que corresponde somente aos dados presentes no gráfico, advindos da aplicação dos filtros realizados no dataset

principal.

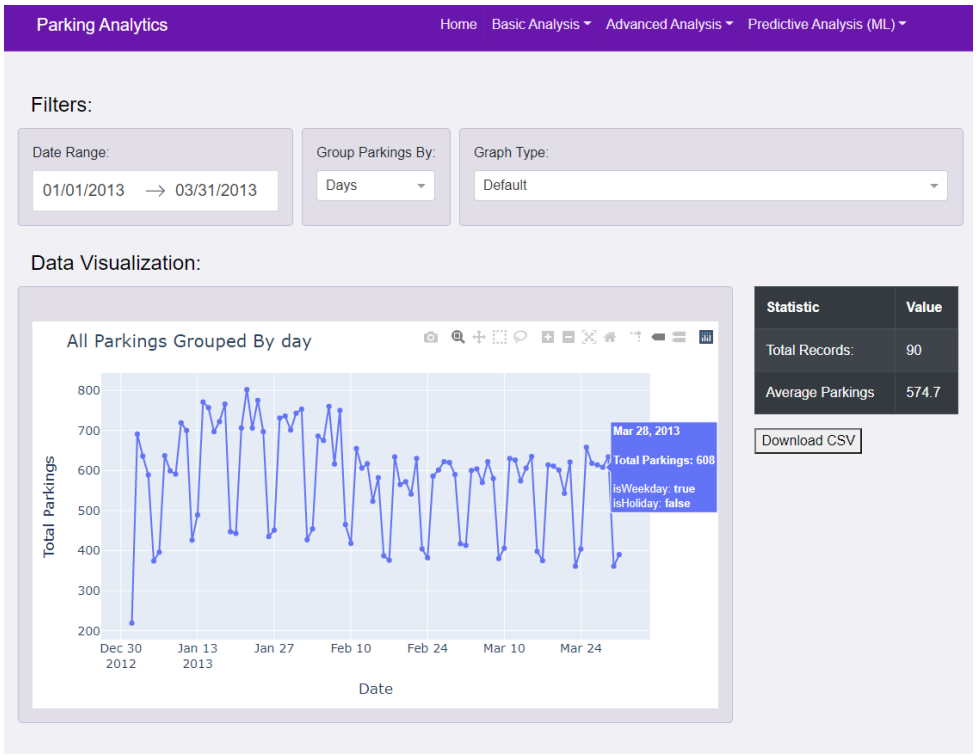


Figura 4.21: Página de Total de Estacionamentos.

4.3.3.3 Visualização em Tabela

Essa página contém a interface mais distinta dentro da aplicação, possuindo um único bloco com uma tabela que o ocupa completamente, sendo a única página da aplicação que não possui nenhum gráfico. A página pode ser vista na figura 4.22.

A sua principal funcionalidade é proporcionar ao gestor uma visualização de todo seu dataset em uma grande tabela, que agrupa todas as linhas e colunas presentes nele. Essa interface assemelha-se a softwares de visualização de planilha. A tabela contida na página é dinâmica, permitindo que o gestor possa filtrar os dados de seu dataset por qualquer valor em qualquer coluna, permitindo também múltiplos filtros.

É possível a partir da página identificar por exemplo, o número de registros de estacionamentos realizados em um determinado dia, em uma determinada região, que registrou

chuva e ocorreu em um final de semana. Esses filtros são aplicados de forma textual na parte superior da tabela e permitem que o gestor possa identificar todas as variáveis de todos os dados que foram filtrados.

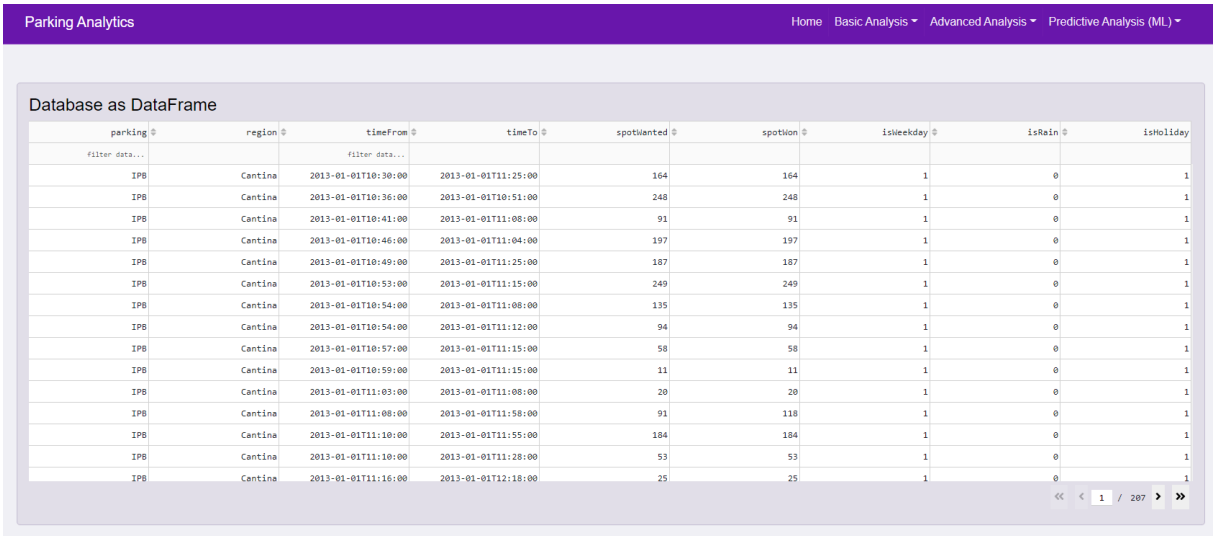


Figura 4.22: Página de Visualização em Tabela.

4.3.4 Páginas de Análises Avançadas

O menu de análises avançadas é acessado pelo campo *"Advanced Analysis"* da figura 4.18. Esse menu tem por objetivo agrupar páginas que permitam uma análise mais profunda dos estacionamentos, que tragam informações mais detalhadas e em maior quantidade sobre aspectos do produto.

4.3.4.1 Características de Estacionamento

A página de características de estacionamento é a única interface presente no menu de análises avançadas, sendo dividida em 2 blocos. O bloco superior contém filtros e o bloco inferior contém análises. A figura 4.23 demonstra uma visualização completa dessa página.

Os filtros assemelham-se aos das páginas de análises básicas, permitindo que o usuário possa selecionar uma faixa de datas do dataset e o estacionamento que pretende analisar.

