UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

VERUSKA BORGES SANTOS

UM ENSEMBLE BASEADO EM ÁRVORES DE DECISÃO PARA PREDIZER A OCORRÊNCIA DE AGLOMERADOS DE ÔNIBUS

CAMPINA GRANDE – PB
2020

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Um Ensemble baseado em Árvores de Decisão para Predizer a Ocorrência de Aglomerados de Ônibus

Veruska Borges Santos

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sistemas de Informação

Carlos Eduardo Santos Pires (UFCG)
Dimas Cassimiro do Nascimento Filho (UFRPE)
(Orientadores)

Campina Grande, Paraíba, Brasil

S237e Santos, Veruska Borges.
Um ensemble baseado em árvores de decisão para predizer a ocorrência de aglomerados de ônibus / Veruska Borges Santos. - Campina Grande, 2021.
97 f. : il. Color

Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2020.
"Orientação: Prof. Dr. Carlos Eduardo Santos Pires, Prof. Dr. Dimas Cassimiro do Nascimento Filho".
Referências.

1. Transporte Público. 2. Aglomerado de Ônibus. 3. Aprendizagem de Máquina. 4. HEADWAY. I. Pires, Carlos Eduardo Santos. II. Nascimento Filho. Dimas Cassimiro do. III. Título.

CDU 004:656.121(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA ITAPUANA SOARES DIAS CRB-15/93

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO CIENCIAS DA COMPUTACAO
Rua Aprígio Veloso, 882, - Bairro Universitário, Campina Grande/PB, CEP 58429-900

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

VERUSKA BORGES SANTOS

UM ENSEMBLE BASEADO EM ÁRVORES DE DECISÃO PARA PREDIZER A OCORRÊNCIA DE AGLOMERADOS DE ÔNIBUS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: 06/10/2020

Prof. Dr. CARLOS EDUARDO SANTOS PIRES, Orientador, UFCG

Prof. Dr. DIMAS CASSIMIRO DO NASCIMENTO FILHO, Orientador, UFRPE

Prof. Dr. LEANDRO BALBY MARINHO, Examinador Interno, UFCG

Prof. Dr. KIEV SANTOS DA GAMA, Examinador Externo, UFPE

Documento assinado eletronicamente por **CARLOS EDUARDO SANTOS PIRES**, **PROFESSOR 3 GRAU**, em 25/02/2021, às 08:35, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **KIEV SANTOS DA GAMA**, **Usuário Externo**, em 25/02/2021, às 10:26, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Dimas Cassimiro do Nascimento Filho**, **Usuário Externo**, em 25/02/2021, às 11:18, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **LEANDRO BALBY MARINHO**, **PROFESSOR 3 GRAU**, em 26/02/2021, às 14:16, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **1300351** e o código CRC **9331817D**.

Referência: Processo nº 23096.040502/2020-17 SEI nº 1300351

Resumo

Atrasos nas viagens e superlotação de ônibus são algumas das insatisfações diárias dos usuários de transporte público. Esses problemas podem estar associados aos aglomerados de ônibus, eventos que ocorrem quando dois ou mais ônibus estão executando a mesma rota juntos, ou seja, chegam no mesmo horário nas paradas de ônibus. Devido à natureza estocástica do tráfego, um horário programado estático não é eficaz para evitar a ocorrência desses eventos; assim, são necessárias ações preventivas para melhorar a confiabilidade do sistema de transporte público. Os trabalhos já propostos no contexto preditivo de aglomerados de ônibus apresentam ainda limitações relacionadas à frequência ou privacidade dos dados utilizados, além da eficácia limitada à contextos específicos. Assim, este trabalho propõe um *ensemble* baseado em modelos de árvores de decisão para prever a formação de aglomerados de ônibus. O *ensemble* utiliza dados de geolocalização de ônibus, dados programados, dados de clima, da situação de tráfego e é composto pelos modelos Random Forest, XGBoost e CatBoost. Além disso, uma técnica de aprendizagem incremental é incorporada ao modelo proposto para continuamente atualizá-lo de acordo com a chegada de novos dados em tempo real. A eficácia do modelo é demonstrada com o uso de dados reais de duas cidades brasileiras e comparada com quatro modelos competidores: Regressão Linear, Regressão Logística, Support Vector Machine e Relevance Vector Machine. De acordo com os resultados, o modelo proposto é capaz de alcançar uma eficácia entre $73\% - 80\%$, superior aos modelos competidores avaliados, e pode ser usado para prever a formação de aglomerados de ônibus em tempo real até dez paradas antes da ocorrência.

Palavras-chave: transporte público, aglomerado de ônibus, aprendizagem de máquina, *headway*.

Abstract

Travel delays and bus overcrowding are some of the daily dissatisfactions of public transportation users. These problems may be caused by bus bunching, an event that occurs when two or more buses are running the same route together, i.e., arriving at the same time at the bus stops. Due to the stochastic nature of the traffic, a static schedule is not effective to avoid the occurrence of these events; thus, preventive actions are necessary to improve the reliability of the public transportation system. The works already proposed in the predictive context of bus bunching still have limitations related to the frequency or privacy of the data used, in addition to the effectiveness limited to specific contexts. Therefore, we propose a decision tree-based ensemble model to predict bus bunching. We use an ensemble of Random Forest, XGBoost and CatBoost models with buses geolocation, scheduled, weather and traffic situation data. In addition, an incremental learning technique was incorporated into the proposed model to continuously update it according new data arrives in real-time. The efficacy of the proposed model has been demonstrated using real data sets of two brazilian cities and has been compared with four competitors: Linear Regression, Logistic Regression, Support Vector Machine and Relevance Vector Machine. According to the results, the proposed model can achieve an efficacy between $73\% - 80\%$, higher than the evaluated competitors models, and can be used to predict bus bunching in real-time up to ten stops before their occurrence.

Keywords: public transportation, bus bunching, machine learning, headway.

Agradecimentos

A Deus, todas as minhas realizações. Não podia ser diferente começar agradecendo a quem é minha base e fortaleza, minha luz a seguir. A fé e persistência me permitiram superar muitas dificuldades e limitações.

À minha família, meus pais Maria e Natal, que me ensinaram os primeiros caminhos da educação e me permitiram seguir os meus sonhos, da forma deles. Às minhas irmãs, Vera e Verônica, que de forma singular, sempre me incentivaram a manter o foco nos estudos e me ajudaram na caminhada até aqui. Aos meus sobrinhos, Bruna, Beatriz, Érica, Lucas e Elisa, que me permitem vivenciar todas as vantagens de ser tia, me lembram o quanto a infância é deliciosa e com quem divido meus momentos de diversão.

Aos meus orientadores, Carlos e Dimas, que são exemplos de profissionais e seres humanos, sempre demonstram muita dedicação e responsabilidade com suas atividades. Ao professor Carlos, por ter aceitado a minha troca de orientação. Já trabalhava em seus projetos, conhecia sua dinâmica de trabalho e queria continuar no mesmo caminho. Ao professor Dimas, por ter sido um grande amigo e aceitado participar da minha orientação. Quanto conhecimento vocês me passaram no decorrer desta pesquisa. Gratidão por todo incentivo, apoio e paciência neste período.

Aos meus colegas do Laboratório de Qualidade de Dados (LQD). Andreza, que dividiu diariamente esta pesquisa comigo, apresentando sugestões, alternativas e, principalmente, me incentivando. À Brasileiro, Nóbrega, Diego e Igor pelas contribuições e correções sugeridas, e principalmente pelos momentos de diversões. À Demetrio, Lucas Barros, Lucas Oliveira e Pedro pelo período que convivemos e todas as experiências vivenciadas, além de muitas risadas. As viagens que fizemos, os joguinhos na hora do almoço e todas as aventuras durante esta jornada foram incríveis e essenciais para recarregar as energias. O ambiente do LQD é muito acolhedor.

A todos os meus colegas da UFCG, dentre eles: Renato, Ítalo, Ricardo, Jair, Vinícius, Allan e Diogo, pelas contribuições diretas e indiretas nesta pesquisa, bem como os muitos momentos de descontração, em especial, os momentos de *board games*.

Ao projeto INES e à CAPES, pelo incentivo e suporte financeiro.

A todos os colaboradores do Computação@UFCG e aos professores que um dia tive a honra de ser aluna, desde a infância. Tenho muita gratidão e admiração pelos mestres educadores.

Conteúdo

Lista de Símbolos

AFC - *Automatic Fare Collection*

AM - *Aprendizagem de Máquina*

API - *Application Programming Interface*

AR - *Autoregressivo*

AVL - *Automatic Vehicle Location*

BB - *Bus Bunching*

BULMA - *BUs Line MAtching*

BUSTE - *BUs Stop Ticketing Estimation*

CatBoost - *Categorical Boosting*

F1 - $F_{measure}$

FE - *Feature Engineering*

FN - *Falso Negativo*

FP - *Falso Positivo*

GPS - *Global Positioning System*

GTFS - *General Transit Feed Specification*

HP - *Horizonte de Predição*

INES - *Instituto Nacional de Ciência e Tecnologia para Engenharia de Software*

LS-SVM - *Least-Squares Support Vector Machine*

LSTM - *Long Short-Term Memory*

MAPE - *Mean Absolute Percentage Error*

MSE - *Mean Squared Error*

RBF - *Radial Basis Function*

RF - *Random Forest*

RK - *Regressão Kernel*

RLi - *Regressão Linear*

RLo - *Regressão Logística*

RMSE - *Root Mean Square Error*

RNA - *Rede Neural Artificial*

RVM - *Relevance Vector Machine*

SVM - *Support Vector Machine*

TP - *Tempo de Predição*

VN - *Verdadeiro Negativo*

VP - *Verdadeiro Positivo*

XGBoost - *eXtreme Gradient Boosting*

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Os transportes públicos (ônibus, metrôs e trens) ainda são os meios mais acessíveis financeiramente para a população brasileira, garantindo o direito de mobilidade. Diariamente, mais de 60 milhões de brasileiros, em especial a parcela mais pobre da população, precisam usar o transporte público para locomover-se de casa até o local de trabalho ou de estudo [3].

Especificamente, 86% das viagens urbanas em transporte público no Brasil são feitas por ônibus [17]. Entretanto, a ineficiência, má qualidade e a falta de confiabilidade neste serviço estão causando a crescente insatisfação dos usuários, que, progressivamente, optam pelo transporte privado quando possível [30]. Como relatado pela Associação Nacional das Empresas de Transportes Urbanos (NTU), a quantidade de passageiros transportados por ônibus no Brasil caiu cerca de 40% nos últimos 30 anos [41]. Enquanto isso, o aumento no uso dos transportes privados, além da geração de perdas financeiras para as empresas de transporte público, prejudica a mobilidade urbana, contribuindo para o aumento de congestionamentos, poluição e acidentes [17; 26].

Essa ineficiência no sistema de transporte público é decorrente da falta de um planejamento eficaz de mobilidade urbana que acompanhe o desenfreado crescimento populacional nas cidades desde o último século. Esse planejamento deve visar ainda o fornecimento de um serviço de qualidade e com baixo custo para os passageiros. Atualmente, as insatisfações mais relatadas por esses usuários concentram-se em tarifas elevadas, trânsito caótico, deslocamento entre casa e trabalho muito demorado e transportes coletivos lotados [50].

Um dos problemas relacionados ao transporte público é o aglomerado de ônibus (em inglês, *Bus Bunching* - BB), um evento que consiste de dois ou mais ônibus executando

a mesma rota juntos, ou seja, separados por um *headway* muito pequeno. *Headway* é a diferença entre o tempo de chegada de dois ônibus da mesma rota na mesma parada de ônibus [36]. Para exemplificar, na Figura 1.1 são exibidos três ônibus da rota 232 às 06:32 am em Curitiba - Paraná. Embora os ônibus A e B estejam a aproximadamente 800 m de distância um do outro, considerando que *headway* $= 1$ (minuto), eles estão aglomerados, pois ambos chegam à mesma parada de ônibus p (situada logo à frente do ônibus A) com menos de um minuto de diferença. Por outro lado, os ônibus B e C não estão aglomerados porque chegam na parada p com cerca de 20 minutos de diferença, $headway = 20$.

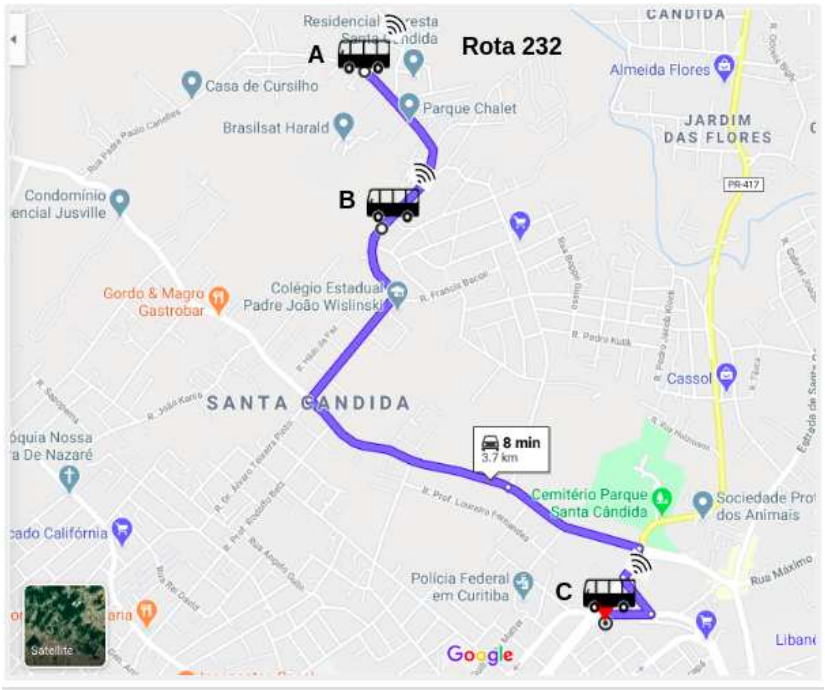

Figura 1.1: Exemplo ilustrativo da ocorrência de aglomerados de ônibus envolvendo os ônibus A e B, na cidade de Curitiba, em 27/05/2019. Mapa de fundo recuperado do Google Maps.

A ocorrência dos aglomerados compromete a qualidade do serviço de ônibus, porque os ônibus envolvidos descumprem seus horários programados. Assim, esse evento é considerado uma das causas de insatisfação dos passageiros, como o aumento do tempo de espera e da viagem, além do desconforto na viagem devido à superlotação. Uma análise na cidade de Curitiba mostrou que, em média, os aglomerados ocorreram em 4% das viagens diárias em maio de 2019, atingindo até 11% nas viagens em 21/05/2019, por exemplo.

1.1 Motivação

Para mitigar os problemas relacionados à qualidade do serviço de ônibus, uma previsão efetiva da ocorrência dos aglomerados de ônibus em tempo real e a determinação de seus fatores de influência podem facilitar a tomada de decisão dos agentes de trânsito e minimizar a ocorrência desse evento. No entanto, prever este evento é considerado um desafio devido ao acontecimento de eventos estocásticos no trânsito (por exemplo, acidentes e condições climáticas podem influenciar a ocorrência dos aglomerados [59]). Além disso, espera-se que um modelo de previsão útil tenha um horizonte de previsão suficientemente longo, ou seja, preveja o evento algumas paradas antes de sua ocorrência, a fim de permitir a implementação de ações preventivas pelos operadores de trânsito [52].

Dessa forma, foi identificada como possibilidade de pesquisa científica a proposição de uma abordagem para predição de aglomerados de ônibus que considere múltiplas fontes de dados e modelos de aprendizagem de máquina (AM). Tal abordagem deve: i) maximizar a eficácia e o horizonte de predição e ii) minimizar o tempo de execução das predições.

1.2 Relevância

Alguns modelos de predição de aglomerados de ônibus já foram desenvolvidos devido à crescente necessidade de informações de transporte público em tempo real. Por exemplo, os trabalhos [36; 2; 59; 52] propõem modelos de predição de aglomerados de ônibus aplicando técnicas diferentes e horizontes de predição variados (de duas a 15 paradas à frente). Além disso, esses trabalhos empregam modelos de AM e usam não apenas dados dos ônibus - GPS (*Global Positioning System*), mas também dados de cartões de passagens - AFC (*Automated Fare Collection*) e dados programados - GTFS (*General Transit Feed Specification*).

A principal desvantagem desses trabalhos é considerar suposições distantes da realidade, o que dificulta a aplicação dos modelos em cidades onde as seguintes suposições não são satisfeitas e os dados são privados e não são disponibilizados. A primeira delas é considerar que um ônibus da mesma rota nunca ultrapassa o outro, mas em um cenário real, devido à incerteza do tráfego, os ônibus podem ultrapassar os outros [49], uma vez permitido pela empresa de transporte.

A segunda suposição é o uso de fontes de dados de alta frequência (por exemplo, GPS) para obter uma boa eficácia. Na prática, a frota de ônibus de algumas cidades está equipada com tecnologias de baixa frequência (envio de dados em um intervalo de 30 a 60 segundos) para reduzir o custo de armazenamento e transmissão de dados [48], o que torna a predição dos aglomerados de ônibus com esses dados esparsos mais desafiadora.

A terceira suposição é o uso de dados privados, como dados de AFC. Em geral, esses dados não estão disponíveis ao público [7] porque contêm informações de receita bruta da empresa e informações pessoais de passageiros, como rotas percorridas. Isso restringe a aplicação do modelo de predição a cidades onde as empresas apresentam resistência no fornecimento desses dados.

Por fim, alguns modelos [36; 52] apresentam resultados de baixa precisão, ou seja, um número significativo de predições incorretas dos aglomerados de ônibus, que prejudicam a confiabilidade do modelo para os usuários da informação, como os agentes de trânsito.

Sendo assim, é necessário um modelo geral e eficaz de predição de aglomerados de ônibus que i) considere como entrada diferentes fontes de dados normalmente disponíveis na maioria das cidades; ii) seja capaz de prever os aglomerados n paradas antes de sua ocorrência; iii) apresente boa precisão, ou seja, minimize alertas falsos da ocorrência dos aglomerados de ônibus; e iv) considere dados em tempo real para atualizar continuamente o modelo.

1.3 Objetivos

Para motivar a elaboração desta pesquisa, a seguinte hipótese geral foi considerada: *a utilização de múltiplas fontes de dados aplicadas a modelos de AM melhoram a eficácia da predição de aglomerados de ônibus, mantendo a eficiência, em termos de predições em tempo real e horizonte de predição.*

Com base nisso, o objetivo geral deste trabalho é contribuir com a confiabilidade dos sistemas de transporte público, por meio da melhoria dos modelos preditivos da ocorrência de aglomerados de ônibus em relação à eficácia, sem prejudicar o horizonte e o tempo de predição.

Visando alcançar este objetivo geral, os seguintes objetivos específicos são definidos:

- Melhorar a eficácia da predição de aglomerado de ônibus por meio da utilização de modelos de AM;
- Descrever o processo de atualização do modelo para considerar a predição de múltiplos passos;
- Combinar abordagens da literatura para integrar as fontes de dados a serem utilizadas no modelo proposto;
- Avaliar a influência da utilização de múltiplas fontes de dados no modelo proposto para predizer os aglomerados;
- Aplicar a aprendizagem incremental ao modelo proposto;
- Investigar a ocorrência de padrões nos dados históricos;
- Comparar a eficácia do modelo proposto com a eficácia de outras abordagens de predição de aglomerados de ônibus da literatura.

1.4 Contribuições

Neste trabalho, é apresentado um modelo para predição de aglomerados de ônibus, composto por um conjunto de árvores de decisão para prever os aglomerados para as próximas n paradas de ônibus (predição de múltiplos passos). Particularmente, foram combinados os modelos-base *Random Forest* (RF), XGBoost e CatBoost em um único modelo (*ensemble*) para realizar as predições com maior eficácia.

Para melhorar a eficácia do modelo, foram utilizadas como entrada diferentes fontes de dados: GPS, GTFS, dados de clima e de situação do trânsito. Até a elaboração desta solução, não foram encontrados trabalhos relacionados que usam dados de clima e de trânsito com abordagens de AM para esse contexto. Devido à influência dessas duas fontes de dados nos serviços de trânsito [25; 14; 19], presume-se que elas ajudarão a antecipar a predição dos aglomerados.