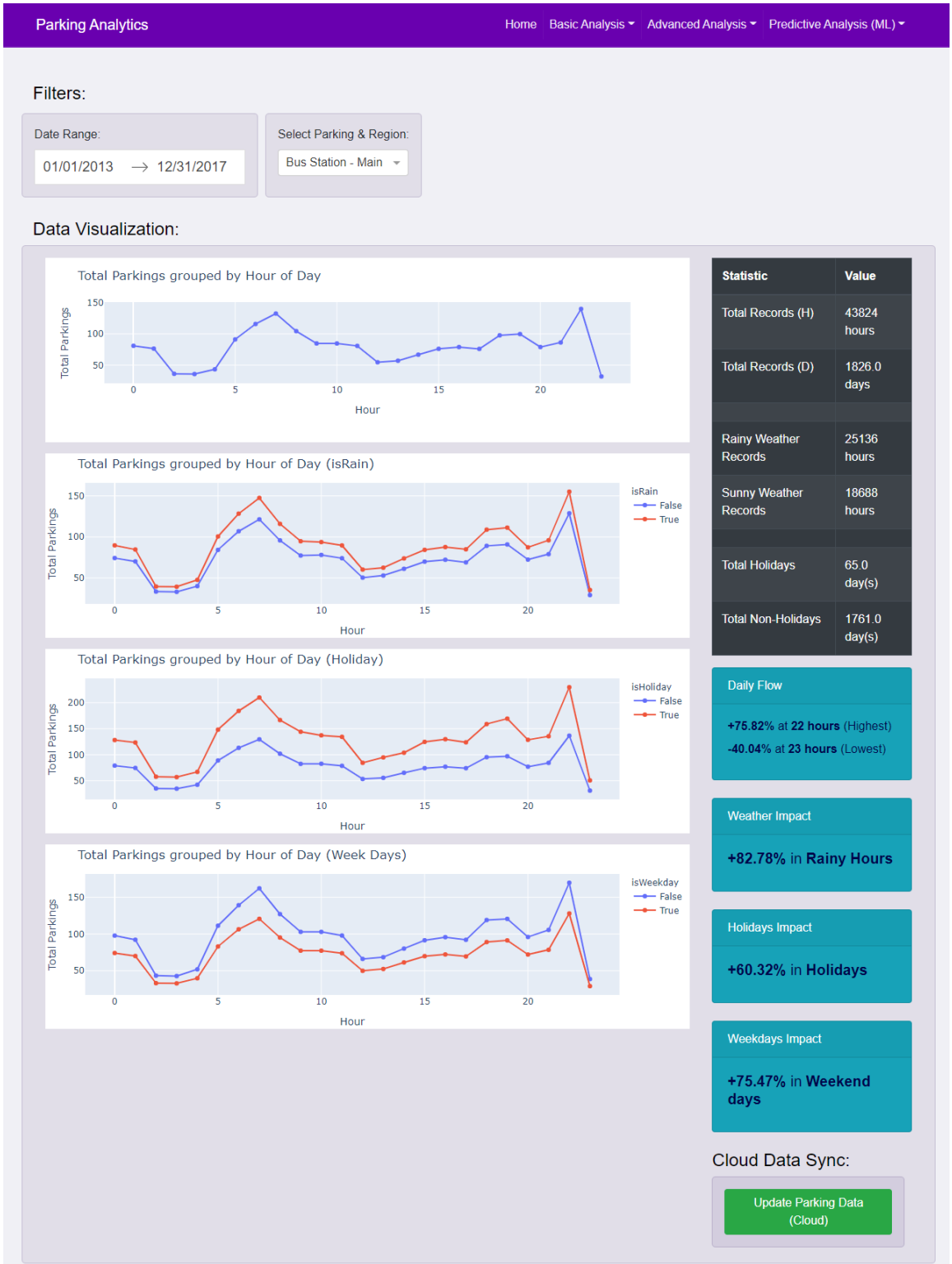


Figura 4.23: Página de Características de Estacionamento.

O bloco inferior de visualizações pode ser dividido em duas partes, a da esquerda que corresponde aos gráficos gerados e a da direita que corresponde às estatísticas encontradas para o estacionamento selecionado na faixa temporal definida para análise. Os 4 gráficos exibem a média de estacionamentos realizados agrupados por horário de funcionamento, sendo muito similar aos gráficos da figura 4.9 e 4.10. O primeiro gráfico conta com uma média geral, enquanto que os gráficos restantes dividem o total de estacionamentos por cada regra de negócio do produto, sendo elas, respectivamente: clima, feriados e finais de semana.

Ao lado direito do gráfico encontram-se uma tabela dinâmica e 4 cartões em azul. A tabela agrupa diversas informações relativas aos dados analisados, identificando por exemplo, o total de feriados e não feriados registrados, o total de estacionamentos registrados com chuva e com sol e o total de registros agrupados por horas e dias. Os 4 cartões apresentados na página tem por objetivo destacar o perfil identificado para o estacionamento, baseando-se na análise feita no conjunto de dados filtrado pelo usuário. Os cartões representam respectivamente:

- Daily Flow: identifica os horários com maior e menor fluxo médio de estacionamentos registrados.
- Weather Impact: identifica em qual clima (sol/chuva) o estacionamento tem maior fluxo e quanto esse percentual é superior.
- Holidays Impact: identifica o impacto de feriados e não feriados no estacionamento, apontando qual tipo de dia tem maior fluxo no estacionamento e quão maior é em média esse fluxo.
- Weekdays Impact: identifica o impacto de dias de semana e finais de semana no estacionamento, apontando qual desses dias tem maior fluxo em média e quanto equivale esse percentual.

Por fim, a página conta com o botão de sincronização na nuvem *"Update Parking Data (Cloud)"*, localizado na parte inferior direita da interface. Esta funcionalidade tem

por objetivo capturar os dados presentes nos cartões, que contém a análise de perfil do estacionamento e realizar a sincronização na nuvem. Esse processo identifica o registro do estacionamento na nuvem e atualiza suas variáveis de perfil presentes no campo *"Statistics"* do estacionamento, já descrito na tabela 3.1. Assim, o perfil do estacionamento é atualizado para todos os outros módulos do Smart Parking que possam se beneficiar dessa informação.

4.3.5 Páginas de Análises Preditivas

Trata-se do conjunto de análises preditivas corresponde às páginas da ferramenta que tem por foco a aplicação de algoritmos de ML e DL sobre o dataset de estacionamentos. Cada página tem como objetivo a aplicação de um algoritmo específico, criando um modelo de previsão e apresentando os resultados de previsões de demanda futura.