Assim, as principais contribuições deste trabalho são:

- Um modelo de predição de aglomerados de ônibus, composto por modelos baseados em árvore de decisão e combinados em uma abordagem *ensemble* de votação, considerando quatro fontes de dados diferentes como entrada;
- Um fluxo de execução empregado para integração das fontes de dados utilizadas;
- Uma avaliação abrangente do modelo preditivo proposto, considerando diferentes cenários; e
- A descoberta de padrões de influência, como o aumento da ocorrência de aglomerados de ônibus em dias e horários específicos da semana.

A avaliação do modelo preditivo proposto considera os seguintes aspectos: i) uso de diferentes modelos, como modelos baseados em árvores de decisão e regressão; ii) combinação de diferentes técnicas de *ensemble*, como *boosting* e *stacking*; iii) presença de dados desbalanceados; e iv) uso de uma técnica de aprendizagem incremental.

Esta pesquisa faz parte do projeto "Construção de uma Infraestrutura de Dados de Mobilidade Urbana", associado ao INES (Instituto Nacional de Ciência e Tecnologia para Engenharia de Software). Até o presente momento, os seguintes indicadores de pesquisa preliminares foram obtidos:

- Apresentação desta pesquisa no Workshop de Teses e Dissertações do XXXIV Simpósio Brasileiro de Banco de Dados (2019);
- Apresentação de uma ferramenta de monitoramento e detecção de aglomerados de ônibus em tempo real na Sessão de Ferramentas e Aplicações do XXXIV Simpósio Brasileiro de Banco de Dados (2019);
- Submissão de artigo intitulado *"A Decision Tree Ensemble Model for Predicting Bus Bunching"* ao *The Computer Journal* (2020).

1.5 Organização do Trabalho

A estrutura deste documento está organizada como segue. No Capítulo 2, é apresentada a fundamentação teórica necessária para compreender o conteúdo do trabalho, como os conceitos relacionados às fontes de dados de transporte público e os conceitos de aprendizagem

de máquina. No Capítulo 3, são apresentados os trabalhos relacionados à esta pesquisa. No Capítulo 4, é apresentada a formalização do problema e a solução proposta de predição de aglomerados de ônibus, incluindo o uso da técnica de aprendizagem incremental e o fluxo de execução de predição para múltiplas paradas. No Capítulo 5, é apresentada a avaliação experimental do modelo proposto nesta pesquisa, seguida da discussão dos resultados. Por fim, no Capítulo 6, são apresentadas as conclusões da pesquisa e as perspectivas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os conceitos necessários para a compreensão do trabalho. Na Seção 2.1, é fornecida uma descrição das fontes de dados relacionadas ao transporte público e utilizadas neste trabalho. Na Seção 2.2, são apresentados os conceitos fundamentais sobre modelos de aprendizagem de máquina e algumas técnicas associadas. Na Seção 2.3, são apresentadas as considerações finais do capítulo.

2.1 Dados Relacionados ao Transporte Público

Combinação, conexão e integração de sistemas e infraestruturas são fundamentais para uma cidade ser inteligente [39]. Com base nisso, serão apresentadas, nesta seção, quatro tipos de fontes de dados comumente disponíveis no contexto das cidades inteligentes, especialmente de transporte público, voltadas para o desenvolvimento da mobilidade urbana: (i) geolocalização dos ônibus; (ii) rotas e horários programados para operação dos ônibus; (iii) situação do trânsito; e (iv) situação climática. A seguir, são apresentados os principais atributos, exemplos de dados e outras características de cada fonte.

2.1.1 Geolocalização dos Ônibus

Esta fonte de dados contém a geolocalização de cada ônibus na cidade. Os sistemas de localização automática de veículos (*Automatic Vehicle Location* - AVL) rastreiam a posição dos veículos geralmente usando o Sistema de Posicionamento Global (*Global Positioning Sys-*

tem - GPS). Os dispositivos GPS nos ônibus enviam continuamente os dados de cada veículo para um servidor, permitindo construir uma visualização em tempo real da movimentação dos ônibus, além de criar um registro histórico das trajetórias percorridas. Cada dado enviado contém, geralmente, informações sobre a rota r do ônibus, a identificação do veículo d, a latitude lat, a longitude lon e o horário de envio t do dado de cada ônibus para o servidor. Uma amostra desses dados no formato JSON pode ser vista na Figura 2.1. A frequência de envio de cada dado de GPS geralmente é de 30 a 60 segundos [48]. Em geral, os dados de GPS são disponibilizados pelas empresas de ônibus por meio de uma API (*Application Programming Interface*).

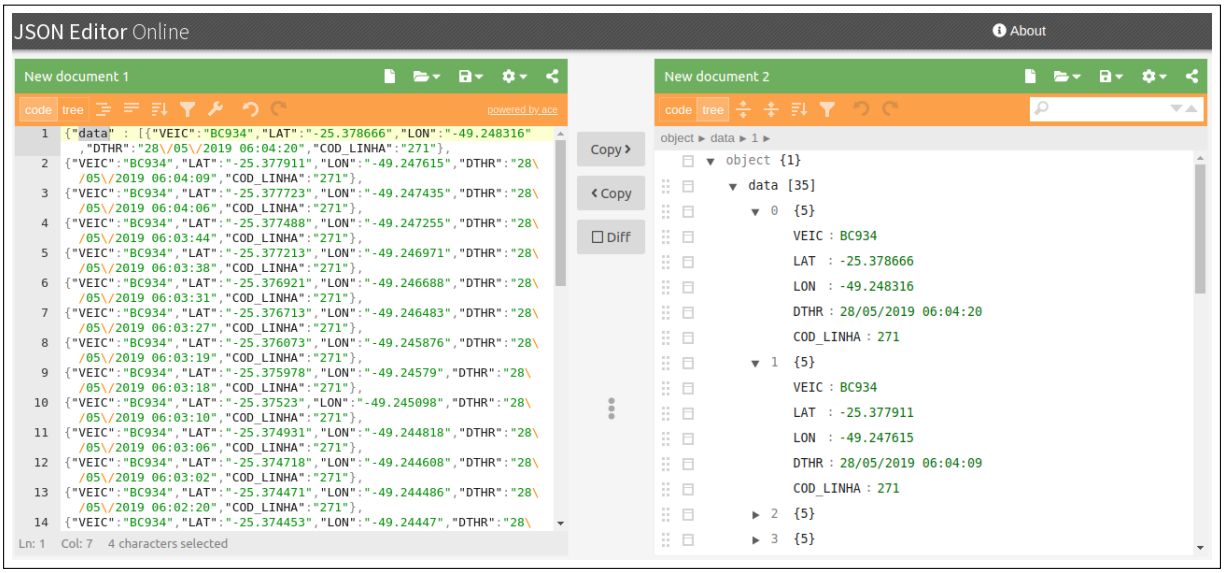

Figura 2.1: Amostra dos dados dos ônibus da cidade de Curitiba utilizando o *JSON Editor Online*, ferramenta para visualização de arquivos JSON.

2.1.2 Rotas e Horários Programados

Esta fonte de dados representa as informações de trânsito programadas para serem executadas pelo transporte público. Um padrão comumente adotado para a descrição dessas informações, como rotas e horários programados, é a Especificação Geral do Feed de Trânsito[1] (*General Transit Feed Specification* - GTFS), que define um formato para os arquivos a serem fornecidos por um operador de ônibus ou autoridade de trânsito da cidade.

[1] *https://developers.google.com/transit/gtfs*

O GTFS é composto por uma série de arquivos (Figura 2.2), em que cada um representa um aspecto específico das informações de trânsito relacionadas ao transporte público, a exemplo do ônibus: rotas, paradas, viagens, horários de chegada nas paradas, entre outros dados programados.

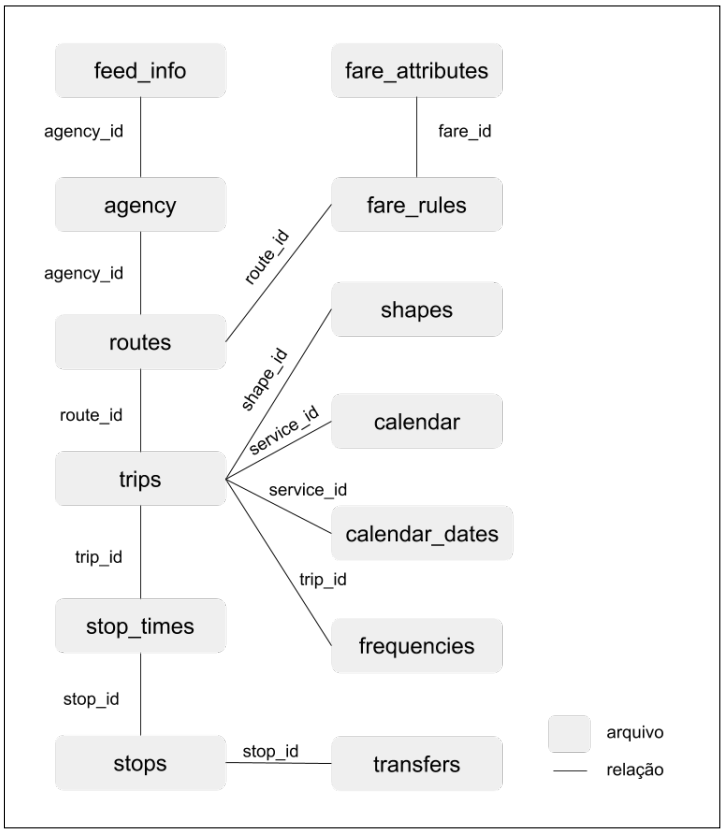

Figura 2.2: Diagrama representando os arquivos do GTFS e suas relações.

São utilizados neste trabalho os seguintes arquivos do padrão GTFS: rotas (*routes*), viagens (*trips*), horários nas paradas (*stop_times*), paradas de ônibus (*stops*), caminho predefinido (*shapes*) e calendário (*calendar*). Exemplos do conteúdo destes arquivos podem ser vistos no Apêndice A.

O arquivo de rotas representa as informações relacionadas às rotas a serem seguidas pelos veículos, como nome da rota e descrição. O arquivo de viagens representa as informações referentes à variação do serviço sobre a rota programada, contendo um identificador, nome, direção e identificador dos outros arquivos. O arquivo de horários nas paradas representa os horários em que, durante uma viagem, o ônibus é esperado estar nas paradas definidas. O arquivo de paradas representa as informações relacionadas às paradas de ônibus, como

código da parada, nome e geolocalização da mesma.

Além disso, o arquivo *shapes* representa a geolocalização detalhada das viagens programadas, contendo um identificador e um conjunto de coordenadas geográficas (latitudes e longitudes), também chamadas de pontos. Cada ponto contém também a informação da sequência, contada a partir do ponto inicial da viagem, e da distância (em metros, em geral) até o ponto inicial. Por fim, o arquivo de calendário define os períodos de funcionalidade dos serviços descritos nos arquivos, como dia da semana e datas.

2.1.3 Situação do Trânsito

Essa fonte de dados contém informações relacionadas à situação do trânsito nas ruas, como ocorrências de acidentes, congestionamentos e vias fechadas. Uma amostra dos dados pode ser vista na Figura 2.3. Essas informações geralmente são coletados por aplicativos de *crowdsourcing*, como Google Maps[2] (Figura 2.4) e Waze[3] (Figura 2.5), onde as informações são inseridas pelos usuários.

location	reliability	reportDescription	speed	subtype	type
{"x":-49.30404,"y":-25.683273}	6	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.301172,"y":-25.685528}	6	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.305859,"y":-25.685267}	7	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.301172,"y":-25.685528}	6	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.302237,"y":-25.684177}	6	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.304236,"y":-25.684836}	6	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.30163,"y":-25.685679}	7	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.300758,"y":-25.683693}	7	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.301222,"y":-25.683845}	7	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.302743,"y":-25.684347}	6	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.304727,"y":-25.684998}	7	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.303258,"y":-25.684516}	7	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.30174,"y":-25.684015}	7	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED
{"x":-49.300662,"y":-25.685359}	6	Local em construção.	0	ROAD_CLOSED_EVENT	ROAD_CLOSED

Figura 2.3: Amostra dos dados de trânsito da cidade de Curitiba.

No mapa do Google Maps, exibido na Figura 2.4, as cores representam o nível do tráfego, sendo verde representando o tráfego normal e sem atrasos, laranja representando quantidade média de tráfego e vermelho representando atrasos no tráfego. Quanto mais escuro o vermelho, mais lenta a velocidade do tráfego na via. Os símbolos representam incidentes, como

[2] *https://www.google.com/maps/*

[3] *https://www.waze.com/pt-BR/livemap*

batidas, construções e vias fechadas [23].

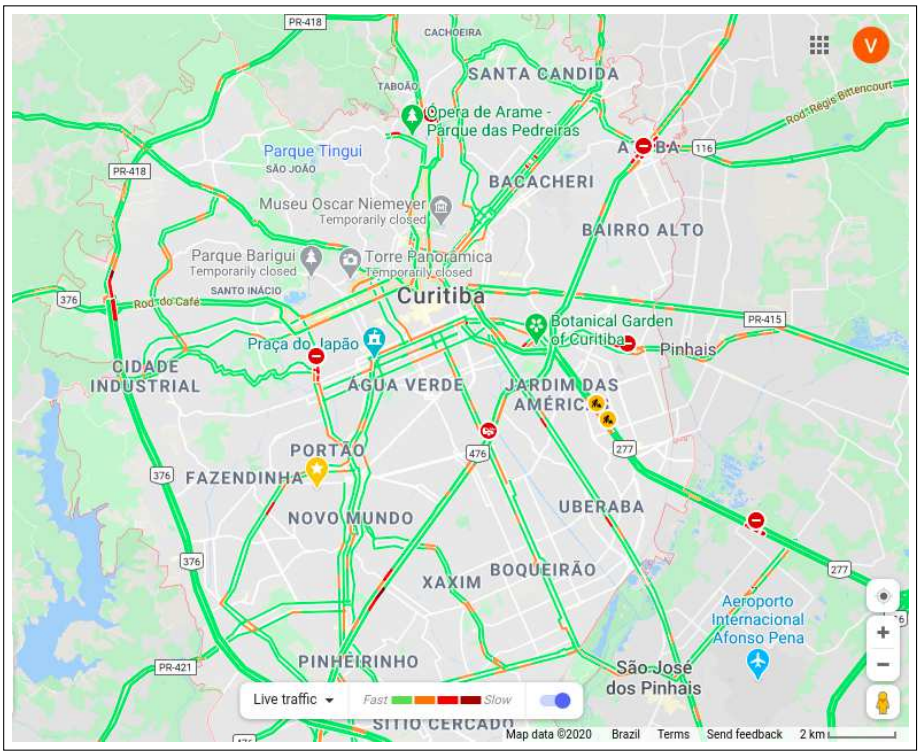

Figura 2.4: Mapa interativo da cidade de Curitiba com informações do trânsito na aplicação Google Maps. Dia 15 de julho de 2020.

No mapa do Waze, exibido na Figura 2.5, a cor vermelha representa o nível do tráfego: quanto mais escuro, mais lenta a velocidade do tráfego na via. Os símbolos representam incidentes, como batidas, construções, postos policiais, buracos e vias fechadas [55].

2.1.4 Situação Climática

De acordo com alguns estudos [25; 54; 20], as condições climáticas adversas influenciam diretamente na qualidade do funcionamento do transporte público. Dessa forma, a fonte de dados sobre situação climática a ser descrita a seguir diz respeito ao clima da cidade, especificamente à precipitação pluviométrica.

Estes dados climáticos são disponibilizados por centros de monitoramento das estações pluviométricas automáticas distribuídas pelos bairros da cidade. Nas cidades brasileiras, por exemplo, essas estações estão espalhadas em regiões em situação de risco no que se refere a ocorrência de desastres naturais. Tais dados podem ser coletados dos centros de monito-

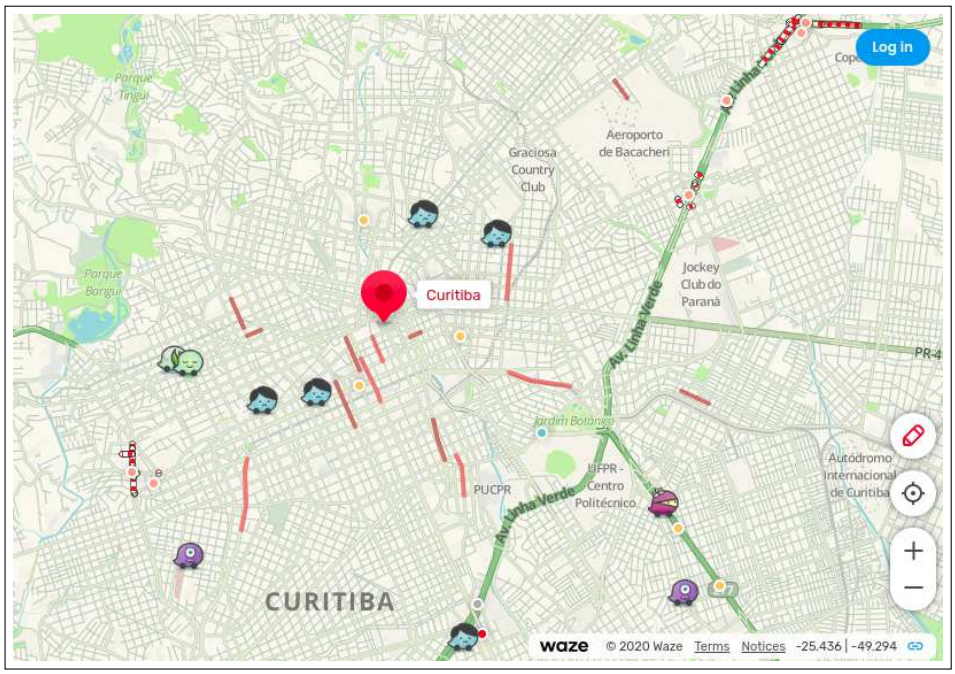

Figura 2.5: Mapa interativo da cidade de Curitiba com informações do trânsito na aplicação Waze. Dia 15 de julho de 2020.

ramento como a CEMADEN[4] (Centro Nacional de Monitoramento e Alertas de Desastres Naturais). Uma amostra dos dados pode ser vista na Figura 2.6.

2.2 Conceitos de Aprendizagem de Máquina

Esta seção apresenta conceitos relacionados ao Aprendizado de Máquina, como a definição dos modelos, técnicas para combiná-los (*ensemble*), predição de múltiplos passos a frente e aprendizagem incremental.

2.2.1 Definição de Aprendizagem de Máquina

Os modelos de AM - um subgrupo da área de Inteligência Artificial - são definidos como um processo automatizado que extrai padrões de dados, isto é, captura a relação entre variáveis descritivas e uma variável alvo [27]. Esses modelos basicamente usam fórmulas matemáticas para aprender uma função ideal $f : X \rightarrow Y$, com $X \subset \mathbb{R}^n$ e $Y \subset \mathbb{R}$, que melhor representa

[4] *http://www.cemaden.gov.br/mapainterativo/*

municipio	codEstacao	uf	nomeEstacao	latitude	longitude	datahora	valorMedida
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 00:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 01:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 02:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 03:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 04:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 05:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 06:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 07:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 08:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 09:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 10:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 11:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 12:40:00.0	0
CURITIBA	410690201A	PR	Butiatuvinha	-49,36184	-25,41118	2019-05-01 13:40:00.0	0

Figura 2.6: Amostra dos dados pluviométricos da cidade de Curitiba.

o problema. Neste contexto, o conjunto X representa as variáveis de entrada[5] (por exemplo, a localização do ônibus) e o conjunto Y se refere ao valor a ser previsto (por exemplo, "os ônibus vão aglomerar" ou "os ônibus não vão aglomerar").

O objetivo dos modelos de AM é estimar uma função $g : X \rightarrow Y \mid g \approx f$, a partir de múltiplas iterações sobre um conjunto de dados de treinamento $D = \{(x_1, y_1), .., (x_{|D|}, y_{|D|})\}$, extraídos de uma base de dados. Esta base geralmente é dividida em duas: treino e teste. Os dados de treino são utilizados para gerar o modelo preditivo e os dados de teste são utilizados para validar este modelo gerado. Nesse caso, o processo chama-se aprendizado supervisionado, pois, como cada instância de dados possui o seu valor real y_i, a cada iteração sobre os dados de treino, o erro do modelo gerado é calculado até atingir a convergência.

Em geral, essa função estimada g é chamada de modelo. Existem algumas variações de modelos (por exemplo, árvore de decisão e *multilayer perceptron*) que podem ser combinadas para serem usadas em diferentes cenários, dependendo do problema.

2.2.2 Abordagem *Ensemble*

Em AM, um *ensemble* representa um conjunto de modelos, chamados aprendizes/modelos-base (*base learners*), construídos para obter melhores resultados pela combinação das predições desses modelos. Os modelos-base podem empregar estruturas diferentes, ser treinados com diferentes subamostras de dados e combinados de maneiras diferentes. O resultado

[5] As variáveis de entrada precisam ser convertidas para valores numéricos.

do *ensemble* é a combinação, geralmente média e moda, das predições fornecidas pelos modelos-base.

Em geral, os modelos-base comumente utilizados são classificados como *weak learners*, ou seja, modelos com esquemas de aprendizado simples [56]. O oposto são *strong learners*, ou seja, modelos mais robustos, criados para alcançar alta eficácia nos dados de teste. Exemplos de técnicas de *ensemble* são mostrados a seguir:

- *Votação*: representa diversos modelos-base (geralmente de tipos diferentes) combinados usando alguma função estatística, como média, mediana e ponderação das predições (Figura 2.7);

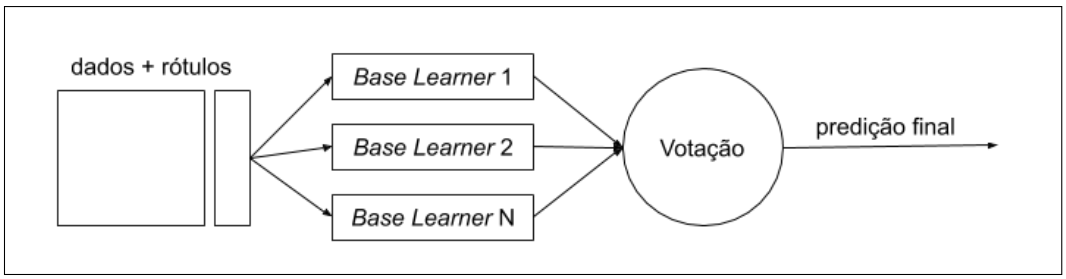

Figura 2.7: Exemplo de *ensemble* com abordagem de votação.

- *Bootstrap aggregation (Bagging)*: refere-se a diversos modelos-base (normalmente do mesmo tipo) treinados em paralelo e utilizando diferentes subconjuntos de dados, escolhidos aleatoriamente com reposição (amostragem *bootstrap*). Esta técnica é exemplificada na Figura 2.8;

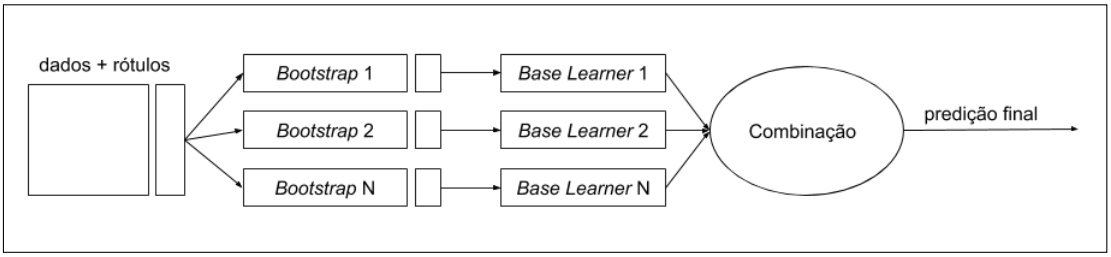

Figura 2.8: Exemplo de *ensemble* com abordagem *bagging*.

- *Boosting*: corresponde a diversos modelos-base combinados sequencialmente, onde cada modelo tem como entrada as predições do modelo anterior, ponderando mais

os dados que foram classificados erroneamente pelos modelos anteriores. Em outras palavras, os modelos-base são treinados em sequência em uma versão ponderada dos dados (Figura 2.9);

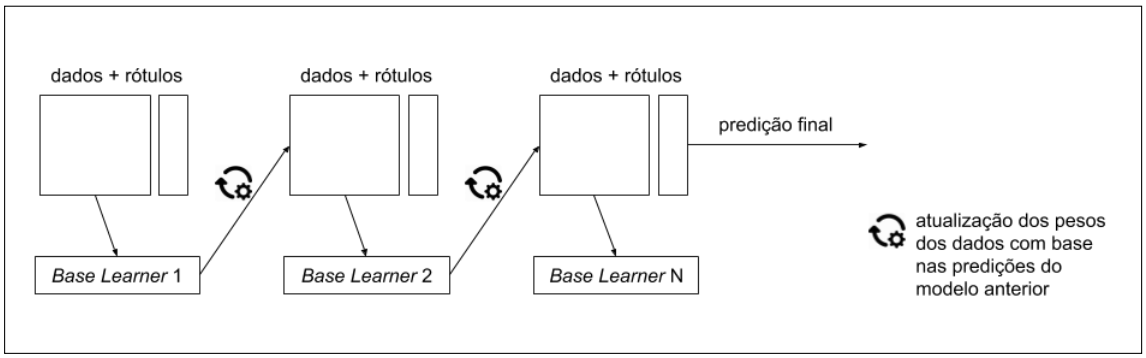

Figura 2.9: Exemplo de *ensemble* com abordagem *boosting*.

- *Stacking*: representa uma sequência de modelos-base, onde cada modelo é treinado usando as saídas do modelo anterior como variáveis de entrada, sendo o último a calcular a predição final chamado de *meta learner*. Nesse caso, diferentes tipos de modelos-base geralmente são utilizados (Figura 2.10).

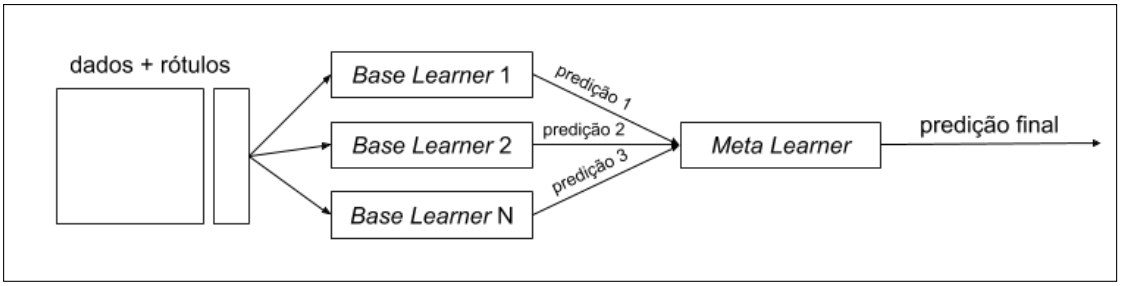

Figura 2.10: Exemplo de *ensemble* com abordagem *stacking*.

2.2.3 Exemplos de Modelos

Neste trabalho, são utilizados *ensembles* baseados em modelos de árvore de decisão, que são basicamente um conjunto de regras do tipo *if* e *else* para cada variável. Cada nó na árvore representa uma variável e suas ramificações representam os valores possíveis. Os nós das folhas das árvores representam o valor da predição. O modelo de árvore de decisão é especialmente atrativo pela rápida convergência (tempo de treinamento) e pela simplicidade na

interpretação da arquitetura e dos resultados. Entretanto, esses modelos apresentam limitação ao lidar com problemas complexos. Desde suas aplicações iniciais ([45], [8]), muitas adaptações surgiram dos modelos de árvore de decisão:

- *Random Forest* (RF): define um conjunto de árvores de decisão em que cada modelo-base (árvore) é treinado com um subconjunto selecionado aleatoriamente do conjunto das variáveis e dos dados de treinamento. O objetivo é reduzir a variância do modelo, isto é, o *overfitting*. Com isso, há um aumento no viés (complexidade do modelo) e alguma perda de interpretabilidade dos resultados, porém aumentando a eficácia final do modelo em relação à uma árvore de decisão individual. Árvores treinadas em diferentes subconjuntos de dados generalizam sua classificação de maneiras complementares, e sua classificação combinada pode ser monotonicamente aprimorada [24]. A saída final do RF é a saída mais recorrente entre os modelos-base (no caso da classificação) ou a média das predições de todas os modelos-base (no caso da regressão), como exibido na Figura 2.11;

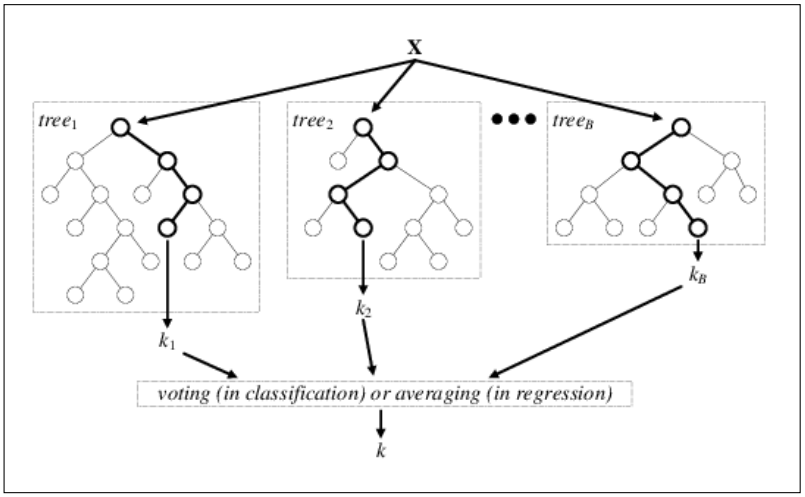

Figura 2.11: Exemplo de modelo *Random Forest*. Fonte: *https://github.com/hvantil/RandomForestTutorial/blob/master/RandomForestTutorial.ipynb*

- *Extreme Gradient Boosting* (XGBoost): é uma adaptação do algoritmo *Gradient Boosting* [21], desenvolvido para ser escalonável e rápido, lidando com dados esparsos [12]. *Gradient Boosting* constrói modelos aditivos - um conjunto de árvores, ajustando sequencialmente um *weak learner* aos atuais "pseudo"-residuais da árvore anterior por

meio dos mínimos quadrados a cada iteração (Figura 2.12). Os pseudo-residuais são o gradiente da função de perda que está sendo minimizada, com relação aos valores da árvore a cada dado de treinamento avaliado no passo atual. Especificamente, a cada iteração, uma subamostra dos dados de treinamento é coletada aleatoriamente (sem reposição) do conjunto completo de dados de treinamento para ajustar a árvore e calcular a atualização do modelo para a iteração atual [21].

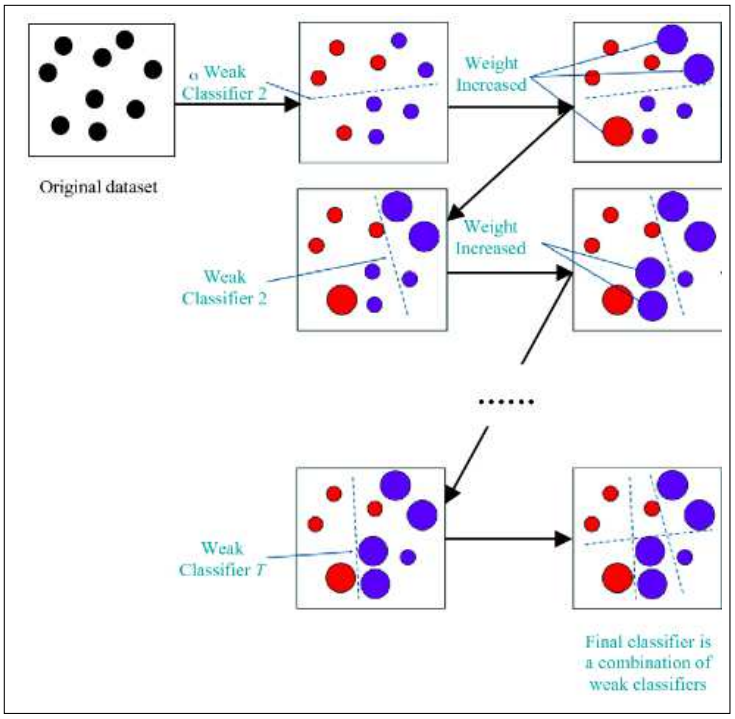

Figura 2.12: Funcionamento de modelo *Gradient Boosting*. Fonte: Adaptação de *https://datascience.eu/pt/aprendizado-de-maquina/gradient-boosting-o-que-voce-precisa-de-saber/*

- *Categorical Boosting* (CatBoost): é outra adaptação do algoritmo *Gradient Boosting*, com suporte a variáveis categóricas, lidando com elas durante o treinamento e reduzindo o tempo de pré-processamento para esta tarefa de conversão das variáveis. Outra característica do algoritmo é a utilização de um novo esquema para calcular os valores das folhas ao selecionar a estrutura da árvore, o que ajuda a reduzir o *overfitting* (modelo excessivamente ajustado) [16]. As mesmas variáveis são usadas para fazer divisões à esquerda e à direita para cada nível da árvore, impulsionando o crescimento de uma árvore balanceada e simétrica, como o exemplo da Figura 2.13.

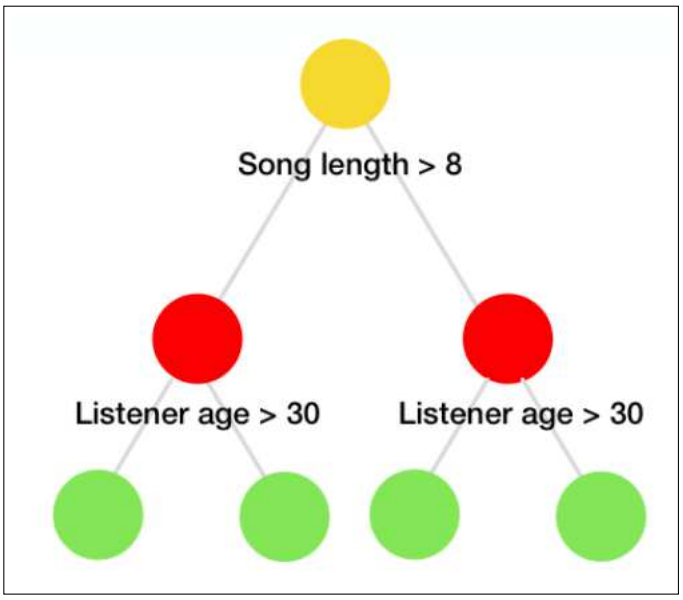

Figura 2.13: Exemplo de árvore de decisão simétrica. Fonte: *https://catboost.ai/news/catboost-enables-fast-gradient-boosting-on-decision-trees-using-gpus*

2.2.4 Predição de Múltiplos Passos

A predição de múltiplos passos a frente, ou o horizonte de predição, define quanto tempo antes da ocorrência de um evento, é possível prevê-lo [1; 13]. Normalmente, os modelos de AM apresentam a predição de um resultado da observação no próximo passo (tempo, por exemplo). Predizer o valor de venda de uma casa, o risco de um cliente tornar-se inadimplente e a ocorrência de acidentes na via são exemplos de problemas de predição de um passo.

A predição múltiplos passos a frente refere-se a necessidade de obter não somente o valor de uma predição, mas múltiplas predições para os passos seguintes. Por exemplo, predizer o valor de venda de uma casa para os próximos três meses, o risco de um cliente torna-se inadimplente nos próximos dois anos e a ocorrência de acidentes nas próximas ruas apresentam como horizonte de predição os três meses, dois anos e as próximas ruas, respectivamente.

Para realizar a predição de múltiplos passos, é necessário atualizar o modelo e/ou os dados de treinamento para considerar o horizonte de predição desejado. Para isto, diversas técnicas podem ser aplicadas [9], como: utilizar um modelo individual para cada passo dese-

jado; aplicar um modelo recursivo que considera como entrada as predições dos passos anteriores; utilizar modelos para séries temporais, como o LSTM (*Long Short-Term Memory*); ou aplicar um modelo que produz múltiplas saídas simultaneamente.

2.2.5 Aprendizagem Incremental

Os modelos de AM são utilizados para obter informações relevantes dos dados coletados e/ou prever algum evento. No entanto, os modelos clássicos de aprendizagem de máquina treinados em lote, nos quais todos os dados são acessados simultaneamente, não integram continuamente novas informações a modelos já construídos, mas reconstroem regularmente novos modelos a partir do zero. Isso não apenas consome muito tempo, mas também leva a modelos potencialmente desatualizados [33].

Ao lidar com problemas dinâmicos no mundo real, o modelo gerado se torna obsoleto rapidamente. Neste contexto, técnicas de aprendizagem incremental podem ser aplicadas para incluir continuamente novos dados no modelo proposto que, como o nome sugere, atualiza o modelo de acordo com os novos dados recebidos.

Conforme declarado em [33], um algoritmo de aprendizagem incremental gera, em um determinado fluxo de dados de treinamento $s_1, s_2, ..., s_t$, uma sequência de modelos $h_1, h_2, ..., h_t$. Nesse caso, s_i é rotulado como dado de treinamento $s_i = (x_i, y_i) \in \mathbb{R}^n \times \{0, 1\}$ e $h_i : X \rightarrow Y \mid |X| \leq p, i > 1$ é uma função dependendo exclusivamente de h_{i-1} e das p recentes instâncias de dados de treinamento $s_{i-(p+1)}, ..., s_i$. Esta técnica de aprendizagem incremental foi aplicada ao modelo proposto e será destacada no Capítulo 4.

2.3 Considerações Finais

Neste capítulo, foram apresentados os principais conceitos relacionados às fontes de dados de transporte público e às técnicas de AM. Foram descritos conceitos das fontes de GPS, GTFS, clima e situação do trânsito, assim como exibidos exemplos dos dados. Em relação à AM, foram apresentados a definição de modelo, aprendizado supervisionado, técnicas de ensemble, exemplos de modelos de árvore de decisão, predição múltiplas paradas e aprendizagem incremental.

No capítulo seguinte, serão apresentados os trabalhos relacionados à predição de aglomerados de ônibus, com destaque para suas vantagens e limitações.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, é apresentada a metodologia empregada para pesquisar os trabalhos que abordam o problema de aglomerado de ônibus (Seção 3.1). Em seguida, na Seção 3.2, são discutidos e comparados os trabalhos relacionados à predição destes aglomerados, além de exibir suas principais características. Por fim, na Seção 3.3, são discutidas as considerações finais deste capítulo

3.1 Metodologia

Os trabalhos selecionados nesta pesquisa foram encontrados a partir da realização de consultas ad-hoc (isto é, variando as chaves de busca) aos sites de pesquisa IEEE Xplore Digital Library[1], ACM Digital Library[2] e Google Scholar[3], consistindo de trabalhos publicados desde 2009. As principais palavras-chave consideradas foram "aglomerado de ônibus", "predição", "ônibus" , "trânsito" e "tráfego", em português, e "bus bunching", "prediction", "bus", "traffic", "transit", em inglês. As palavras-chave foram verificadas tanto no título dos trabalhos como no conteúdo dos trabalhos.

Os artigos que continham as palavras chaves foram selecionados para leitura, enquanto os demais foram descartados. Além disso, com base nas referências apresentadas nas publicações encontradas, foram selecionados outros trabalhos adicionais para compor a pesquisa (técnica conhecida como *snowball sampling* ou amostragem de bola de neve [43;

[1] *https://ieeexplore.ieee.org/Xplore/home.jsp*

[2] *https://dl.acm.org/*

[3] *https://scholar.google.com/*

18]). Aproximadamente 25 trabalhos específicos sobre aglomerados de ônibus foram estudados, considerando os relacionados à análise, detecção e predição deste tipo de evento. Após análise exploratória destes, onze trabalhos sobre predição do aglomerados de ônibus foram filtrados e apresentados a seguir.

3.2 Predição de Aglomerados de Ônibus

Desde meados do século passado, o problema dos aglomerados de ônibus tem sido estudado por pesquisadores e especialistas em transporte, quando os autores de [40], usando um modelo matemático simplificado, provaram a instabilidade de uma rota de ônibus, ou seja, a tendência dos ônibus aglomerarem. Desde então, surgiram vários trabalhos relacionados a esse problema, alguns dos quais se concentram na determinação das características espaço-temporais e causas dos aglomerados, enquanto outros focam na predição destes eventos.

Os autores em [46] analisaram dados de GPS e concluíram que o desvio do horário programado é o fator mais influente para a ocorrência de aglomerados de ônibus. Por sua vez, os autores em [4], usando dados de GPS e AFC, identificaram que fatores como rotas de alta frequência (ou seja, rotas atendidas por muitos ônibus), alta demanda de passageiros, alta variabilidade de demanda e pontos de ônibus colocados no final do rota contribuem para aumentar a ocorrência dos aglomerados. Esses são alguns exemplos de trabalhos relacionados à análise dos eventos de aglomerados de ônibus.