Todas as páginas do conjunto possuem uma estrutura similar, contendo os seguintes blocos:

- Filtros: localizados na parte parte superior da página, esse bloco contém filtros comuns e filtros específicos de cada algoritmo. Os filtros comuns são: filtro de seleção de estacionamento, que permite selecionar o estacionamento no qual deseja-se realizar as previsões de demanda e criação do modelo. O filtro de treino/teste, que permite ao gestor selecionar o percentual de dados que pretende utilizar para treino e para teste, o que implica diretamente no tempo de execução do modelo e resultados produzidos pelo mesmo. Os filtros específicos contém dados de configuração de cada algoritmo, assim com descritos na seção 4.2.3. Esses filtros permitem ao gestor o controle das variáveis utilizadas na criação do modelo, podendo por exemplo, definir o número de clusters que deseja ao utilizar na previsão com K-Means.

- Gráfico: localizado na parte central da página, o bloco contém um gráfico com dados agrupados por dias. Além disso, o gráfico possui três legendas, que podem ser visualizadas nas figuras 4.24, 4.25 e 4.26. Essas legendas são compostas por: *"Real Data"*, correspondente aos dados reais do dataset, *"Predict Data"* correspondentes

aos valores previstos pelo modelo após a aplicação no próprio dataset e *"Future Demand"* que contém a previsão de demanda futura gerada pelo modelo, a qual não existem dados reais para serem comparados.

- Tabela: localizada na parte inferior da página, a tabela dinâmica apresenta de forma textual e percentual os resultados obtidos pelo modelo. Os resultados da tabela são obtidos a partir da aplicação da equação de nível de proximidade, detalhada na equação 4.3, gerando uma tabela com estrutura semelhante à tabela 4.6, porém abrangendo 7 níveis percentuais.

As páginas podem ser acessadas pelo botão *"Predictive Analysis"* do menu da figura 4.18. Ao todo existem três páginas, todas possuem o mesmo objetivo de previsão de demanda, contudo cada uma contém um algoritmo de previsão diferente, sendo os mesmos algoritmos aplicados nas seções 4.2.3.1, 4.2.3.2 e 4.2.3.3.

4.3.5.1 Previsão com K-Means

Essa interface tem por objetivo a criação e aplicação de um modelo K-Means no dataset de estacionamentos. Essa previsão, como detalhada na seção 4.2.3.1, permite a criação de modelos de previsão de dados com o uso de clusters. A página pode ser visualizada na figura 4.24.

A página conta com dois filtros específicos do algoritmo. O primeiro trata-se de um conjunto de caixas de marcação que permitem ao gestor selecionar quais variáveis do dataset que o modelo irá considerar durante a criação dos clusters. O segundo filtro corresponde ao número de clusters que serão criados no modelo. Ambos os filtros permitem ao gestor gerar previsões mais simples ou mais complexas, que por sua vez, consomem um tempo menor ou maior para serem produzidas.

4.3.5.2 Previsão com Random Forest

A interface de previsão com Random Forest apresenta uma página para a criação e aplicação de um algoritmo de RF, baseado em ML e detalhado na seção 4.2.3.2, cujo modelo de

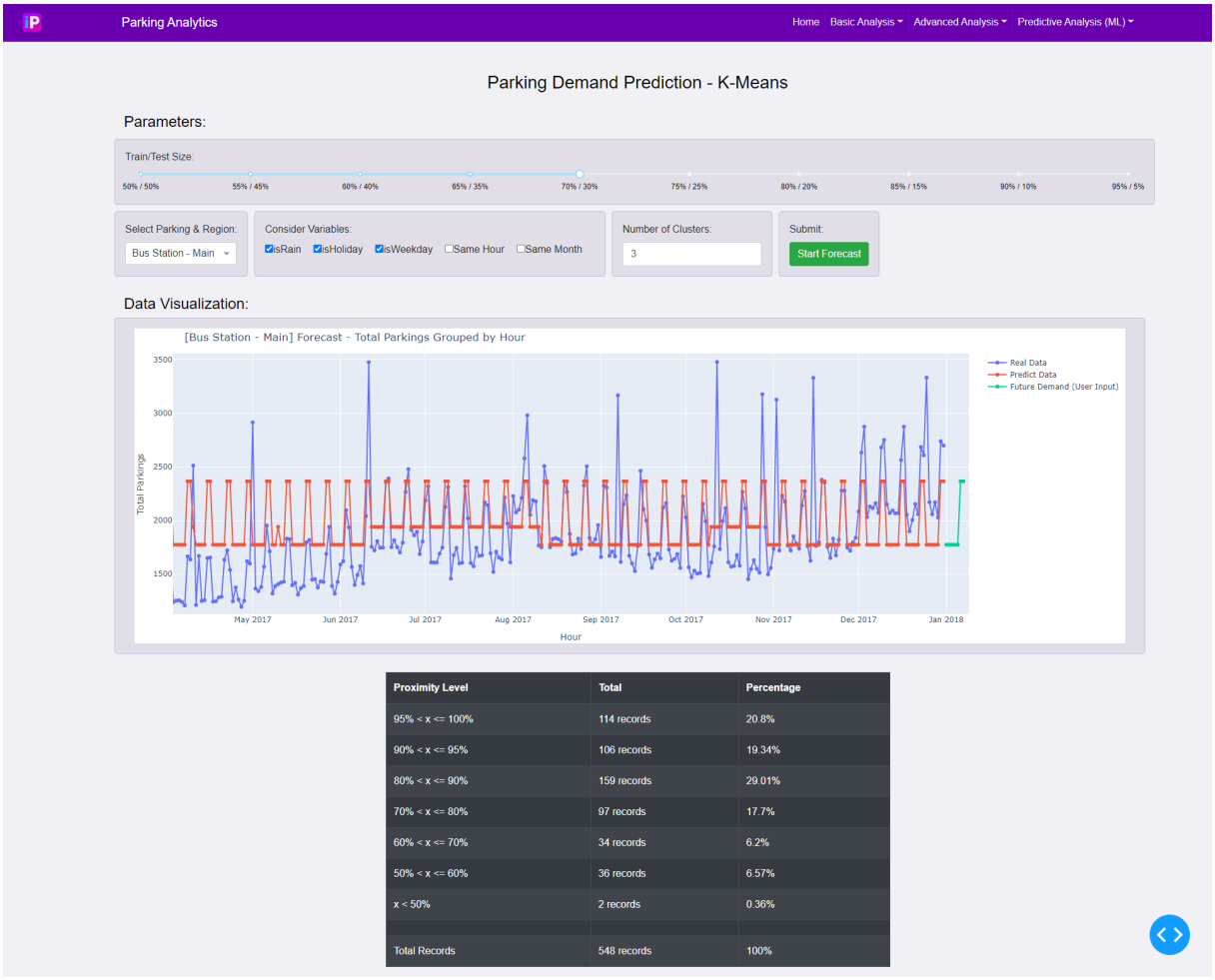


Figura 4.24: Página de Previsão de Demanda com K-Means.

previsão baseia-se na criação de árvores de decisão. A figura 4.25 apresenta a visualização da página.

Como o algoritmo de RF não possuí características que permitem a escolha de variáveis com muita flexibilidade, a página não conta com filtros próprios, sendo a única cujo modelo gerado não permite ao gestor definir variáveis específicas do algoritmo. Dessa maneira, o modelo é gerado pelos filtros comuns e os resultados produzidos são visualizados nos gráficos e na tabela presente na página.

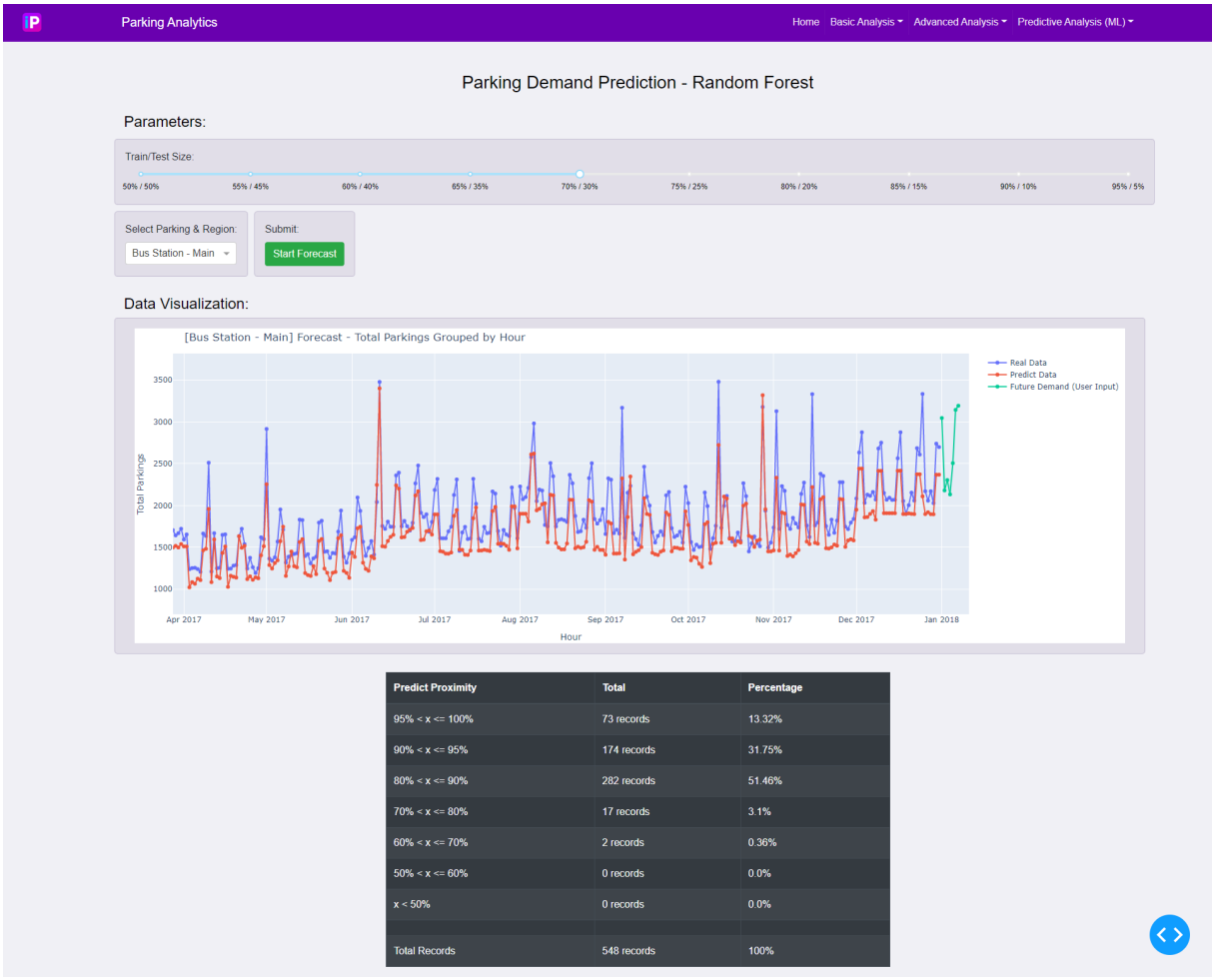


Figura 4.25: Página de Previsão de Demanda com RF.