Outros trabalhos, por sua vez, se concentram na predição dos aglomerados de ônibus, foco desta pesquisa. Em [38], um modelo de predição de aglomerados de ônibus é apresentado, com base nas predições do horário de chegada nas paradas, usando dados de GTFS, além de dados em tempo real e históricos do GPS. Os experimentos foram conduzidos nos dados resultantes da aplicação do modelo em todos os ônibus de quatro rotas da cidade de Miami, durante oito dias. A eficácia, baseada no RMSE e na acurácia, foi avaliada com horizonte de predição entre 10 e 60 minutos. A desvantagem deste modelo é que o resultado da precisão depende da alta frequência de envio do GPS, ou seja, quando o intervalo de envio de um dado de geolocalização para outro é de até 20 segundos. Na prática, nem todas as empresas de transporte possuem dispositivos GPS modernos de alta frequência.

Como mostrado em [37], os autores apresentaram um modelo de predição de aglome-

rados de ônibus em tempo real, utilizando a predição do *headway*. Em [36], os autores complementaram este modelo com a sugestão de ações corretivas a serem tomadas para mitigar o problema dos aglomerados após a predição já realizada. Para atualizar constantemente o modelo, os autores aplicam aprendizagem incremental reutilizando os resíduos de cada predição (diferença entre o valor predito e o rótulo) para melhorar as predições posteriores. Assim que cada *headway* real se torna disponível, as predições das paradas seguintes são atualizadas considerando uma porcentagem dos últimos resíduos. Usando dados de GPS e GTFS aplicados aos modelos de Regressão Linear (RLi) e Rede Neural Artificial (RNA), os resultados atingiram uma precisão de 54%. Isso significa que muitos dos eventos classificados como aglomerados de ônibus foram, na verdade, resultados falso-positivos. O desempenho do modelo foi avaliado usando dados coletados de 18 rotas de ônibus da cidade de Porto, Portugal, durante o período de um ano.

No trabalho de [2], resultado da dissertação de [1], os autores propõem um modelo preditivo de aglomerados de ônibus, usando predição de *headway* em tempo real, e sugerem ações de controle para evitar desvio excessivo do *headway* programado. Os autores avaliaram quatro modelos de AM com dados de GPS e GTFS: RNA, RLi, Regressão Kernel (RK) e modelo Autoregressivo (AR). O desempenho foi avaliado usando dados coletados no período de um mês de uma rota de ônibus da cidade de Dublin, e também de uma rota artificial. A eficácia foi avaliada no horizonte de predição de até 10 minutos. Os modelos apresentaram desempenho semelhante: valor de RMSE (*Root Mean Square Error*) entre um e dois minutos. Esse resultado é alcançado usando algumas premissas distantes do que é observado na prática, como a capacidade ilimitada de passageiros no ônibus e o fato de um ônibus da mesma rota nunca ultrapassar o outro.

Modelos de regressão, chamados LS-SVM (*Least-squares Support Vector Machine*) e RVM (*Relevance Vector Machine*), foram propostos pelos autores em [58] e [59], respectivamente, para realizar predições de *headway* usando dados de AFC. O desempenho dos modelos foi avaliado em um conjunto de paradas de ônibus de duas rotas da cidade de Pequim, sendo estes dados coletados no período de quatro meses. A eficácia foi avaliada no horizonte de predição de até 5 paradas. Os dados que foram empregados são mais esparsos que os dados de geolocalização. Como os sistemas AFC coletam dados de embarque e, algumas vezes, desembarque de passageiros em cada veículo [7;

59], esses dados só existem nos pontos de ônibus onde os passageiros embarcam/desembarcam, embora saiba-se que nem sempre há passageiros embarcando/desembarcando em todas as paradas de ônibus. Além disso, os dados de AFC geralmente são privados e dificilmente disponibilizados pelas empresas, pois permitem obter informações sobre as receitas da empresas de ônibus e as rotas de passageiros [60; 53].

Em 2018, os autores em [11] patentearam um método que possui duas partes: a primeira faz diversas predições sobre as rotas (onde o ônibus estará, possivelmente), onde o ônibus é provável de aglomerar, a demanda de passageiros e o tempo das trajetórias; a segunda parte recomenda diversas ações preventivas para impedir os aglomerados de ônibus. A desvantagem do trabalho é que não há qualquer descrição do componente de predição e da sua eficácia.

Em [57], os autores também utilizam dados AFC e variações do modelo SVM para predizer os aglomerados, a partir da predição do *headway* nas paradas de ônibus. Em resumo, o processo preditivo é dividido em três etapas: primeiro, os fatores que influenciam na irregularidade do *headway* são definidos com base nos dados históricos do AFC; segundo, modelos SVM e LS-SVM são usados para predizer os *headways* para a próxima parada de ônibus; e, por último, a correlação entre *headway* e os aglomerados é definida para detectar a ocorrência dos mesmos. O desempenho dos modelos foi avaliado em um conjunto de paradas de ônibus de uma rota da cidade de Changzhou, coletados no período de dois meses. Como destacado pelos autores, a desvantagem dos modelos aplicados é o custo elevado de processamento dos dados, sendo inviável para aplicação em grandes quantidades de dados de treinamento.

Por fim, no trabalho de [52], resultado da tese de [51], os autores exploram uma metodologia preditiva probabilística para prever se um ônibus irá ou não aglomerar em uma parada a seguir. Eles usam um modelo de Regressão Logística (RLo) baseado em registros de GPS de ônibus com pelo menos k paradas à frente e analisam o *trade-off* entre "sensibilidade" e "especificidade". Os resultados indicaram uma sensibilidade de até 80%, que é a proporção de eventos de aglomerados de ônibus corretamente identificados, ao custo da perda da especificidade, que é a proporção de predições corretas de não aglomerados em relação a todos os eventos. Da mesma forma que em [36], a consequência de resultados falso-positivos é a perda da confiabilidade dos resultados do modelo e o tempo investido dos agentes de

trânsito ao executar ações preventivas desnecessárias a partir destas predições. O desempenho do modelo foi avaliado usando duas semanas de dados coletados de uma rota de ônibus da cidade de Kyoto, cuja técnica de balanceamento *undersampling* foi aplicada para evitar enviesamento do modelo. A eficácia foi avaliada no horizonte de predição de até 15 paradas.

Na Tabela 3.1, é apresentado um resumo dos trabalhos relacionados descritos anteriormente. Cada autor usa diferentes modelos de AM para predizer aglomerados de ônibus e a maioria deles usa as mesmas fontes de dados para a entrada do modelo: GPS e GTFS. Somente os autores [36] aplicam a técnica de aprendizagem incremental para atualizar continuamente o modelo. Além disso, a aplicação de diferentes medidas de eficácia dificulta a comparação de qualidade entre esses modelos.

3.3 Considerações Finais

Neste capítulo, foram apresentados os trabalhos relacionados à predição de aglomerados de ônibus. Em geral, os trabalhos utilizam modelos de AM e dados de AFC ou GPS e GTFS para realizar as predições. Além de realizarem os experimentos com dados de diferentes cidades, os autores também aplicam métricas distintas que dificultam a comparação de eficácia. Além disso, somente um trabalho aplica a técnica de aprendizagem incremental para atualizar o modelo de predição.

No capítulo seguinte, será apresentada a solução proposta de predição de aglomerados de ônibus, cujas principais diferenças são o modelo utilizado e as fontes de dados empregadas.

Tabela 3.1: Resumo dos trabalhos relacionados à predição de aglomerados de ônibus

Artigo	Modelo Preditivo	Fonte(s) de Dados	Aprend. Incremental	Eficácia	Cidades Analisadas
[38]	Não informado	GPS, GTFS	Não	RMSE = 2-8 min Acurácia = 68-80%	Miami-Dade (EUA)
[36]	RLi, RNA	GPS, GTFS	Sim	Acurácia = 98% Precisão = 54% Cobertura = 75%	Porto (Portugal)
[2]	RLi, RK, RNA, AR	GPS, GTFS	Não	RMSE = 1-2 min	Dublin (Irlanda)
[58]	LS-SVM	AFC	Não	MAPE = 1-16% RMSE = 1-6%	Pequim (China)
[59]	RVM	AFC	Não	MAPE = 24% RMSE = 3%	Pequim (China)
[11]	Não informado	GPS, GTFS, AFC	Não	Não informado	Não informado
[57]	SVM	AFC	Não	Acurácia = 97-98% Precisão = 88-100% Especif. = 98-99%	Changzhou (China)
[52]	RLo	GPS	Não	Acurácia = 91-96% Especif. = 90-99% Sensib. = 34-80%	Kyoto (Japão)

Capítulo 4

Solução Proposta

Neste capítulo, será apresentada a solução proposta para predição de aglomerados de ônibus em tempo real. Na Seção 4.1, é exibida a formalização do problema de predição de aglomerados de ônibus. Na Seção 4.2, são apresentados alguns padrões da ocorrência dos aglomerados de ônibus, identificados nos dados analisados. Na Seção 4.3, é apresentada a solução proposta com ênfase nas etapas do fluxo de execução e nos passos de cada uma. Em seguida, na Seção 4.4, é apresentado como a aprendizagem incremental pode ser incorporada ao modelo proposto para atualizá-lo continuamente. Na Seção 4.5, é descrita a predição para múltiplas paradas de ônibus consecutivas de uma rota. Por fim, na Seção 4.6 são discutidas as considerações finais deste capítulo.

4.1 Formalização do Problema de Predição de Aglomerados de Ônibus

Esta seção apresenta a formalização do problema de predição de aglomerados de ônibus. Neste sentido, sejam $B_r = \{b_1, b_2, ..., b_m\}$ um conjunto de ônibus executando a rota r e $T_{b_i} = \{t_{i,1}, t_{i,2}, ..., t_{i,s}\}$ um conjunto de horários, tal que $t_{i,k}$ denota o horário de chegada do ônibus i na parada k e s denota o número total de paradas de ônibus.

Considerando a parada de ônibus k, o *headway* atual h_k^a entre dois ônibus b_i e b_{i+1}, executando a mesma rota, pode ser definido como: $h_k^a = |t_{i+1,k}^a - t_{i,k}^a|$ (em minutos). Com isto, a ocorrência de aglomerado de ônibus envolvendo b_i e b_{i+1} é definida como:

$$BB_k = \begin{cases} 1, & \text{se } h_k^a < \alpha \\ 0, & \text{c.c.} \end{cases} \tag{4.1}$$

onde α é um limiar geralmente baseado nos dados programados pelas empresas de ônibus. Neste trabalho, considera-se $\alpha = h_k^p/4$, onde $h_k^p = |t_{i+1,k}^p - t_{i,k}^p|$ é o *headway* programado, como proposto em [37]. Dessa forma, o desafio é propor um modelo para predizer a ocorrência dos aglomerados entre cada par de ônibus da mesma rota, estabelecendo um relacionamento de entrada-saída:

$$\hat{BB_k} = f(X) \tag{4.2}$$

onde X representa o conjunto de variáveis de entrada.

Além disso, é necessário predizer os aglomerados com elevado horizonte de predição (HP), ou seja, n paradas de ônibus antes da ocorrência do aglomerado na parada k. Neste sentido, o objetivo é predizer este evento na parada $k - n$, com n sendo um limiar definido pelo usuário (por exemplo, um agente de transporte público ou de trânsito).

Assim, considerando o modelo de predição de aglomerados de ônibus, os objetivos são:

$$\textbf{minimizar } TP \tag{4.3}$$

$$\textbf{maximizar } HP \tag{4.4}$$

$$\textbf{maximizar } F_{measure} \tag{4.5}$$

onde TP é o tempo de predição dos aglomerados e $F_{measure}$ é uma medida de eficácia para avaliar se o modelo está (ou não) predizendo corretamente. Em geral, quanto maior o HP, menor serão os resultados de eficácia do modelo, porque é desafiador predizer eventos estocásticos no trânsito com antecedência [36] [59].

4.2 Padrões Identificados da Ocorrência dos Aglomerados de Ônibus

Ao analisar a ocorrência dos aglomerados de ônibus nos dados referentes às duas cidades consideradas nesta pesquisa, alguns padrões foram identificados através das visualizações a seguir, como:

- Tendência dos ônibus aglomerarem em horário de pico: 8h e 19h;
- Alta ocorrência de aglomerados no final do dia, às 23h;
- Baixa ocorrência dos aglomerados nas primeiras horas do dia;
- Ocorrência maior dos aglomerados no início da semana;
- Diminuição da ocorrência dos aglomerados nos finais de semana.

Nas Figuras 4.1 e 4.2, são exibidas as proporções da ocorrência de aglomerados de ônibus nas viagens, por hora, nas cidades de Curitiba e Cidade A, respectivamente. O eixo horizontal representa as horas do dia e o eixo vertical representa as proporções de aglomerados de ônibus. Nestas duas imagens, observa-se que os pontos máximos das curvas ocorrem em torno das 8h e 19h, conhecidos como horários de pico no Brasil: início e fim aproximado das jornadas diárias de estudo e trabalho. Além disso, há diminuição na ocorrência dos aglomerados nas primeiras horas do dia, pois não há demanda de passageiros e as frotas de ônibus ainda estão começando a circular. Por fim, é possível observar um aumento na ocorrência dos aglomerados no final do dia, às 23h, possivelmente quando os ônibus começam a deslocar-se para a garagem ao finalizar o expediente.

As proporções da ocorrência de aglomerados de ônibus nas viagens, por dia da semana, nas cidades de Curitiba e Cidade A são apresentadas nas Figuras 4.3 e 4.4, respectivamente. O eixo horizontal representa os dias analisados (agrupados por dia da semana) e o eixo vertical representa as proporções de aglomerados de ônibus. Em ambas as figuras, os pontos que representam a maior ocorrência dos aglomerados indicam os dias relacionados primordialmente às segundas e quintas na Cidade A e terças em Curitiba. Os dias com menor frequência de aglomerados costumam ser nos finais de semana, principalmente aos domingos e sábados, quando há diminuição na demanda por transporte público.

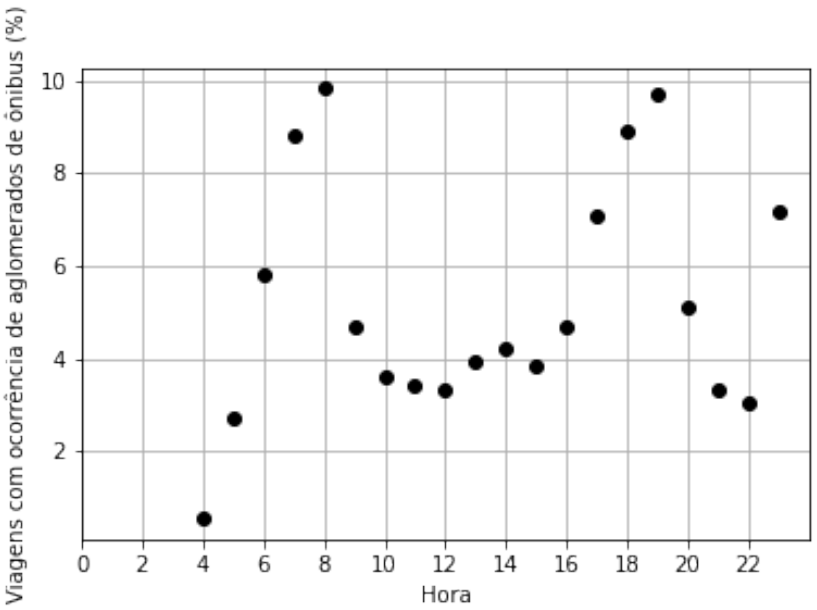

Figura 4.1: Proporção da ocorrência de aglomerados de ônibus por hora. Dados de maio de 2019, Curitiba.

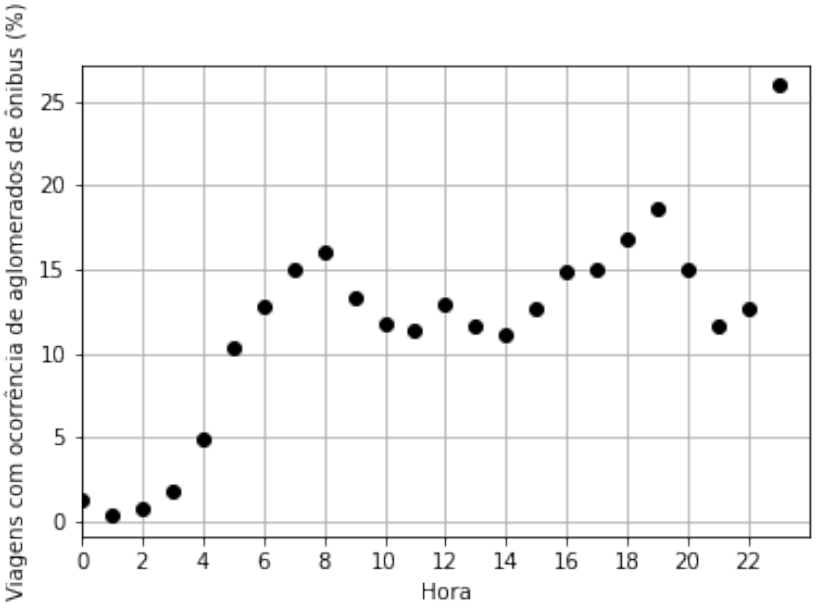

Figura 4.2: Proporção da ocorrência de aglomerados de ônibus por hora. Dados de dezembro de 2018, Cidade A.

Estes resultados indicam padrões para os quais a atenção dos agentes de trânsito pode se voltar para estudo e planejamento, visando mitigar a ocorrência dos aglomerados de ônibus.

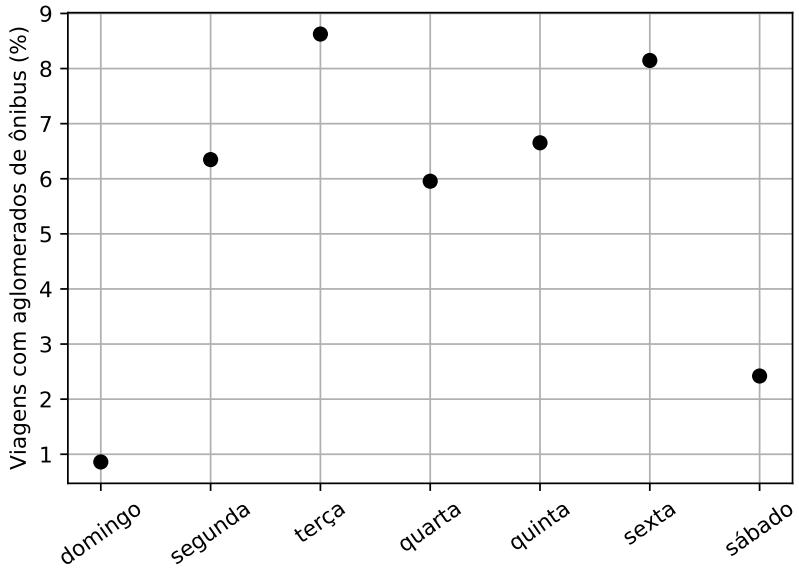

Figura 4.3: Proporção da ocorrência de aglomerados de ônibus por dia da semana. Dados de maio de 2019, Curitiba.

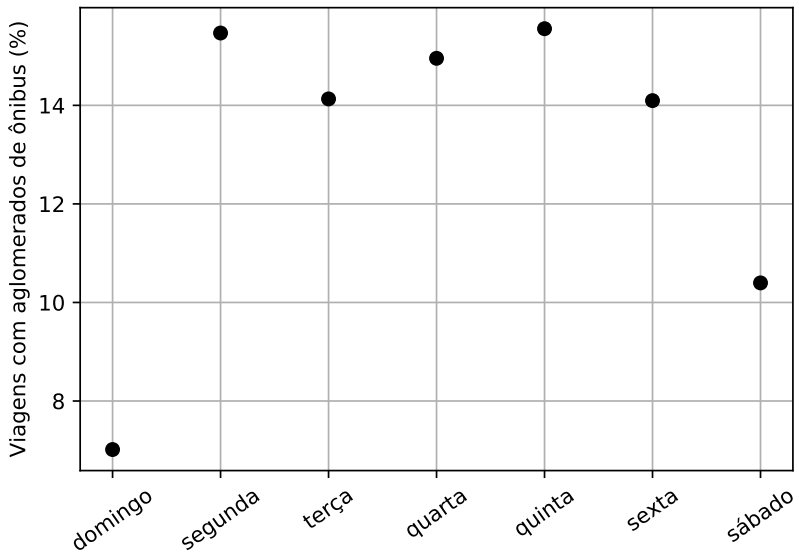

Figura 4.4: Proporção da ocorrência de aglomerados de ônibus por dia da semana. Dados de dezembro de 2018, Cidade A.

4.3 Modelo para Predição de Aglomerados de Ônibus

Considerando que os eventos no trânsito são estocásticos, ou seja, acontecem de forma aleatória, predizer estes eventos, como o aglomerado de ônibus, com antecedência é um desafio.

Para atenuar a ocorrência deste evento no trânsito, é necessário definir um modelo de predição robusto que identifique a ocorrência dos aglomerados de forma eficaz e eficiente, ou seja, com precisão e com antecedência.

Neste trabalho, foram aplicados modelos de aprendizagem de máquina para recuperar padrões dos dados históricos (GPS, GTFS, clima e situação da via) que indicam a ocorrência dos aglomerados. Estes modelos, em oposição aos modelos estatísticos, buscam descrever as propriedades dos dados sem o conhecimento prévio da distribuição dos mesmos. Por não dependerem explicitamente de parâmetros para modelar o comportamento do evento, esses modelos são mais simples de serem ajustados e demonstram considerável desempenho mesmo quando aplicados a relacionamentos complexos e não lineares [44].

Neste sentido, foi considerada a técnica de *ensemble*, que combina diversos modelos de AM com o mesmo propósito. O *ensemble* pode obter maior precisão por meio de diferentes estratégias de combinação. Além disso, modelos com esta técnica apresentam uma capacidade de generalização mais forte que os modelos individuais de AM [42]. Sabendo que os modelos-base são independentes, o erro de predição do modelo diminui quando a abordagem de *ensemble* é usada, por meio da utilização da "sabedoria das multidões"(*wisdom of crowds*) para realizar uma predição [31; 32]. Mesmo que o modelo *ensemble* utilize vários modelos-base internamente, ele atua e funciona como um único modelo. Portanto, uma das motivações da utilização desta técnica é a diminuição do erro de generalização, visando melhorar a estabilidade e eficácia das predições.

Para aplicar a técnica de *ensemble*, três modelos baseados em árvores de decisão são considerados neste trabalho. Tais modelos geralmente apresentam resultados satisfatórios devido ao fato de que os algoritmos de aprendizado baseados em árvore são capazes de se adaptarem a relacionamentos complexos e, ao mesmo tempo, são eficazes em termos de custos computacionais [29]. Como indicado por [59], o horário de chegada do ônibus exibe um comportamento não linear e irregular, especialmente durante os horários de pico; assim, o aglomerado de ônibus também apresenta essa relação não linear e complexa entre as variáveis.

Comparados a outros modelos, como RNA, Regressão e *K-Nearest Neighbours* (KNN), os modelos baseados em árvores de decisão apresentaram melhor desempenho, com atenção especial para RF, XGBoost e CatBoost. A justificativa é que esses modelos também usam

uma técnica de *ensemble*, combinando várias árvores com diferentes abordagens. Essas combinações geralmente evitam o ajuste excessivo e generalizam o modelo a partir dos dados. Em vez de aplicar modelos simples, estes modelos mais robustos - RF, XGBoost e CatBoost - foram considerados como modelos-base para o *ensemble*, assumindo que uma combinação das decisões de um grupo de modelos diferentes é capaz de superar um único modelo [56].

Considerando os argumentos acima, este trabalho apresenta um modelo *ensemble* para predição de aglomerados de ônibus, que considera as predições de diferentes modelos-base, são executados em paralelo, são treinados nos mesmos dados e combinados com a abordagem de votação (Figura 4.5). O objetivo é criar um modelo eficaz e genérico que possa ser aplicado em todas as frotas de ônibus de uma cidade em tempo real. Como mostrado na Figura 4.5, o modelo emprega quatro fontes de dados distintas: GPS, GTFS, situação climática e situação do trânsito. Após a integração dos dados, cada modelo-base construído realiza a predição dos aglomerados de ônibus em paralelo. Em seguida, é aplicada a técnica de votação sobre as saídas destes modelos e a predição final é o valor que ocorreu com maior frequência entre eles. Esta abordagem pode ser aplicada em tempo real, considerando que este processo de predição consome menos de um segundo para cada instância de entrada.

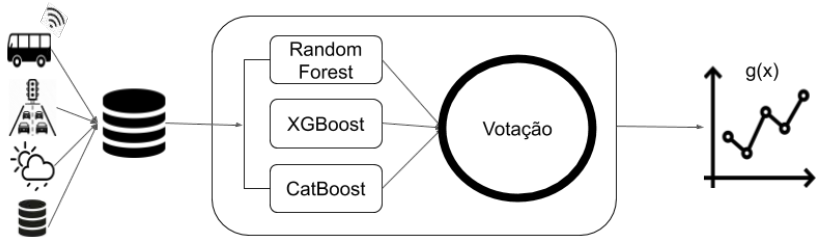

Figura 4.5: Modelo *ensemble* baseado na técnica de votação para predizer aglomerados de ônibus.

Além da técnica de votação, outras técnicas de *ensemble*, como *boosting* e *stacking*, foram avaliadas, mas não superaram os modelos-base. Em alguns casos, o *ensemble* com a técnica *boosting* falhou, indicando que o modelo combinado especializou-se nos dados de treinamento (*overfitted*) [56]. Por outro lado, se o primeiro modelo de base do *stacking* não apresentar uma boa eficácia, os erros serão propagados para os outros modelos na sequência e o resultado final não será superior ao modelo único.

Para melhor entender o funcionamento deste modelo proposto, é apresentado na Figura

4.6 o fluxo de execução do modelo, cujas etapas estão detalhadas nas seções a seguir: pré-processamento de dados, treinamento do modelo e teste do modelo.

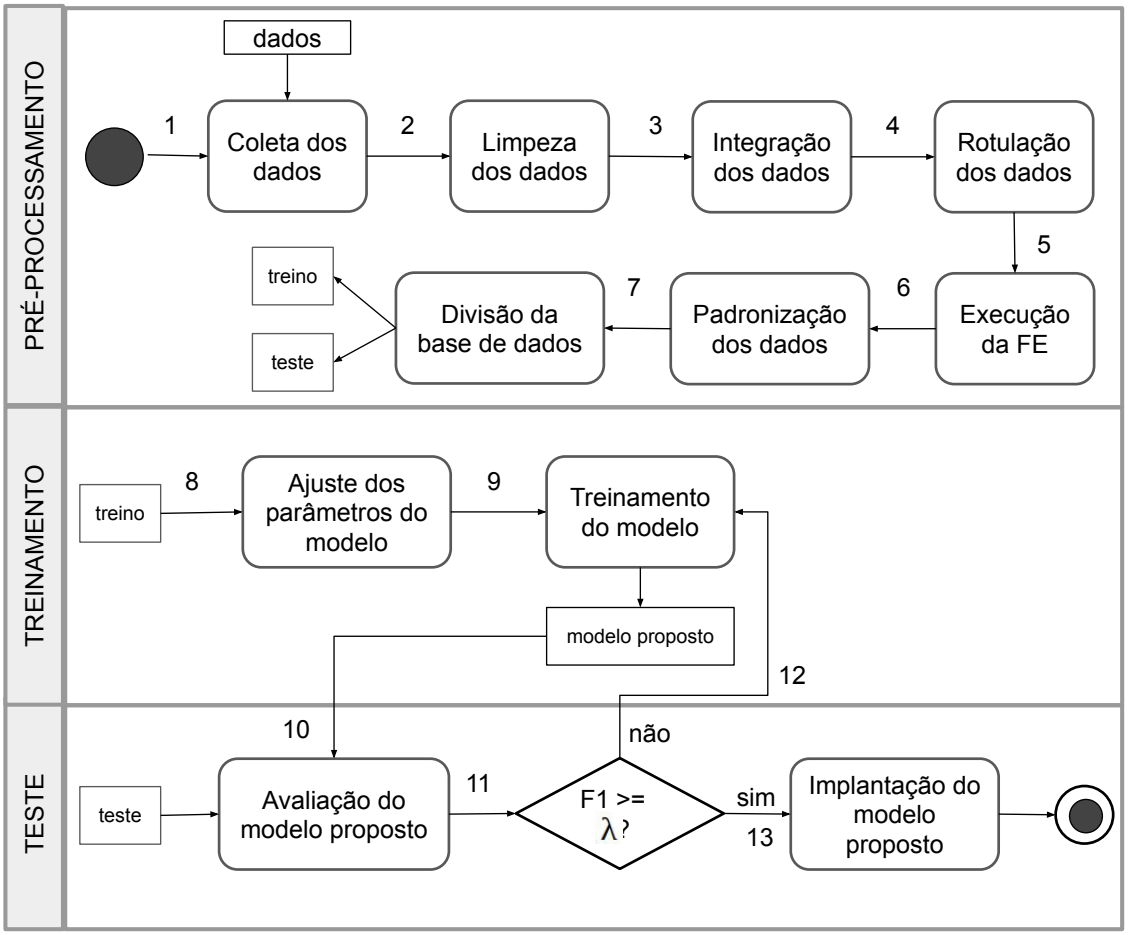

Figura 4.6: Diagrama de atividades do modelo proposto para predição de aglomerados de ônibus.

4.3.1 Etapa de Pré-processamento

Na primeira etapa, ocorre o pré-processamento dos dados, que consiste na coleta, limpeza, integração, rotulação dos dados, execução da *feature engineering* (FE), padronização dos dados e divisão da base de dados em treino e teste. Estes sete passos da etapa de pré-processamento são descritos a seguir.

Coleta de Dados

Primeiro, os dados são coletados e processados. Fontes de dados acessíveis, ou seja, facilmente disponibilizadas pelas empresas de transporte ou órgãos de trânsito, são usadas para facilitar a adaptabilidade do modelo proposto para diversas cidades. Os dados de GPS representam as localizações do ônibus em tempo real, enquanto os dados do GTFS fornecem horários programados, o que ajuda a predizer, por exemplo, desvios de trajetória e atrasos no horário de chegada nas paradas. Como os dados climáticos influenciam diretamente na situação do tráfego, são utilizadas informações de precipitação para determinar o impacto das condições climáticas. Por fim, a situação do tráfego também é considerada para melhorar a predição dos aglomerados de ônibus com base nos eventos que ocorrem no trânsito.

Limpeza dos Dados

Em seguida, cada conjunto de dados coletado é analisado a fim de remover as instâncias que apresentam os seguintes valores discrepantes ou atípicos: situação de tráfego referente a cidades diferentes das analisadas e valores ausentes em variáveis relevantes do GPS, como rota, latitude e longitude. A remoção dos dados de cidades diferentes foi realizada porque otimiza o tempo de processamento das operações com estes dados. Enquanto isso, variáveis de GPS, como a rota, são essenciais para identificar o trajeto predefinido dos ônibus. Além disso, a geolocalização do GPS (latitude e longitude) é necessária para identificar o caminho percorrido pelo veículo no horário informado, por isso, instâncias sem os valores destas variáveis foram removidas.

Integração dos Dados

Os quatro conjuntos de dados são então integrados para formar uma única base de dados usando a rota e as coordenadas geográficas como chave para integração. Para isso, foram utilizadas medidas de distância entre os pontos de geolocalização das instâncias de cada fonte de dados. Por exemplo, cada parada de ônibus é associada ao ponto de *shape* da mesma rota mais próximo (em termos de distância). Este processo está detalhado no Capítulo 5.

Rotulação dos Dados

Após a integração dos dados, cada instância do conjunto de dados (linha) corresponde a um ônibus e o seu consecutivo da mesma rota na parada k. Para rotular cada instância, ou seja,

classificar se os dois ônibus estão (ou não) aglomerados, é utilizada a Equação 4.1. Com os dados rotulados, será possível aplicar o aprendizado supervisionado no treinamento dos modelos utilizados.

Execução da *Feature Engineering*

Neste passo, duas técnicas de FE são aplicadas para aprimorar a base de dados integrada: imputação numérica e extração de variável. Quando uma variável numérica está vazia, é realizada a imputação, ou seja, o campo da instância com valor ausente é substituído pela mediana dos valores da variável. Para cada variável do tipo data, os valores de dia, dia da semana, mês e ano são extraídos como novas variáveis. A seleção automática e manual de variáveis (*feature selection*) não funcionou nesse cenário, pois cada variável apresenta baixa correlação individual com a ocorrência de aglomerados de ônibus.

Na Figura 4.7 é mostrada a correlação de um subconjunto de variáveis em ordem descendente, em que o eixo vertical representa as variáveis de entrada do modelo (variáveis independentes), o eixo horizontal representa as variáveis de ocorrência de aglomerados de ônibus e a luminosidade da cor representa os valores de correlação (valores próximos a 1 ou -1 indicam correlação forte, enquanto valores próximos a 0 indicam correlação fraca) entre cada par de variáveis com o teste de Kendall [28]. Este teste não mostrou correlação significativa entre as variáveis e a ocorrência dos aglomerados, indicando que uma única variável por si só não explica a ocorrência desse evento.

Padronização dos Dados

Após a engenharia de variáveis, o intervalo dos dados de cada variável é alterado (*feature scaling*), ou seja, o intervalo dos dados é padronizado para evitar o enviesamento dos modelos e lidar com variáveis em diferentes escalas. Para resolver isso, todos os valores de cada variável são redimensionados para um intervalo entre 0 e 1, utilizando a Equação 4.6 para cada valor das variáveis:

$$x_{novo} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{4.6}$$

onde x_i é o valor i da variável atual x, enquanto x_{max} e x_{min} são os valores máximo e mínimo desta variável nos dados analisados, respectivamente.

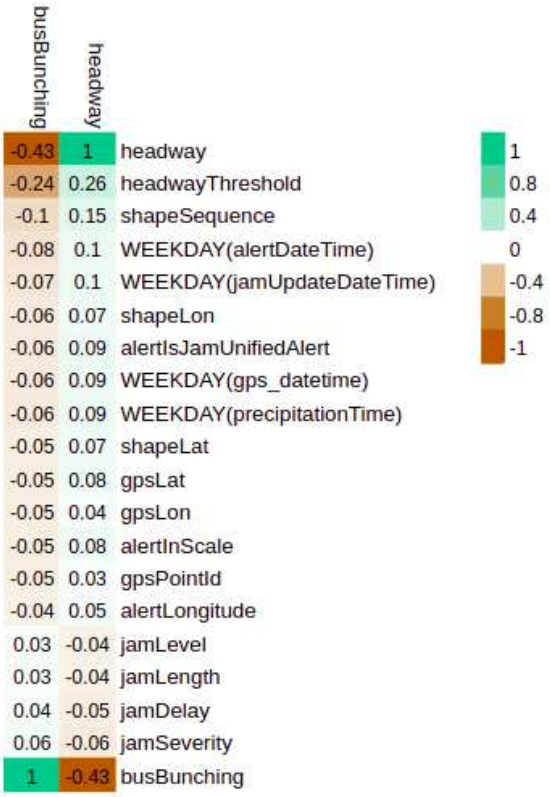

Figura 4.7: Valores da correlação entre as variáveis do modelo utilizando o teste Kendall.

Para realizar este processo em toda a base de dados, é necessário converter as variáveis categóricas em variáveis numéricas. Para isso, foram aplicadas técnicas como *One Hot Encoder*[1] e *Label Encoder*[2]. A primeira técnica divide a coluna da variável categórica em múltiplas colunas, criando uma nova coluna binária para cada categoria da variável. Os valores destas novas colunas são 1s ou 0s, sendo 1 representando o valor daquela instância. A técnica *Label Encoder* transforma os valores das variável categórica em números com valor entre 0 e $n\_categorias - 1$.

Particionamento da Base de Dados

Por fim, o conjunto de dados é subdividido em duas partes: 80% para treinamento e 20% para testar o modelo. A proporção da divisão não é igual porque é necessário disponibilizar para o treinamento do modelo a maior quantidade de dados possível. Esta proporção é comumente

[1] *https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html*, *https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html*

[2] *https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html*

utilizada, mas não é padrão, podendo variar de acordo com a necessidade de cada problema.

4.3.2 Etapa de Treinamento

Após processar os dados e adaptá-los para serem utilizados nos modelos de AM, na etapa de treinamento o modelo proposto é construído e ajustado, como descrito a seguir.

Ajuste dos Parâmetros do Modelo

Nesta etapa, o conjunto de dados de treino é utilizado para encontrar os melhores valores dos parâmetros dos modelos. Para encontrar estes valores[3], foi realizado o ajuste (*tuning*) manual de parâmetros, escolhendo um intervalo de valores com base nos valores padrão de cada API[4] utilizada. O ajuste automático apresentou-se muito custoso em termos de tempo de execução, pois um conjunto de possíveis valores é atribuído para cada parâmetro e todas as combinações resultantes do produto cartesiano desses valores são avaliadas.

Treinamento do Modelo

Após a definição dos valores dos parâmetros, o treinamento do modelo é realizado. Múltiplas iterações são executadas nos dados de treinamento para aprender as relações entre GPS, GTFS, dados de clima e de tráfego com a ocorrência dos aglomerados de ônibus, até atingir a convergência. O resultado dessa fase é o modelo proposto, ou seja, uma função estimada de predição de aglomerados de ônibus (Equação 4.2).

4.3.3 Etapa de Testes

Por fim, na etapa de teste, o modelo proposto é avaliado e disponibilizado para implantação, como descrito nos seguintes passos.

Avaliação do Modelo Proposto

Na última etapa, os dados de teste são utilizados para validar o modelo proposto, comparando cada predição com os rótulos. Para avaliação, as medidas de eficácia utilizadas são

[3]Os melhores valores de parâmetros encontrados estão descritos no Apêndice C

[4]Scikit-learn para *Random Forest*, catboost.ai para CatBoost e xgboost.readthedocs.io para XGBoost.

acurácia, precisão, cobertura e F1.

Implantação do Modelo Proposto

Após a validação, o resultado é comparado com um valor predefinido, a exemplo de $F1 \geq \lambda$. Neste trabalho, se $F1 \geq \lambda = 0,8$, o modelo é usado para prever aglomerados de ônibus em tempo real para dados ainda não considerados; caso contrário, o treinamento deverá ser refeito considerando outros parâmetros, valores, dados ou configurações até atingir a eficácia desejada.

Após implantação do modelo em tempo real, a cada execução do mesmo, os dados de entrada necessitam apenas serem pré-processados (passos 1,2,3,5 e 6 da Figura 4.6) para que o modelo consuma-os e produza as respectivas predições.

A seguir, são apresentados o processo de utilização da aprendizagem incremental no modelo proposto e o processo de predição para múltiplas paradas de ônibus.

4.4 Aprendizagem Incremental

Para atualizar continuamente o modelo proposto, ou seja, incorporar novos dados ao modelo já treinado, é aplicada uma técnica de aprendizado incremental, como ilustrado na Figura 4.8.

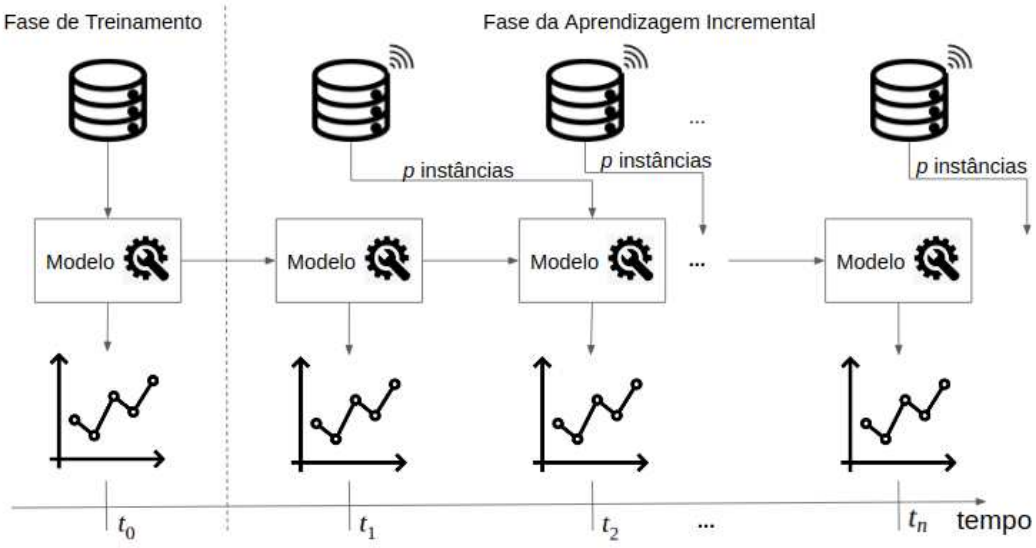

Figura 4.8: Fluxo de execução da aprendizagem incremental.