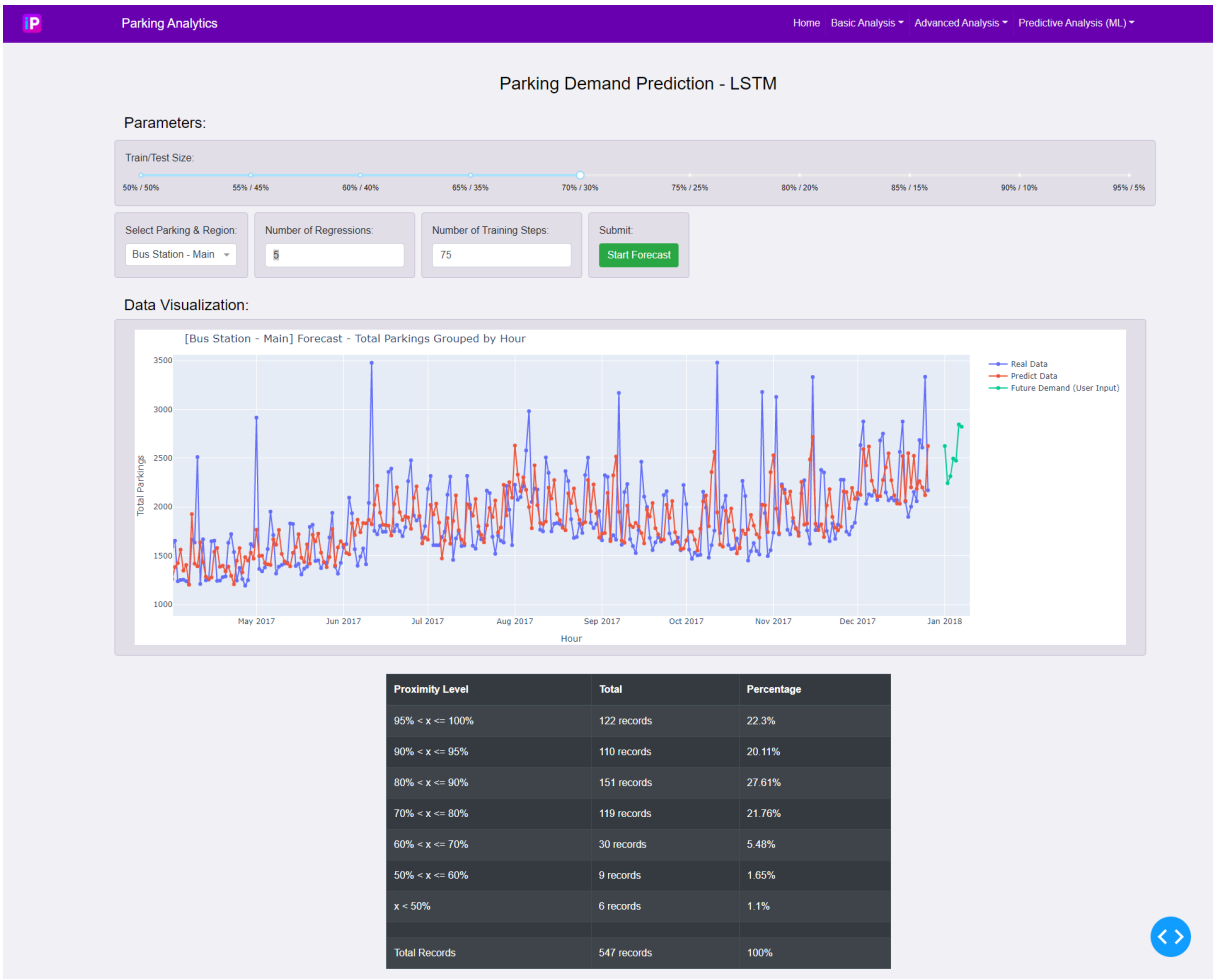
4.3.5.3 Previsão com LSTM

A página de Previsão com LSTM conta com uma interface que permite o desenvolvimento e aplicação do algoritmo LSTM, baseado em DL e descrito na seção 4.2.3.3. Essa interface diferencia-se das anteriores por possuir um modelo cujo treinamento é realizado utilizando NN. Devido a esse fator, a etapa de desenvolvimento do modelo torna-se mais complexa e muitas vezes mais lenta, quando comparado às interfaces que se utilizam em modelos de ML. A página pode ser visualizada na figura 4.26.

Os filtros específicos do modelo permitem ao gestor a definição da quantidade de regressões, que definem quantos valores anteriores serão considerados em cada etapa do

treinamento, e a quantidade de etapas de treinamento que o modelo irá utilizar.

Vale ressaltar que quanto maior a quantidade de regressões e etapas de treinamento, maior será o tempo de execução para a criação do modelo. Além disso, quantidades maiores não especificamente garantem uma maior precisão. Isso ocorre pelo fato de que cada treinamento contém etapas de cálculos que geram aleatoriedade, podendo produzir resultados diferentes de previsão em cada nova execução.



Proximity Level	Total	Percentage
95% < x <= 100%	122 records	22.3%
90% < x <= 95%	110 records	20.11%
80% < x <= 90%	151 records	27.61%
70% < x <= 80%	119 records	21.76%
60% < x <= 70%	30 records	5.48%
50% < x <= 60%	9 records	1.65%
x < 50%	6 records	1.1%
Total Records	547 records	100%

Figura 4.26: Página de Previsão de Demanda com LSTM.

Capítulo 5

Conclusões e Trabalhos Futuros

Esse capítulo tem por objetivo realizar uma visão geral de todo o desenvolvimento do trabalho, com o intuito de verificar se os requisitos levantados no início da pesquisa foram atendidos pela solução proposta.

O conceito e aplicação de estacionamentos inteligentes estão em constante crescimento, principalmente no contexto de cidades inteligentes, sendo uma realidade que brevemente fará parte do contexto social. Identificar os padrões de dados desse tipo de setor e conseguir realizar análises e previsões sobre os mesmos, traz aos gestores um maior controle e precisão na tomada de decisões. Além disso, um modelo de dados estruturado se faz necessário para o bom funcionamento de um sistema de Smart Parking, que é constituído pela união e trabalho conjunto de diversos módulos.

O objetivo proposto da união estrutural pôde ser alcançado com o uso de padrões bem estabelecidos e amplamente utilizados, como o Fiware, que serviu de base para o desenvolvimento da estrutura que conseguiu abranger as variáveis de um Smart Parking e características desejáveis do produto, como a consideração de fatores climáticos e feriados. Essa estrutura também permitiu a execução de modelos de análise de dados que puderam se beneficiar dessas características, a fim de destacá-las e auxiliar o gestor na identificação do perfil de cada estacionamento e para tomadas de decisão sobre o produto.

Para alcançar o desenvolvimento de um sistema de BI que pudesse integrar a estrutura e beneficiá-la, foram utilizadas técnicas de mineração de dados, que abrangessem desde a

extração de dados, até sua limpeza e visualização, passando por análises e aplicações de ML e DL. Posteriormente os resultados foram tratados e desenvolvidos com o uso conceitos de BI, o que permitiu a produção de uma ferramenta capaz de auxiliar o gestor em diversos aspectos, como a visualização individual e filtrada de diversos tipos de informação e a previsão de demanda com diferentes algoritmos.

A aplicação de mineração de dados em datasets públicos permitiu a realização de visualizações gráficas, que responderam perguntas de negócio no contexto de gestão de Smart Parkings. Essas perguntas vão desde o fluxo diário do estacionamento, horários de maior e menor fluxo, previsão de demanda futura e impacto de fatores como o clima e feriados sobre o fluxo médio comum. Além disso, a mineração foi essencial para identificar que não seria possível produzir análises para todas as variáveis do modelo proposto, como o impacto do clima e feriados sobre o fluxo, sendo necessário a criação de uma etapa adicional de simulação de dados.

A etapa de simulação apresentou uma ferramenta que permitiu a criação de estacionamentos baseada em variáveis de perfil, possibilitando a criação de datasets para diversos estacionamentos. Os datasets simulados contém perfis distintos, com extensa quantidade de dados, e conseguiram abranger todas as variáveis individuais do produto. Além disso, as análises comparativas entre datasets reais e simulados mostraram que a simulação conseguiu produzir padrões de dados com variação similar ao de estacionamentos reais. Esse fator permitiu que simulações pudessem ser usadas em etapas futuras, pois mesmo que os resultados sejam baseados em dados simulados, esses dados possuem um padrão que se assemelha ao real.