No tempo t_0, o modelo é treinado com dados históricos e produz a função estimada para predição dos aglomerados de ônibus. Uma vez treinado, o modelo é utilizado para predizer os aglomerados assim que os dados em tempo real forem recebidos; portanto, os dados recebidos no tempo t_1 serão usados para atualizar o modelo no tempo t_2 quando o rótulo for conhecido, e assim por diante. Para isso, o modelo se adapta gradualmente, ou seja, h_{i+1} é construído com base em h_i e nas p instâncias de dados recebidas no intervalo de tempo, sem um retreinamento completo e preservando o conhecimento adquirido anteriormente.

Neste trabalho, a janela de tempo utilizada para realizar a atualização do modelo, isto é, o período de tempo para acumulação de dados, foi de $t_{i+1} - t_i = 3,1$ horas para a Cidade A, cujo intervalo representa aproximadamente 286.526 instâncias, e $t_{i+1} - t_i = 7$ horas para a Cidade de Curitiba, representando aproximadamente 1.242.314 instâncias. Essa janela é definida pelo usuário com base no melhor desempenho do modelo para cada cidade.

Para incorporar a técnica de aprendizagem incremental ao modelo proposto, foram utilizados os seguintes parâmetros em cada modelo-base:

- **init_model:** este parâmetro da função de ajuste no CatBoost permite a incorporação de novos dados em um modelo previamente treinado;
- **xgb_model:** este parâmetro da função de ajuste no XGBoost também permite a incorporação de novos dados em um modelo previamente treinado;
- **warm_start:** este parâmetro no RF permite a incorporação de novos dados em um modelo previamente treinado, adicionando mais árvores para os novos dados.

Portanto, a cada atualização incremental (janela de dados), os modelos-base são treinados a partir do modelo anterior e apenas com os novos dados de entrada.

A aplicação da aprendizagem incremental ao modelo proposto, onde cada modelo-base é atualizado individualmente, proporciona um pequeno aumento na eficácia ao considerar a atualização do modelo, como será mostrado no Capítulo 5.

4.5 Predição de Múltiplas Paradas Consecutivas

Os agentes de transporte público necessitam saber com que antecedência um aviso de aglomerados de ônibus pode ser lançado. Quanto mais cedo um aglomerado for previsto, mais

eficazes serão as estratégias de controle reativas aplicadas pelos agentes [58]. Neste sentido, o modelo proposto foi atualizado para considerar a predição de múltiplas paradas.

Considerando a parada de ônibus k de determinada rota como referência, deseja-se realizar as predições para pelo menos as $k + n$ paradas de ônibus seguintes da mesma rota com o objetivo de facilitar a tomada de decisão dos agentes de transporte público. No exemplo da Figura 4.9, tendo em vista que os ônibus analisados são b_1 e b_2, atendendo a mesma rota, o modelo proposto realiza a predição da ocorrência de aglomerado entre esses dois ônibus para as $n = 5$ paradas seguintes: s_1, s_2, s_3, s_4 e s_5.

Figura 4.9: Exemplo da predição de aglomerados de ônibus para cinco paradas consecutivas.

Considerando a próxima parada de ônibus k como referência, os modelos típicos de AM fazem a predição da ocorrência do aglomerado para esta parada. É necessário atualizar o modelo e/ou os dados de treinamento para considerar o horizonte de predição desejado, conforme indicado na Seção 2.2.4.

Neste trabalho, foi aplicada a técnica em que o modelo é ajustado para produzir múltiplas saídas, ou seja, uma sequência de predições para as próximas paradas de ônibus. Esta técnica apresenta as seguintes vantagens: (i) a alteração apenas nos dados de entrada do modelo é relativamente mais simples que as outras abordagens; (ii) a possível interdependência entre as múltiplas predições não é perdida; e (iii) ao não utilizar as predições anteriores como entrada, possíveis erros não são acumulados. Dessa forma, para realizar tal tarefa de predição, os dados e o modelo proposto precisam ser atualizados com a informação dos rótulos das

paradas seguintes.

Em geral, uma instância dos dados de treinamento possui um conjunto de variáveis X e seu rótulo y_1, conforme destacado na Etapa 1 da Figura 4.10. No entanto, para fazer predições para as próximas n paradas de ônibus, os dados de treinamento são atualizados com os próximos rótulos $y_2, ..., y_n$ de cada instância dos dados. Em seguida, o *ensemble* é treinado com esse conjunto de dados atualizado. Assim, para cada instância de dados de entrada, os modelos-base predizem os valores para as próximas n paradas (Etapa 2). Por fim, na Etapa 3, a abordagem de votação calcula a predição final do aglomerado de ônibus para as próximas n paradas para cada dado de entrada.

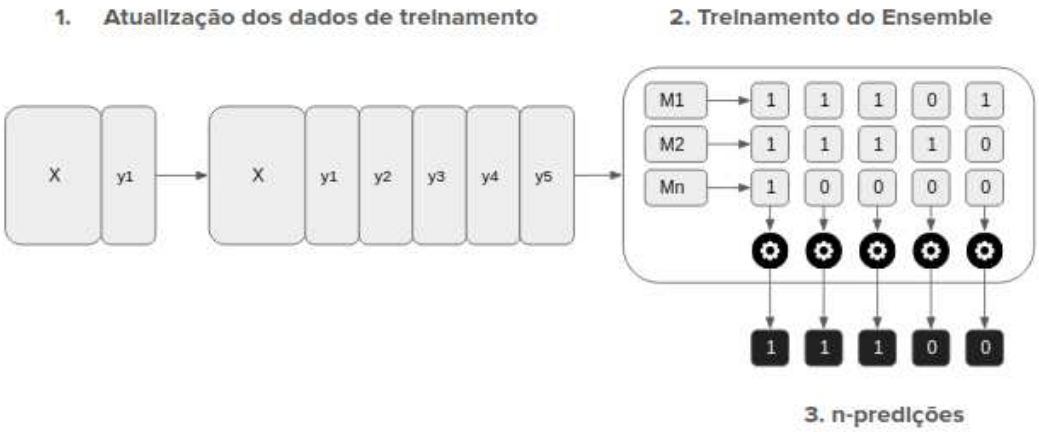

Figura 4.10: Fluxo de execução para predição de aglomerados de ônibus em múltiplas paradas consecutivas.

4.6 Considerações Finais

Neste capítulo, foram apresentados a formalização do problema, alguns padrões que indicam a ocorrência dos aglomerados de ônibus em relação ao horário do dia e dias da semana e, por fim, a solução proposta. Foi apresentado o modelo de predição de aglomerados de ônibus que considera a técnica de *ensemble* para combinar os modelos RF, XGBoost e CatBoost, além de empregar dados de quatro fontes: GPS, GTFS, clima e situação da via. As etapas do fluxo de execução do modelo e os passos de cada uma foram descritos em detalhes, bem como a aplicação da técnica de aprendizagem incremental e a predição para múltiplas paradas consecutivas.

No capítulo seguinte, será apresentada a avaliação experimental do modelo, desde a metodologia empregada até a discussão dos resultados alcançados.

Capítulo 5

Avaliação Experimental

Neste capítulo, é apresentada a avaliação experimental realizada no modelo proposto. Os experimentos foram projetados com o intuito de mensurar a eficácia e eficiência do modelo, principalmente em termos de $F_{measure}$, tempo de predição e horizonte de predição. O modelo proposto foi comparado com outros já empregados no problema analisado: Regressão Linear, Regressão Logística, Support Vector Machine e Relevance Vector Machine. Para avaliar tais modelos, foram utilizadas bases de dados reais de duas cidades brasileiras, cujos detalhes são apresentados na Seção 5.5.

Para conduzir a avaliação experimental do modelo proposto nesta pesquisa, os principais passos da metodologia incluíram a definição: (i) das questões de pesquisa; (ii) das métricas de avaliação; (iii) dos testes estatísticos; (iv) da técnica de ajuste de hiperparâmetro; e, por fim, (v) dos dados a serem avaliados.

Na Seção 5.1, a seguir, são apresentadas as questões de pesquisa que motivaram a execução dos experimentos. Na Seção 5.2, são descritas as métricas utilizadas para mensurar o modelo avaliado. Já na Seção 5.3, são definidos os testes estatísticos empregados nos experimentos para avaliar a significância estatística dos resultados. Na Seção 5.4, são apresentadas as técnicas de ajuste de hiperparâmetros empregadas. Na Seção 5.5, são apresentadas as bases de dados utilizadas e o processo de integração das mesmas. Na Seção 5.6, são descritos os diferentes cenários avaliados e as suas configurações. Na Seção 5.7, é apresentado o resumo dos resultados de cada experimento. Na Seção 5.8, são discutidas as ameaças à validade do modelo proposto. Por fim, na Seção 5.9, são apresentadas as considerações finais deste capítulo.

5.1 Questões de Pesquisa

Para guiar a avaliação do modelo proposto de predição de aglomerados de ônibus, as questões de pesquisa (QP) foram divididas em três cenários de avaliação.

O primeiro cenário engloba questões de pesquisa relativas à **eficácia** do modelo:

- **QP1:** A combinação (*ensemble*) de modelos de AM baseados em árvores de decisão permite predizer com eficácia a ocorrência de aglomerados de ônibus em diferentes cidades?
- **QP2:** A combinação de modelos de AM para predizer aglomerados de ônibus produz um resultado superior em relação ao uso de modelos individuais?
- **QP3:** O modelo de predição proposto apresenta uma diferença significativa, em termos de eficácia, quando a quantidade de dados de treinamento utilizada é aumentada?
- **QP4:** Qual(is) fonte(s) de dados é(são) a(s) mais relevante(s) para predizer aglomerados de ônibus: GPS, GTFS, situação do clima e/ou situação do trânsito?
- **QP5:** A aplicação da aprendizagem incremental ao modelo proposto melhora a eficácia das predições de aglomerados de ônibus?

O segundo cenário engloba uma questão relacionada à **eficiência** do modelo, em termos de tempo para predição e horizonte de predição:

- **QP6:** A combinação de modelos baseados em árvores de decisão permite uma predição eficiente, em termos de tempo para predição e horizonte de predição, da ocorrência de aglomerados de ônibus em tempo real?

O terceiro cenário considera a questão relacionada à comparação do modelo proposto com os modelos competidores:

- **QP7:** O modelo proposto possui eficácia superior em relação aos modelos de AM utilizados no estado da arte para predizer a ocorrência de aglomerados de ônibus?

5.2 Métricas Utilizadas

Para responder às questões de pesquisa definidas na seção anterior, são utilizadas as seguintes métricas de avaliação:

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN} \tag{5.1}$$

$$Precisão = \frac{VP}{VP + FP} \tag{5.2}$$

$$Cobertura = \frac{VP}{VP + FN} \tag{5.3}$$

$$F_{measure} = \frac{2 \times Precisão \times Cobertura}{Precisão + Cobertura} \tag{5.4}$$

onde VP (Verdadeiro Positivo) representa as ocorrências dos aglomerados de ônibus corretamente identificadas e VN (Verdadeiro Negativo) representa as não ocorrências de aglomerados corretamente identificadas. Além disso, FN (Falso Negativo) representa as ocorrências de aglomerados de ônibus que não foram identificadas e FP (Falso Positivo) representa os aglomerados previstos que não ocorreram. Essas medidas produzem o valor 0 ou 1 para cada par de instâncias avaliadas (valor previsto e seu rótulo). Dessa maneira, $Acurácia$, $Precisão$, $Cobertura$ e $F_{measure}$ apresentam valores entre $[0, 1]$, de maneira que, quão mais próximos de 1, melhor o resultado.

Nesta pesquisa, ao lidar com um problema de classes desbalanceadas, a principal métrica de eficácia a ser considerada é a $F_{measure}$, pois ela representa a combinação de quanto o modelo acertou em termos de precisão e cobertura.

5.3 Testes Estatísticos

Para avaliar a significância dos resultados experimentais desta pesquisa, dois testes estatísticos não-paramétricos foram aplicados: o Teste de Mann-Whitney e o Teste de Friedman. Os testes não-paramétricos não assumem distribuição específica dos dados [15; 10]; assim, são indicados em cenários em que a distribuição das amostras de dados é desconhecida.

Os testes foram empregados de acordo com as características de paridade das amostras de dados utilizadas em cada experimento. A paridade diz respeito à distribuição dos dados. Amostras pareadas são aquelas relacionadas [15], que foram extraídas da mesma população. Por sua vez, amostras não-pareadas são aquelas independentes, que não são relacionadas.

Basicamente, para cada cenário de avaliação dos experimentos conduzidos, a hipótese nula diz respeito à não haver diferença significativa nos resultados analisados, isto é, na distribuição das predições avaliadas. Por sua vez, a hipótese alternativa refere-se à diferença significativa nos resultados analisados; com indícios de diferença na distribuição das predições. Nos experimentos conduzidos, a hipótese nula é rejeitada quando o *p_value* $< 0,05$, aceitando-se, portanto, a hipótese alternativa.

5.3.1 Teste de Mann-Whitney

O Teste de Mann-Whitney [34], em homenagem à Henry Mann e Donald Whitney, é um teste de significância estatística não-paramétrica para determinar se duas amostras independentes foram retiradas de uma população com a mesma distribuição [10]. A hipótese nula é que não há diferença entre as distribuições das amostras de dados. A rejeição dessa hipótese sugere que provavelmente há alguma diferença entre as amostras. Mais especificamente, o teste determina se é igualmente provável que qualquer valor selecionado aleatoriamente de uma amostra seja maior ou menor do que um valor na outra distribuição. Se violado, sugere distribuições diferentes.

5.3.2 Teste de Friedman

Para avaliar se mais de duas amostras diferentes têm a mesma distribuição ou não, o Teste de Friedman [22] pode ser utilizado. Tal teste, nomeado em homenagem à Milton Friedman, é não-paramétrico e considera que as amostras são pareadas [10]. A hipótese nula é que as várias amostras pareadas têm a mesma distribuição. Uma rejeição da hipótese nula indica que uma ou mais das amostras tem uma distribuição diferente.

5.4 Ajuste dos Hiperparâmetros

O ajuste de hiperparâmetros é o processo de otimização automática dos valores dos hiperparâmetros de um modelo de ML. Os hiperparâmetros referem-se a todos os parâmetros de um modelo que não são atualizados durante a fase de treinamento e são usados para configurar o modelo [6]. Este processo define a combinação dos melhores valores (dentro do conjunto fornecido) dos hiperparâmetros a serem utilizados.

Algumas técnicas comumente utilizadas para realizar este processo são a Busca Manual (Manual Search), Busca Aleatória (Random Search) e Busca em Grade (Grid Search) [6; 5]. O primeiro refere-se à escolha manual de alguns hiperparâmetros do modelo com base no conhecimento do domínio. Em seguida, o modelo é treinado e avaliado com os valores escolhidos. Este processo repete-se até que uma eficácia satisfatória seja obtida. Na busca aleatória, uma grade dos valores dos hiperparâmetros é criada e então combinações aleatórias são extraídas para testar no modelo. Por fim, na Grid Search, uma grade dos valores dos hiperparâmetros também é criada e todas as possíveis combinações de valores são avaliadas no modelo. As duas últimas apresentam a desvantagem do elevado custo de processamento, em relação ao tempo de execução e, dependendo do modelo, ao consumo de memória também.

Nesta pesquisa, foram utilizadas a Busca Manual, que apresentou os melhores resultados, e a Grid Search para avaliação mais profunda do modelo proposto, em conjunto com a aplicação dos testes estatísticos. Devido às limitações computacionais, mas para permitir a avaliação com mais possibilidades de valores de parâmetros, foi realizada a Grid Search no modelo proposto com uma base de dados com 5% da quantidade total de dados de cada cidade analisada. Assim, os melhores valores foram utilizados nos demais cenários de experimentos.

5.5 Bases de Dados

Neste trabalho, são utilizadas quatro fontes de dados: GPS, GTFS, clima e situação do trânsito. Esses dados foram integrados de acordo com medidas de distância entre os pontos de geolocalização de cada instância de dados.

É importante destacar que as fontes de dados GPS e GTFS não possuem pontos de

interseção explícitos, que possam ser usados para integrar facilmente suas informações. Para superar esse desafio, foram utilizadas adaptações de duas abordagens de correspondências entre entidades baseadas no *framework* de processamento distribuído Apache Spark[1]: BULMA (*BUs Line MAtching*) e BUSTE (*BUs Stop Ticketing Estimation*) [35; 7].

O algoritmo BULMA faz a correspondência da trajetória do ônibus com o *shape* correto, quando existem vários *shapes* para a mesma rota na base do GTFS. Por sua vez, o algoritmo BUSTE faz a associação dos dados de embarque dos passageiros (extraídos da fonte AFC) com as paradas de ônibus e as trajetórias do GPS. O BUSTE baseia-se na saída do BULMA, reunindo um conjunto de dados que descreve as viagens de ônibus da cidade e os embarques de passageiros associados ao longo de um período de tempo.

Neste trabalho, o BULMA foi utilizado para realizar a integração dos dados de GPS e GTFS, ou seja, associar cada viagem dos ônibus com o respectivo *shape*. O algoritmo BUSTE foi adaptado para considerar apenas a etapa de interpolação dos horários das paradas de ônibus, descartando as etapas que lidam com os dados AFC.

O processo de integração das fontes de dados é realizado em três etapas. Primeiro, são integrados os dados de GPS com GTFS, porque o GPS contém as informações de localização do ônibus e o GTFS contém a localização das paradas de ônibus, que são as unidades de avaliação dos aglomerados. Em seguida, são integrados os dados de clima e de trânsito com as paradas de ônibus nos respectivos horários. Este processo de integração é descrito a seguir e destacado na Figura 5.1.

1. **Integração das fontes GPS e GTFS:** para associar cada ponto de GPS ao ponto de *shape* mais próximo (Figuras 5.1a e 5.1b), foi utilizado o algoritmo BULMA. Este algoritmo tem o objetivo de identificar o *shape* exato executado pelo ônibus, além de corrigir erros de GPS, como pontos de geolocalização fora da rua. Em seguida, os horários de chegada dos ônibus nas paradas sem registros de GPS são interpolados, usando a adaptação do algoritmo BUSTE. Para isso, são utilizados os dados dos pontos GPS enviados antes e depois da parada, conforme mostrado na Figura 5.1c. Nesta etapa, são mantidos apenas os pontos de GPS associados aos pontos de parada de ônibus, ou seja, são descartados os dados enviados entre as paradas (Figura 5.1d);

[1] *https://spark.apache.org/*

2. **Integração das fontes GPS, GTFS e situação do clima:** para cada parada de ônibus, são associados os dados de clima relacionados à precipitação mais recentes - enviados uma hora antes ou até uma hora depois do tempo do ponto GPS. Os dados são coletados da estação meteorológica mais próxima da parada de ônibus analisada, conforme exibido na Figura 5.1e;

3. **Integração das fontes GPS, GTFS, situação do clima e do trânsito:** como mostrado na Figura 5.1f, para cada parada de ônibus, são associados os dados de tráfego que atendem as seguintes condições: i) são relacionados à mesma rua do ponto GPS e ii) são enviados antes ou ao mesmo tempo que o ponto de GPS. Quando nenhum dado de trânsito é encontrado para a rua em questão, é considerada a situação de trânsito "NORMAL".

Na avaliação experimental, bases de dados de duas cidades brasileiras foram utilizadas, considerando todas as rotas e todos os ônibus: Curitiba e Cidade A[2]. Em relação a Curitiba, os dados foram coletados em maio de 2019, enquanto que os dados da Cidade A foram coletados em dezembro de 2018. O intervalo de tempo entre os pontos de GPS nos dados históricos para essas cidades é, em geral, de 30 segundos, embora para Curitiba solicitações em tempo real possam ser feitas apenas a cada 120 segundos. As características das bases de dados integradas[3] são apresentadas na Tabela 5.1.

Tabela 5.1: Descrição das bases de dados utilizadas nos experimentos.

Cidade (BR)	Tamanho (GB)	Linhas (Instâncias)	Colunas	Dias	Rotas	Ônibus	Aglomerados (%)
Curitiba	3,7	6.211.570	119	31	219	1.496	5,6
Cidade A	0,7517	1.432.633	111	12	106	789	13,5

[2]Devido a termos contratuais, não é possível fornecer o nome e os dados da cidade.

[3]A base da Cidade A possui menos colunas porque algumas delas não contêm valores para os dias analisados e foram removidas. Por exemplo, aquelas relacionados à data e hora da expiração do alerta de congestionamento.

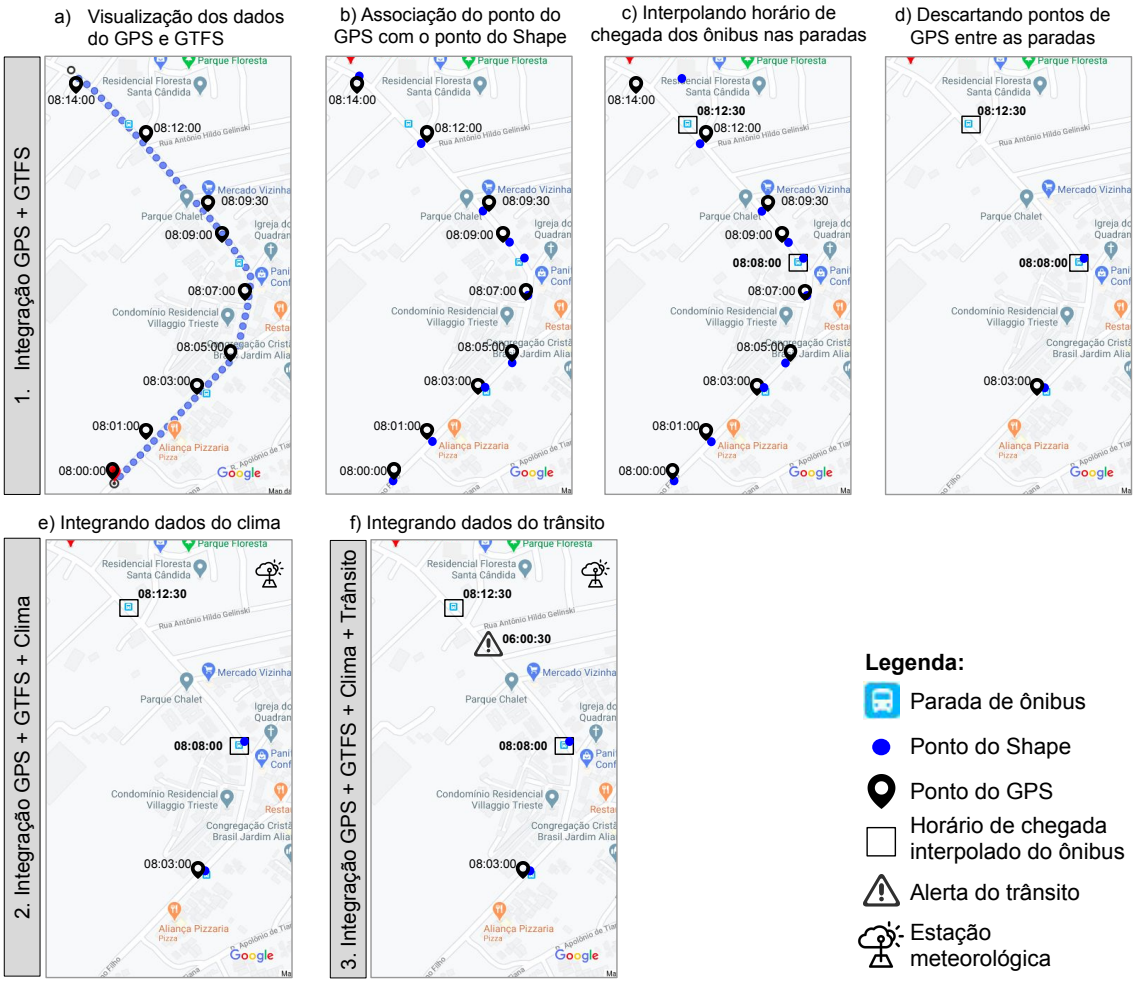

Figura 5.1: Passos da integração dos dados de GPS, GTFS, clima e trânsito. Mapa do fundo recuperado do Google Maps.

5.6 Experimentos

Nesta seção, são apresentados os experimentos utilizados para avaliar o modelo proposto[4], considerando diferentes perspectivas: combinação dos modelos, quantidade de dados de treinamento utilizada, relevância das variáveis, uso de aprendizagem incremental, tempo de predição, horizonte de predição e comparação com outros modelos competidores. Essas perspectivas foram agrupadas em três cenários: avaliação de eficácia, avaliação de eficiência e comparação com modelos competidores. Em cada cenário de avaliação serão exibidos e discutidos os resultados obtidos por meio do ajuste dos parâmetros com busca manual,

[4]O código fonte desta pesquisa e os dados de Curitiba podem ser encontrados no repositório *https://github.com/veruskasantos/INESProject*.

seguidos dos resultados obtidos com o *grid search*.

Os experimentos foram realizados em um computador com Ubuntu 18.04 64-bits, Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, 32GB de RAM, 12CPUs.

5.6.1 Avaliação da Eficácia

Para mensurar a eficácia, o modelo proposto foi avaliado considerando quatro perspectivas: combinação dos modelos, quantidade de dados de treinamento utilizada, relevância das variáveis e aplicação da aprendizagem incremental.

Combinação dos Modelos

Alguns autores [2; 36; 59] classificam as rotas em relação à frequência (quantidade de ônibus servindo-a) e demostram que os modelos obtém melhor desempenho em alguns tipos de rotas. Por exemplo, o modelo dos autores em [36], apresenta desempenho inferior nas rotas de baixa frequência. Neste trabalho, o modelo proposto tem o intuito de ser aplicado em todas as rotas das cidades avaliadas.

Neste experimento, o *ensemble* proposto foi aplicado em todas as rotas de ônibus de Curitiba e da Cidade A. Os resultados são exibidos na Figura 5.2. No gráfico desta figura e das seguintes, o eixo horizontal representa as medidas de eficácia e o eixo vertical representa os valores dessas medidas. Especialmente na Figura 5.2, a luminosidade da cor representa as cidades analisadas.

Em relação à QP1, a combinação de modelos de AM baseados em árvores de decisão permite predizer com $F_{measure} \geq 80\%$ a ocorrência de aglomerados de ônibus em diferentes cidades, apresentando $precisão \geq 88\%$ e $cobertura \geq 72\%$. Esses resultados indicam considerável confiabilidade nas predições do modelo proposto. Diferentemente de alguns trabalhos, o *ensemble* proposto foi avaliado considerando todas as rotas de ônibus de cada cidade, sendo os valores de eficácia semelhantes para ambas, indicando a adaptabilidade do modelo para diferentes cidades e tipos de rotas.

Os resultados obtidos com o ajuste de parâmetros com *grid search* são exibidos na Figura 5.3, cujos resultados de $cobertura$ e $F_{measure}$ foram aproximadamente 30% e 20% inferior na Cidade A, respectivamente, e apresentaram pouca diferença para os dados da cidade de Curitiba. O teste de Mann-Whitney, aplicado no conjunto das predições de cada cidade,

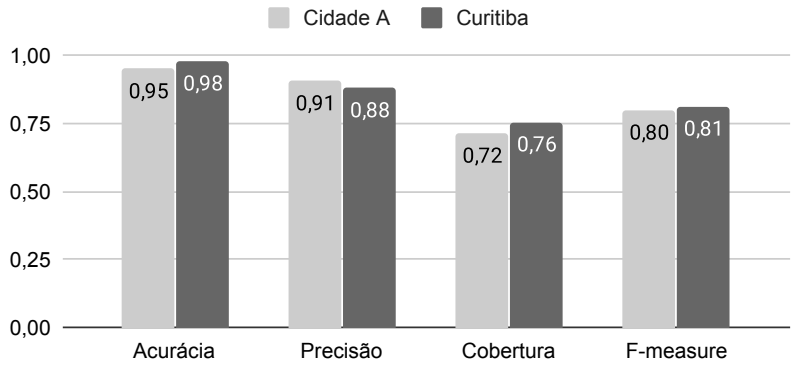

Figura 5.2: Avaliação da eficácia do modelo de predição de aglomerados de ônibus por cidade. Ajuste de parâmetros com busca manual.

apresentou *estatística* $= 175073905579,5$ e *p-value* ≈ 0.0, aceitando, assim, a hipótese alternativa de que há provável diferença significativa nesses resultados.

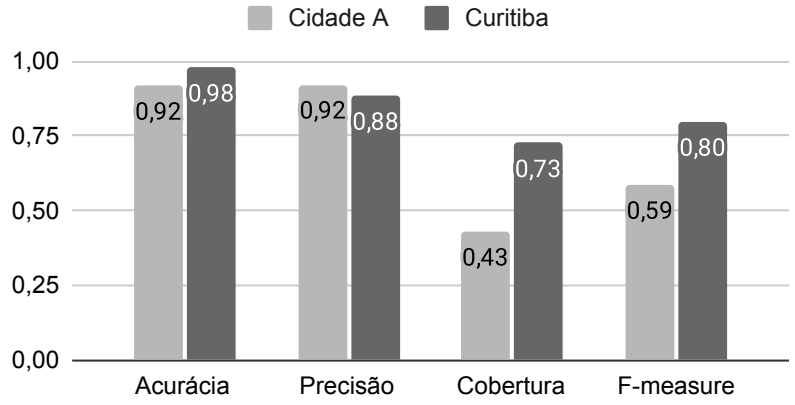

Figura 5.3: Avaliação da eficácia do modelo de predição de aglomerados de ônibus por cidade. Ajuste de parâmetros com *grid search*.

Em relação à QP2, a Figura 5.4 apresenta uma comparação em termos de eficácia entre o *ensemble* proposto e seus modelos-base treinados individualmente, ambos para a Cidade A. Nesta figura, a luminosidade da cor representa os modelos avaliados. A ideia de usar uma combinação de modelos é complementar o erro de um modelo com o acerto de outro modelo. Assim, o *ensemble* proposto tem pelo menos o mesmo resultado que o melhor modelo-base. Neste experimento, o modelo-base com maior pontuação de $F_{measure}$ é o RF, embora sua precisão não seja a mais alta. Devido à combinação de predições de modelos, o *ensemble*

conseguiu superar o valor da precisão do RF.

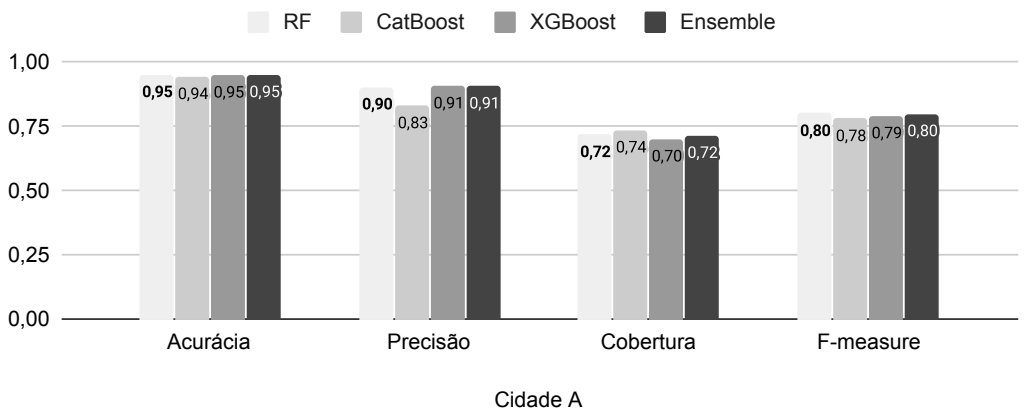

Figura 5.4: Comparação da eficácia do *ensemble* e dos modelos-base individuais (Cidade A). Ajuste de parâmetros com busca manual.

Ainda em relação à QP2, os experimentos conduzidos com o ajuste de parâmetros com *grid search*, aplicados nos dados da Cidade A e Curitiba, têm os resultados apresentados nas Figuras 5.5 e 5.6, respectivamente. Com os dados da Cidade A, percebe-se que o desempenho do XGBoost foi bem inferior aos demais modelos-base, que também foram inferiores quando comparados com os resultados usando os valores de parâmetros da busca manual (Figura 5.4), interferindo diretamente na qualidade do *ensemble*. O teste de Friedman, aplicado no conjunto das predições de cada modelo avaliado, apresentou *estatística* $= 39528,7$ e *p-value* ≈ 0.0, aceitando, assim, a hipótese alternativa de que há diferença nas distribuições das predições de cada modelo.

Por sua vez, o CatBoost foi o modelo-base do *ensemble* com menor desempenho, quando aplicado aos dados de Curitiba (Figura 5.6); entretanto, a eficácia geral do *ensemble* neste cenário ainda manteve-se similar quando utilizado os valores da busca manual. De modo parecido, o teste de Friedman, aplicado no conjunto das predições de cada modelo avaliado, apresentou *estatística* $= 10547,3$ e *p-value* ≈ 0.0, aceitando, assim, a hipótese alternativa de que há diferença nas distribuições das predições de cada modelo.

Quantidade de Dados de Treinamento Utilizada

Lidar com dados desbalanceados, como no cenário dos aglomerados de ônibus, é sempre um desafio. Na Cidade A, por exemplo, a proporção dos aglomerados é de 1:7, ou seja, para cada

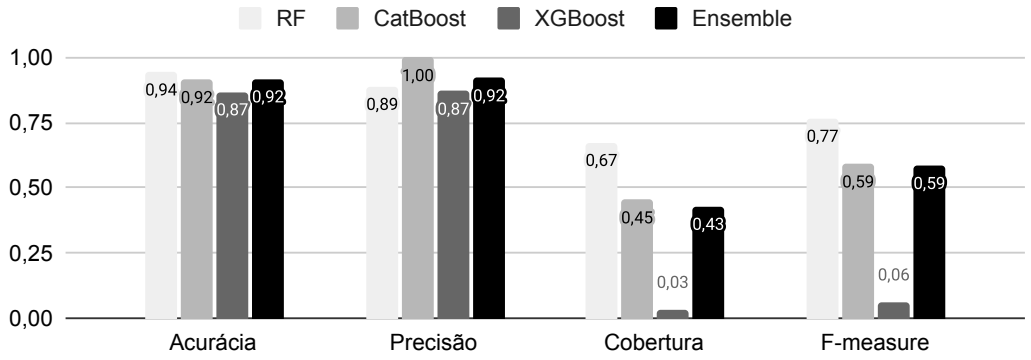

Figura 5.5: Comparação da eficácia do *ensemble* e dos modelos-base individuais (Cidade A). Ajuste de parâmetros com *grid search*.

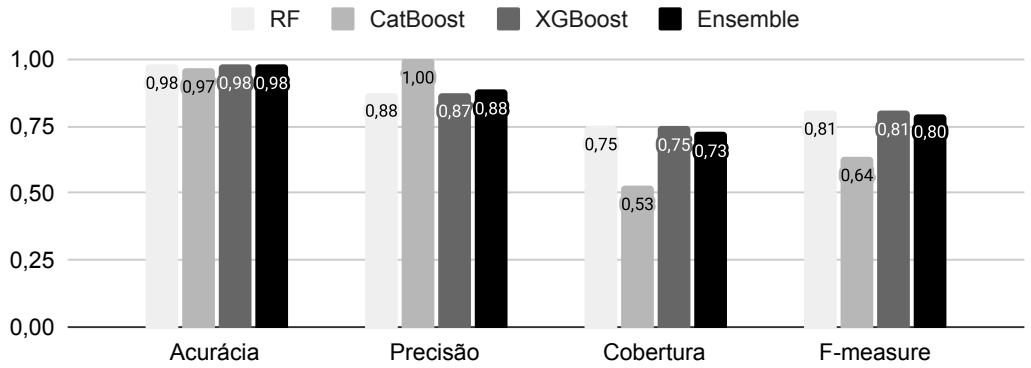

Figura 5.6: Comparação da eficácia do *ensemble* e dos modelos-base individuais (Curitiba). Ajuste de parâmetros com *grid search*.

sete instâncias sem aglomerados da base de dados, uma instância apresenta o aglomerado de ônibus. Enquanto isso, nos dados de Curitiba a proporção é de aproximadamente 1:18. Na Figura 5.7, são apresentados os resultados do uso de diferentes quantidades de dados para predizer os aglomerados considerando o desbalanceamento das classes, onde a luminosidade da cor representa a quantidade de dados utilizada para o treinamento do modelo.

Os dados da Cidade A foram divididos em cinco subconjuntos: 5%, 25%, 52%, 75% e 100% de todo o conjunto de dados; esses subconjuntos correspondem a dados coletados ao longo de 1, 4, 6, 9 e 12 dias, respectivamente. Neste experimento, a base de teste considerada foi a mesma para os cinco subconjuntos utilizados, neste caso, 20% dos dados da base completa. Em relação à QP3, esses resultados indicam que, quanto mais dados, melhores os resultados de eficácia do modelo proposto, ao considerar também o desbalanceamento das

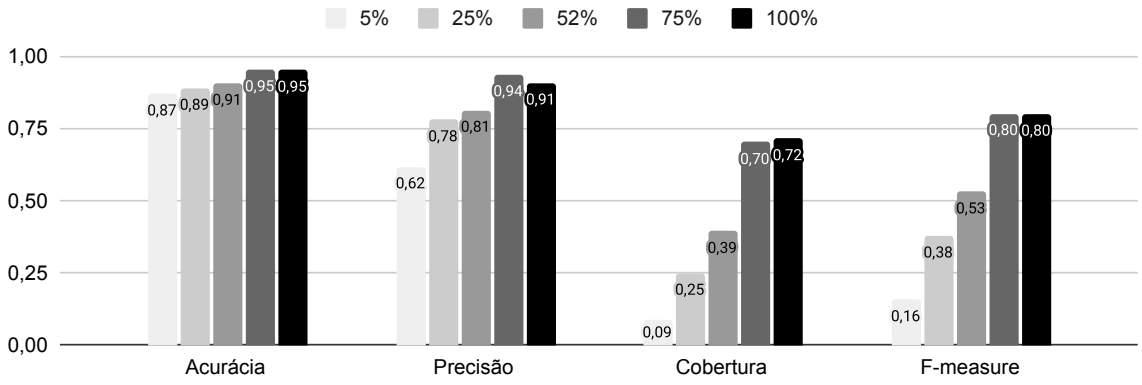

Figura 5.7: Avaliação do impacto da quantidade de dados na eficácia do modelo de predição de aglomerados de ônibus (Cidade A). Ajuste de parâmetros com busca manual.

classes.

Os resultados obtidos por meio do ajuste de parâmetros com *grid search* são exibidos nas Figuras 5.8 e 5.9, com os dados da Cidade A e Curitiba, respectivamente. Assim como na busca manual, esses resultados também confirmam que, o aumento na quantidade de dados de treinamento utilizados, tende a aumentar a eficácia do modelo proposto. A exceção ocorre ao utilizar 75% da base de dados em ambos as cidades, cuja eficácia já aproxima-se da até então obtida com 100% dos dados na Cidade A, e em Curitiba, a eficácia foi 4% superior em relação à utilização da base completa.

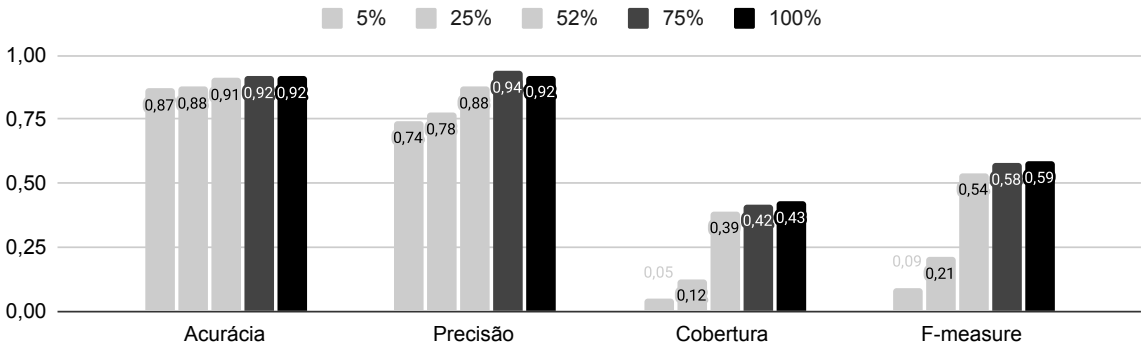

Figura 5.8: Avaliação do impacto da quantidade de dados na eficácia do modelo de predição de aglomerados de ônibus (Cidade A). Ajuste de parâmetros com *grid search*.