Após a etapa de simulação foi possível contar com o acesso à datasets extensos e completos, que possibilitaram a aplicação de algoritmos de IA. O uso de diferentes abordagens de ML e DL sobre o mesmo tipo e conjunto de dados permitiu a comparação entre resultados obtidos por diferentes algoritmos, garantindo assim uma maior fidelidade na previsão gerada por cada abordagem.

Contando com algoritmos para preparação, limpeza e análise de dados, em conjunto com algoritmos de ML e DL, foi possível realizar a criação da plataforma de BI. Essa

plataforma teve por objetivo conectar todos os processos realizados anteriormente em uma única solução, de fácil uso e focada no gestor. Esses objetivos foram alcançados por meio da divisão da solução em diferentes visões, cada uma contendo funcionalidades específicas e visando responder diferentes perguntas de negócio.

A página inicial ao trazer uma visualização diária e um mapa contendo todos os estacionamentos auxilia o gestor a identificar o funcionamento e estado atual de cada um dos seus estacionamentos em tempo real. As páginas de análises básicas buscam auxiliar o gestor na tomada de decisões referentes à identificação do perfil de seus clientes e na identificação de padrões de fluxo e movimentação do estacionamento. A página de análises avançadas traz ao gestor a possibilidade de identificar de forma detalhada o perfil de cada estacionamento e atualizar suas características diretamente na nuvem. Fator esse que beneficia outros módulos do sistema de Smart Parking que podem utilizar essas estatísticas como base para tomada de decisões como, por exemplo, tarifação dinâmica. As páginas de análises preditivas trazem ao gestor a possibilidade de aplicação de soluções de ML e DL modernas em seu sistema, auxiliando na identificação de movimentações futuras. Fornecendo ao gestor, dessa maneira, uma base sólida para a tomada de decisões de forma antecipada sobre cada um de seus estacionamentos.

Com isso, a plataforma desenvolvida conta com muitas características, que por sua vez auxiliam o gestor em diferentes tipos de decisão. Contudo, ela ainda pode ser bastante incrementada por meio da adição de novas páginas, que possam trazer novas análises, visualizações e algoritmos de previsão, agregando assim, ainda mais valor ao produto.

O uso de uma metodologia linear e a escolha por uma única linguagem de programação permitiu que todos os módulos implementados em todas as etapas do trabalho fossem aproveitados e reutilizados na solução final. Além disso, essa característica permite que a solução possa ser complemente gerida e incrementada com o uso de uma única linguagem, o que a torna mais simples de ser implementada em desenvolvimentos futuros relacionados ao produto, visto que com apenas uma linguagem é possível gerir todos os processos.

O uso da nuvem integrado com o desenvolvimento de um padrão de estrutura de dados para o produto permitirão que futuros módulos possam integrar-se à estrutura de maneira

mais simples, utilizando-se de variáveis comuns entre todos os módulos. A funcionalidade de atualizar estatísticas do estacionamento diretamente na nuvem, que visam identificar o perfil de cada estacionamento, permite que módulos possam consumir essa informação em tempo real. Isso traz ao gestor a possibilidade de que suas ações no sistema de BI possam auxiliar na tomada de decisões de outros módulos, os quais podem se beneficiar dessa informação, como os módulos de leilão de vagas e de tarifação dinâmica.

A adaptação dos módulos já existentes para a nova estrutura também serão trabalhos futuros, enquanto que, novos módulos introduzidos poderão beneficiar-se dessa estrutura logo no início da integração, obtendo assim uma adaptação mais simples e rápida ao produto. Outro importante fator a ser considerado para trabalhos futuros está no uso da aplicação desenvolvida sobre uma base de dados real de estacionamentos, que contenha as variáveis individuais do negócio e que poderá ser obtida somente após a introdução do produto Smart Parking no mercado.

Por fim, o uso de aplicações com foco em BI mostrou-se relevante para o contexto de Smart Parkings, sendo essencial para a gestão de produtos e serviços escaláveis que geram grandes volumes de dados. Os benefícios trazidos pelo auxilio na tomada de decisão podem reduzir riscos do produto, aumentar sua rentabilidade e melhorar a qualidade do serviço para clientes. Dessa maneira, a pesquisa buscou agregar valor ao produto por meio do desenvolvimento de um padrão de dados e um módulo de BI, que integrou conceitos de mineração e análise de dados, além de aplicações de ML e DL.

Bibliografia

[1] 2018 kaggle machine learning data science survey. https://www.kaggle.com/kaggle/kaggle-survey-2018. Online; accessed 04 April 2020.

[2] Amazon web services (aws) - cloud computing services. https://aws.amazon.com/. Online; accessed 22 April 2020.

[3] Google cloud: Cloud computing services. https://cloud.google.com. Online; accessed 22 April 2020.

[4] Matplotlib: Visualization with python. https://matplotlib.org. Online; accessed 10 April 2020.

[5] Microsoft azure: Cloud computing services. https://azure.microsoft.com. Online; accessed 22 April 2020.

[6] Abhishek Tiwari. Comparative study on time series forecasting using deep learning models. 2020.

[7] Vivek Agarwal. Research on data preprocessing and categorization technique for smartphone review analysis. *International Journal of Computer Applications*, 131(4):30–36, December 2015.

[8] Tanwir Ahmad, Junaid Iqbal, Adnan Ashraf, Dragos Truscan, and Ivan Porres. Model-based testing using UML activity diagrams: A systematic mapping study. *Computer Science Review*, 33:98–112, August 2019.

[9] A. R. Ajiboye, R. Abdullah-Arshah, H. Qin, and H. Isah-Kebbe. EVALUATING THE EFFECT OF DATASET SIZE ON PREDICTIVE MODEL USING SUPERVISED LEARNING TECHNIQUE. *International Journal of Computer Systems & Software Engineering*, 1(1):75–84, February 2015.

[10] Ghulam Ali, Tariq Ali, Muhammad Irfan, Umar Draz, Muhammad Sohail, Adam Glowacz, Maciej Sulowicz, Ryszard Mielnik, Zaid Bin Faheem, and Claudia Martis. IoT based smart parking system using deep long short memory network. *Electronics*, 9(10):1696, October 2020.

[11] Navneet Anand. Growing population needs smart cities. *The Pioneer*, July 2016.

[12] Gonzalo Antolín, Ángel Ibeas, Borja Alonso, and Luigi dellOlio. Modelling parking behaviour considering users heterogeneities. *Transport Policy*, 67:23–30, September 2018.

[13] Miguel Correia Arnaldo Gouveia. *Deep Learning for Network Intrusion Detection: An Empirical Assessment*. Chapman and Hall/CRC, December 2020.

[14] Taylor B Arnold. kerasR: R interface to the keras deep learning library. *The Journal of Open Source Software*, 2(14):296, June 2017.

[15] Faraz Malik Awan, Yasir Saleem, Roberto Minerva, and Noel Crespi. A comparative analysis of machine/deep learning models for parking space availability prediction. *Sensors*, 20(1):322, January 2020.

[16] Lori Bowen Ayre and Jim Craner. Open data: What it is and why you should care. *Public Library Quarterly*, 36(2):173–184, April 2017.

[17] Stefan Baack. Datafication and empowerment: How the open data movement re-articulates notions of democracy, participation, and journalism. *Big Data & Society*, 2(2):205395171559463, July 2015.

[18] Jonathan Barker and Sabih ur Rehman. Investigating the use of machine learning for smart parking applications. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, October 2019.

[19] David Beniaguev. Historical hourly weather data 2012-2017. `https://www.kaggle.com/selfishgene/historical-hourly-weather-data`. Online; accessed 06 July 2020.

[20] M. Bharati and Bharati Ramageri. Data mining techniques and applications. *Indian Journal of Computer Science and Engineering*, 1, 12 2010.

[21] Michael L. Brodie. *What Is Data Science?*, pages 101–130. Springer International Publishing, Cham, 2019.

[22] T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? February 2014.

[23] Feng Chen, Pan Deng, Jiafu Wan, Daqiang Zhang, Athanasios V. Vasilakos, and Xiaohui Rong. Data mining for the internet of things: Literature review and challenges. *International Journal of Distributed Sensor Networks*, 11(8):431047, January 2015.

[24] DigitalSign. Digitalsign's solution for a smarter city, 2020.

[25] Niklas Donges. A complete guide to the random forest algorithm, Jun 2019.

[26] Michel Dumontier and Tobias Kuhn. Data science – methods, infrastructure, and applications. *Data Science*, 1(1–2):1–5, December 2017.

[27] CEF Digital Connecting Europe. *Context Broker: Collect data from different sources and support smart decisions at the right time.* 2018.

[28] Emilio Ferrara, Pasquale De Meo, Giacomo Fiumara, and Robert Baumgartner. Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, 70:301–323, November 2014.

[29] FIWARE. *OFF STREET PARKING*, 2018.

[30] FIWARE. *ON STREET PARKING*, 2018.

[31] FIWARE. *PARKING HARMONIZED DATA MODELS*, 2018.

[32] I. Ganchev, Zhanlin Ji, and M. ODroma. A generic IoT architecture for smart cities. In *25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communities Technologies (ISSC 2014/CIICT 2014)*. Institution of Engineering and Technology, 2014.

[33] Michael J. Garbade. Understanding k-means clustering in machine learning, Sep 2018.

[34] Disha Garg, Samiya Khan, and Mansaf Alam. Integrative use of IoT and deep learning for agricultural applications. In *Proceedings of ICETIT 2019*, pages 521–531. Springer International Publishing, September 2019.

[35] Aditya Gaur, Bryan Scotney, Gerard Parr, and Sally McClean. Smart city architecture and its applications based on IoT. *Procedia Computer Science*, 52:1089–1094, 2015.

[36] Vladik; Gholamy, Afshin; Kreinovich and Olga Kosheleva. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. 2018.

[37] Cyrill Glockner. Simulators: The key training environment for applied deep reinforcement learning, Feb 2018.

[38] Vishal Goar, Prof S Sarangdevot, Govind Tanwar, and Dr Anand Sharma. Improve performance of extract, transform and load (etl) in data warehouse. *International Journal on Computer Science and Engineering*, 2, 05 2010.

[39] Sam Goundar. *Chapter 4 - Understanding Cloud Computing*. 03 2012.

[40] ACT Government. Smart parking stays. https://www.data.act.gov.au/Transport/Smart-Parking-Stays/3vsj-zpk7. Online; accessed 04 April 2020.

[41] Carlos Eduardo Frickmann Young Guilherme Szczerbacki Besserman Vianna. Em busca do tempo perdido: Uma estimativa do produto perdido em trÂnsito no brasil*. *Revista de Economia Contemporânea (2015) 19(3): p. 403-416 (Journal of Contemporary Economics)*, December 2015.

[42] Ravi Kumar Gupta and Geeta Rani. Machine learning and IoT based real time parking system: Challenges and implementation. *SSRN Electronic Journal*, 2020.

[43] Zozo Hassan, Hesham Ali, and Mahmoud Badawy. Internet of things (iot): Definitions, challenges, and recent research directions. *International Journal of Computer Applications*, 128:975–8887, 10 2015.

[44] Wu He, Gongjun Yan, and Li Da Xu. Developing vehicular data cloud services in the IoT environment. *IEEE Transactions on Industrial Informatics*, 10(2):1587–1595, May 2014.

[45] Karel Horak and Robert Sablatnig. Deep learning concepts and datasets for image recognition: overview 2019. In Xudong Jiang and Jenq-Neng Hwang, editors, *Eleventh International Conference on Digital Image Processing (ICDIP 2019)*. SPIE, August 2019.

[46] Nur Hordri, Siti Yuhaniz, and Siti Mariyam Shamsuddin. Deep learning and its applications: A review. 10 2016.

[47] Susan Imberman. Effective use of the kdd process and data mining for computer performance professionals. pages 611–620, 01 2001.

[48] Indigo. Business intelligence: The next big thing in parking, 2016.

[49] IronHack. Data science vs data analytics, 2020.

[50] JetBrains. Python developers survey 2019 results. `https://www.jetbrains.com/lp/python-developers-survey-2019/`. Online; accessed 1 April 2020.

[51] M.G. Karlaftis and E.I. Vlahogianni. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):387–399, June 2011.

[52] Chamandeep Kaur. The cloud computing and internet of things (IoT). *International Journal of Scientific Research in Science, Engineering and Technology*, pages 19–22, January 2020.

[53] Radosław Klimek and P. Szwed. Formal analysis of use case diagrams. *Comput. Sci.*, 11:115–, 2010.

[54] Vaibhav Kumar and M. L. Deep learning as a frontier of machine learning: A review. *International Journal of Computer Applications*, 182(1):22–30, July 2018.

[55] Yangxin Lin, Ping Wang, and Meng Ma. Intelligent transportation system(ITS): Concept, challenge and opportunity. In *2017 IEEE 3rd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, May 2017.

[56] Armando B. Mendes, Luís Cavique, and Jorge M.A. Santos. DATA MINING PROCESS MODELS: A ROADMAP FOR KNOWLEDGE DISCOVERY. In *Quantitative Modelling in Marketing and Management*, pages 405–433. WORLD SCIENTIFIC, October 2012.

[57] Chandrahas Mishra and D. L. Gupta. Deep machine learning and neural networks: An overview. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 6(2):66, June 2017.

[58] Saraju P. Mohanty, Uma Choppali, and Elias Kougianos. Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consumer Electronics Magazine*, 5(3):60–70, July 2016.

[59] Moh Sukron Mufaqih, Emil R. Kaburuan, and Gunawan Wang. Applying smart parking system with internet of things (IoT) design. *IOP Conference Series: Materials Science and Engineering*, 725:012095, January 2020.

[60] Rajneesh Mungrah and Zarine Cadersaib. Cloud application integration methodology using enterprise application integration. In *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*. IEEE, December 2017.

[61] Mustakim Mustakim. Effectiveness of k-means clustering to distribute training data and testing data on k-nearest neighbor classification. *Journal of Theoretical and Applied Information Technology*, 95:5693–5700, 11 2017.

[62] Derrick Mwiti. Dash for beginners.

[63] Adegboyega Ojo, Edward Curry, and Fatemeh Ahmadi Zeleti. A tale of open data innovations in five smart cities. In *2015 48th Hawaii International Conference on System Sciences*. IEEE, January 2015.

[64] Oladele Tinuke Omolewa, Aro Taye Oladele, Adegun Adekanmi Adeyinka, and Ogundokun Roseline Oluwaseun. Prediction of student's academic performance using k-means clustering and multiple linear regressions. *Journal of Engineering and Applied Sciences*, 14(22):8254–8260, October 2019.

[65] Rajiv Pandey and Manoj Dhoundiyal. Quantitative evaluation of big data categorical variables through r. *Procedia Computer Science*, 46:582–588, 2015.

[66] Waldemar Parkitny. Analysis of depedences between using of parking places and chosen parameters of weather on the example of underground parking in cracow.

IOP Conference Series: Earth and Environmental Science, 95:052011, December 2017.

[67] Andreia Penso Pereira, Bruno Paula Cardoso, and Raul M. S. Laureano. Business intelligence: Performance and sustainability measures in an ETL process. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, June 2018.

[68] André Petermann, Martin Junghanns, Robert Müller, and Erhard Rahm. Graph-based data integration and business intelligence with BIIIG. *Proceedings of the VLDB Endowment*, 7(13):1577–1580, August 2014.

[69] Sabina Pokhrel. Smart parking — an application of ai, Oct 2019.

[70] Hamilton Pozo. Sistema holÔnico de manufatura e a nova gestÃo administrativa para o sÉculo xxi.

[71] Dorian Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 1999.

[72] Sebastian Raschka, Joshua Patterson, and Corey Nolet. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, 11(4):193, April 2020.

[73] SUNIL RAY. 7 regression techniques you should know!, Aug 2015.

[74] Youssra Riahi, , and Sara Riahi and. Big data and big data analytics: concepts, types and technologies. *International Journal of Research and Engineering*, 5(9):524–528, November 2018.

[75] Qusay I. Sarhan and Idrees S. Gawdan. Web applications and web services: A comparative study. *Science Journal of University of Zakho*, 6(1):35, March 2018.

[76] Marko Sarstedt and Erik Mooi. Regression analysis. In *Springer Texts in Business and Economics*, pages 193–233. Springer Berlin Heidelberg, 2014.

[77] PULKIT SHARMA. The most comprehensive guide to k-means clustering you'll ever need, Aug 2019.

[78] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, Mar 2020.

[79] Arisa Shollo. Using business intelligence in it governance decision making. In Markus Nüttgens, Andreas Gadatsch, Karlheinz Kautz, Ingrid Schirmer, and Nadine Blinn, editors, *Governance and Sustainability in Information Systems. Managing the Transfer and Diffusion of IT*, pages 3–15, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[80] Shahan Yamin Siddiqui, Muhammad Adnan Khan, Sagheer Abbas, and Farrukh Khan. Smart occupancy detection for road traffic parking using deep extreme learning machine. *Journal of King Saud University - Computer and Information Sciences*, February 2020.

[81] Smarking. Business intelligence and yield management for parking, 2020.

[82] Katy Stalcup. Aws vs azure vs google cloud market share 2020: What the latest data shows, Nov 2020.

[83] Ralf Staudemeyer and Eric Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 09 2019.

[84] Dirk Stengel, Georgio M. Calori, and Peter V. Giannoudis. Graphical data presentation. *Injury*, 39(6):659–665, June 2008.

[85] Hardik Tanti, Pratik Kasodariya, Shikha Patel, and Dhaval H Rangrej and. Smart parking system based on IOT. *International Journal of Engineering Research and*, V9(05), May 2020.

[86] Intelegain Team. What is business intelligence and how is it different from data science?, Apr 2020.

[87] Gheorghe Tecuci. Artificial intelligence. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):168–180, December 2011.

[88] Antony Unwin. Why is data visualization important? what is important in data visualization? *Harvard Data Science Review*, 2(1), 1 2020. https://hdsr.mitpress.mit.edu/pub/zok97i7p.

[89] Weijie Wang and Yanmin Lu. Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. *IOP Conference Series: Materials Science and Engineering*, 324:012049, March 2018.

[90] Paul Wessel. What is smart parking? January 2016.

[91] Tony Yiu. Understanding random forest, Jun 2019.

[92] Liang Yu, Binbin Li, and Bin Jiao. Research and implementation of CNN based on TensorFlow. *IOP Conference Series: Materials Science and Engineering*, 490:042022, April 2019.

[93] Narmeen Zakaria and Jawwad A. Smart city architecture: Vision and challenges. *International Journal of Advanced Computer Science and Applications*, 6(11), 2015.

[94] Dabao Zhang. A coefficient of determination for generalized linear models. *The American Statistician*, 71(4):310–316, October 2017.

[95] Lelin Zhang, Bang Zhang, Ting Guo, Fang Chen, Peter Runcie, Bronwyn Cameron, and Roger Rooney. *Linking Complex Urban Systems: Insights from Cross-Domain Urban Data Analysis*, pages 221–239. Springer Singapore, Singapore, 2020.

[96] Xiaofei; Chen Jun; Yan Xingchen; Wang Tao. Zhu, Yating; Ye. Impact of cruising for parking on travel time of traffic flow. *Sustainability 12, no. 8: 3079.*, April 2020.

[97] Yangyong Zhu and Yun Xiong. Towards data science. *Data Science Journal*, 14(0):8, May 2015.