O teste de Friedman, aplicado no conjunto das predições de cada modelo avaliado na Figura 5.8, apresentou *estatística* $= 33142,6$ e *p-value* ≈ 0.0, aceitando, assim, a hipótese

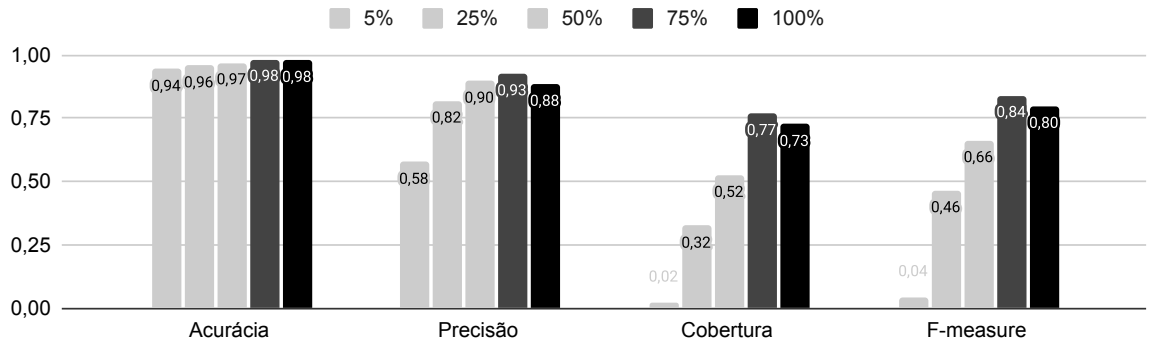

Figura 5.9: Avaliação do impacto da quantidade de dados na eficácia do modelo de predição de aglomerados de ônibus (Curitiba). Ajuste de parâmetros com *grid search*.

alternativa de que há diferença nas distribuições das predições de cada modelo. Da mesma forma, no experimento da Figura 5.9, o mesmo teste foi aplicado e apresentou *estatística* $= 110284.5$ e *p-value* ≈ 0.0, indicando conclusão similar: aceitação da hipótese alternativa de que há diferença nas distribuições das predições de cada modelo.

Relevância das Variáveis

Para avaliar quais fontes de dados têm mais influência sobre a predição dos aglomerados, elas foram testadas individualmente, exceto GPS e GTFS, devido à sua interdependência: GTFS representa os dados programados dos ônibus e não pode ser usado individualmente para predizer aglomerados de ônibus.

Na Figura 5.10, é exibida a avaliação do modelo proposto com diferentes combinações das fontes de dados, onde a luminosidade da cor representa estas combinações. Em relação à QP4, os resultados mostram que as fontes de dados apresentam significância semelhante, com destaque especial para GPS + GTFS, que são as fontes primordiais a serem utilizadas. Além disso, é possível observar que a adição dos dados das fontes de situação do trânsito e do clima não apresentou melhoria na eficácia do modelo proposto.

Os resultados do ajuste de parâmetros com *grid search* são exibidos nas Figuras 5.11 e 5.12 para os dados da Cidades A e Curitiba, respectivamente. Assim como observado com os valores obtidos com a busca manual, as fontes de GPS+GTFS são as mais revelantes para serem utilizadas no modelo proposto, apresentando destaque na sua eficácia. Neste caso, a adição da fonte de clima também manteve a qualidade.

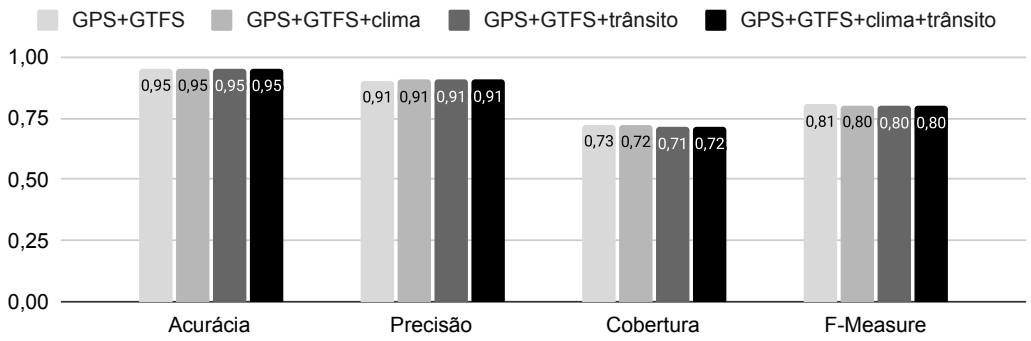

Figura 5.10: Avaliação de múltiplas combinações de fontes de dados (Cidade A). Ajuste de parâmetros com busca manual.

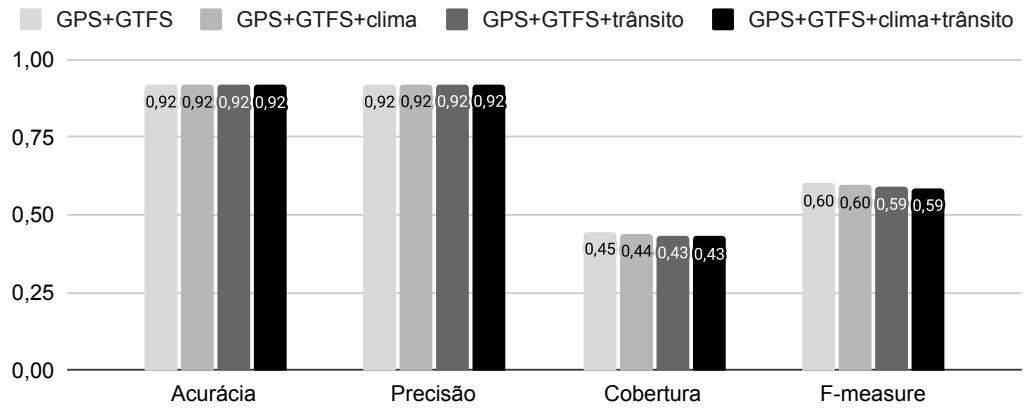

Figura 5.11: Avaliação de múltiplas combinações de fontes de dados (Cidade A). Ajuste de parâmetros com *grid search*.

Além disso, é possível notar que, no caso de uma menor quantidade de dados de treinamento, apenas os dados de GPS+GTFS poderia ser suficiente para atingir os mesmos resultados de eficácia, o que economiza o tempo de coleta de dados dessas fontes, neste caso. O acréscimo da situação do trânsito e do clima ajuda a aumentar as informações do modelo, mantendo a eficácia, pois a previsão do aglomerado de ônibus é um problema complexo e a avaliação do modelo proposto considera todas as rotas e ônibus da cidade. Portanto, mais dados de treinamento são necessários para identificar padrões de tráfego e características meteorológicas, bem como aumentar a eficácia.

Por fim, o teste de Friedman, aplicado no conjunto das predições de cada modelo avaliado na Figura 5.11, apresentou *estatística* $= 156,3$ e *p-value* ≈ 0.0, aceitando, assim,

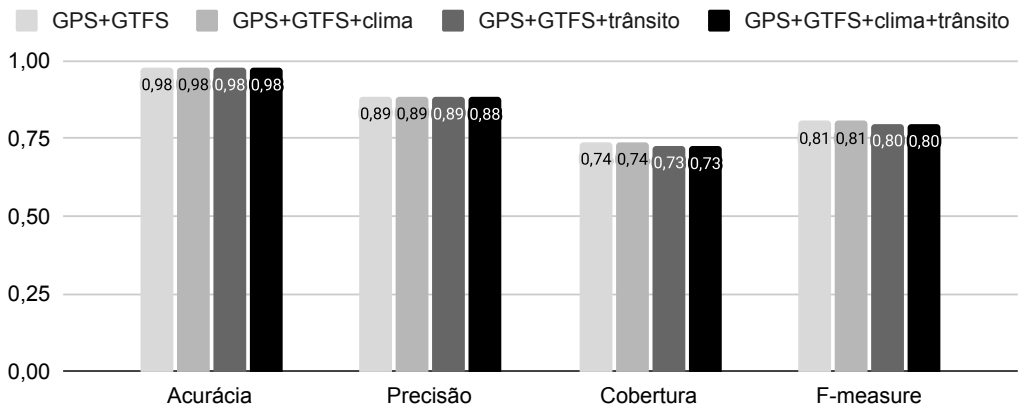

Figura 5.12: Avaliação de múltiplas combinações de fontes de dados (Curitiba). Ajuste de parâmetros com *grid search*.

a hipótese alternativa de que há diferença nas distribuições das predições de cada modelo. Da mesma forma, no experimento da Figura 5.12, o mesmo teste foi aplicado e apresentou *estatística* $= 229$ e *p-value* ≈ 0.0, indicando conclusão similar: aceitação da hipótese alternativa de que há diferença nas distribuições das predições de cada modelo.

Aplicação da Aprendizagem Incremental

Na Figura 5.13, são apresentados os resultados da aplicação da aprendizagem incremental ao modelo proposto, onde a luminosidade da cor representa a janela de incrementos aplicada. Foram simulados incrementos de 20% (ou seja, 286.526 instâncias) da quantidade total de dados da Cidade A. Os incrementos correspondem a aproximadamente 3 horas de coleta de dados, ou seja, o modelo é atualizado em intervalos de 3h. Para realizar esse experimento, o conjunto de dados completo foi dividido em três subconjuntos: 40% para treinar, 40% para simular incrementos e 20% para testar o modelo. Nesse caso, os 40% dos dados usados para simular incrementos permitem avaliar dois incrementos de 20%. Como é possível visualizar na figura, a aplicação da técnica de aprendizagem incremental melhora moderadamente a qualidade das predições dos aglomerados: mais incrementos provavelmente continuarão a aumentar a eficácia (QP5).

Da mesma forma, os experimentos foram conduzidos com o ajuste de parâmetros utilizando o *grid search* e também para os dados de Curitiba. Neste último, a janela de 20% de incrementos corresponde a $1.242.314$ instâncias, ou aproximadamente 7 horas de coleta de

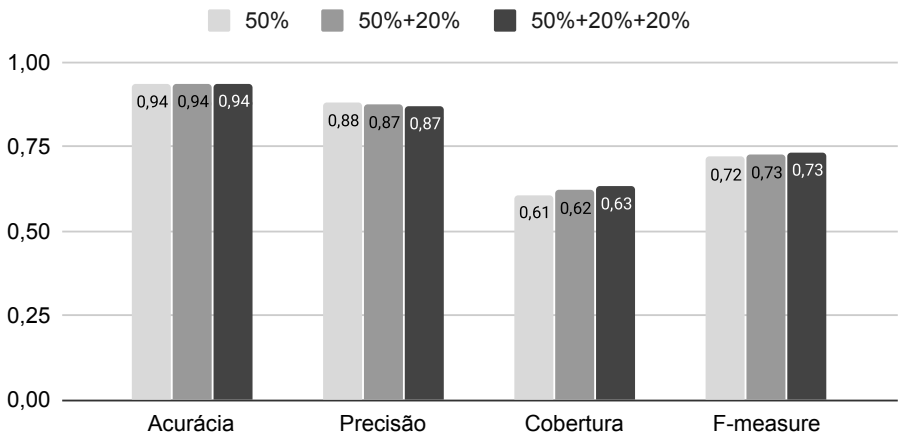

Figura 5.13: Avaliação da aprendizagem incremental (Cidade A). Ajuste de parâmetros com busca manual.

dados. Os resultados desta técnica com os dados da Cidade A e de Curitiba são exibidos nas Figuras 5.14 e 5.15, respectivamente. Os experimentos também mostraram resultados similares, com indicações de aumento na eficácia ao aplicar a aprendizagem incremental.

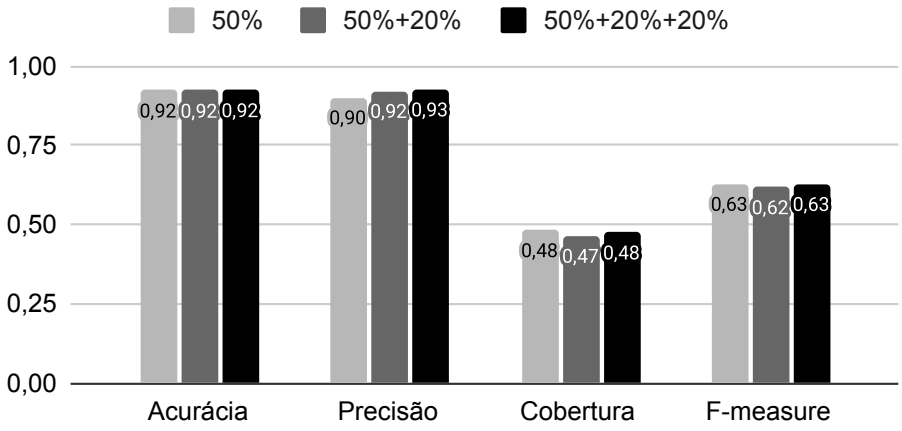

Figura 5.14: Avaliação da aprendizagem incremental (Cidade A). Ajuste de parâmetros com *grid search*.

O teste de Friedman, aplicado no conjunto das predições de cada modelo avaliado na Figura 5.14, apresentou *estatística* $= 266$ e *p-value* ≈ 0.0, aceitando, assim, a hipótese alternativa de que há diferença nas distribuições das predições de cada modelo. Da mesma forma, no experimento da Figura 5.15, o mesmo teste foi aplicado e apresentou *estatística* $= 386,8$ e *p-value* ≈ 0.0, indicando conclusão similar: aceitação da hipótese alternativa de

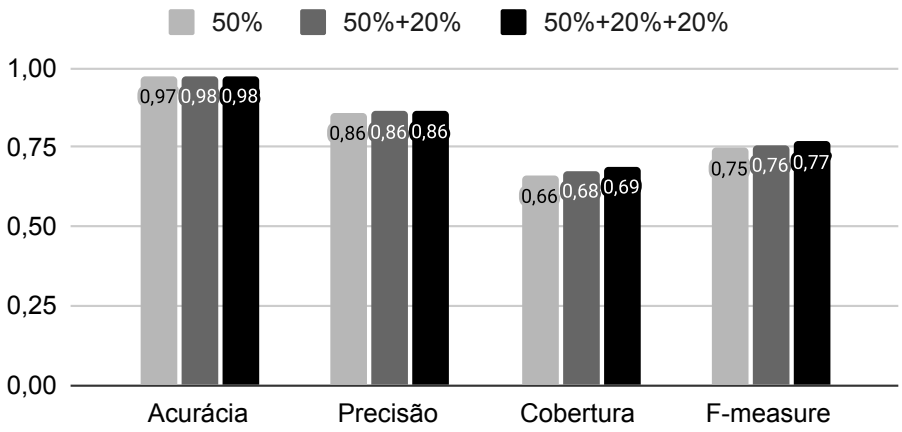

Figura 5.15: Avaliação da aprendizagem incremental (Curitiba). Ajuste de parâmetros com *grid search*.

que há diferença nas distribuições das predições de cada modelo.

5.6.2 Avaliação da Eficiência

Para mensurar a eficiência (QP6), o modelo proposto foi avaliado considerando duas perspectivas: tempo de predição da ocorrência dos aglomerados de ônibus e horizonte de predição. O tempo de execução de todos os experimentos demonstrados nesta seção, que utilizaram os valores de parâmetros obtidos com *grid search*, são apresentados no Apêndice D.

Tempo de Treinamento do Modelo e de Predição dos Aglomerados

Na Tabela 5.2, são exibidos os tempos de treinamento e de predição do modelo proposto, considerando a execução do mesmo com os valores de parâmetros obtidos com a busca manual, que apresentou melhor eficácia. A fase de treinamento requer minutos ou horas para ser concluída, dependendo da quantidade de dados utilizada. O tempo de execução necessário para executar a fase de treinamento de cada cidade considera o modelo que consome mais tempo[5] e que os modelos-base do *ensemble* proposto são executados em

[5]Depende da quantidade de dados utilizada em cada cidade e dos valores dos parâmetros. Com o ajuste de parâmetros com busca manual, o modelo mais custoso foi o RF; enquanto que, com os valores obtidos no *grid search*, o mais custoso foi o Catboost.

paralelo. O tempo de predição (de uma instância por vez) indica a capacidade do modelo proposto de predizer em tempo real (menos de um segundo) as ocorrências de aglomerados de ônibus à medida que novos dados são recebidos.

Tabela 5.2: Tempo de treinamento do *ensemble* e tempo de predição.

Cidade	Tempo de Treinamento (min)	Tempo de Predição (sec)
Curitiba	∼1.399	∼0,39
Cidade A	∼45	∼0,29

Horizonte de Predição

Um dos desafios relacionados à predição dos aglomerados de ônibus é identificar e prevenir um evento com antecedência. Essa ação antecipada ajuda a informar os agentes de trânsito o quanto antes que o aglomerado acontecerá. O modelo proposto foi avaliado com o seguinte horizonte de predição: $1 <= n <= 10$ paradas à frente. Os resultados da aplicação do modelo aos dados da Cidade A e considerando o uso dos parâmetros obtidos com busca manual estão representados na Figura 5.16. Nesta figura, os pontos representam os valores de eficácia, a luminosidade da cor representa as medidas de eficácia, as linhas representam a tendência dos valores e, por fim, o eixo horizontal representa o número de paradas avaliadas.

O experimento mostra que, quanto maior o valor de n, mais difícil é predizer a ocorrência de um aglomerado de ônibus (menor a eficácia). Ainda assim, os resultados indicam que pelo menos $n = 10$ paradas antes da ocorrência, o modelo proposto neste trabalho é capaz de predizer o aglomerado, porque apresentou $F_{measure} \geq 73\%$, mantendo a mesma precisão por pelo menos quatro paradas consecutivas. Apesar das linhas de tendência serem decrescente - como esperado em predições de múltiplos passos, esse experimento mostra que a perda da eficácia do *ensemble* proposto é relativamente pequena após o aumento no número de paradas; em alguns casos, a eficácia permanece a mesma em duas paradas consecutivas.

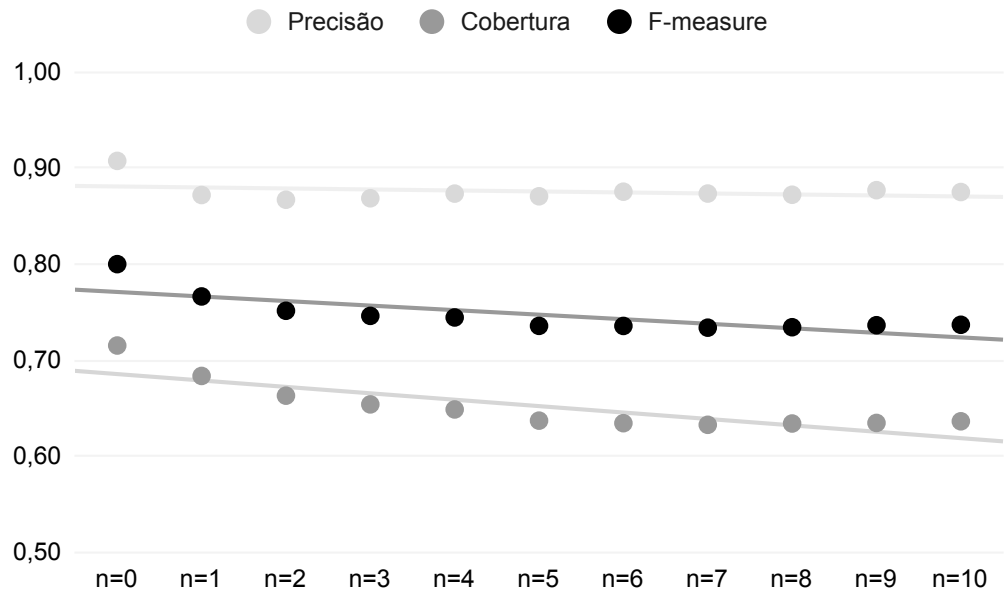

Figura 5.16: Avaliação do *ensemble* com predições para $1 <= n <= 10$ paradas à frente (Cidade A). Ajuste de parâmetros com busca manual.

5.6.3 Comparação com Competidores

Neste trabalho, o modelo proposto[6] foi comparado com quatro outros modelos competidores usados em trabalhos relacionados recentes [59; 52]: RLi, RLo, SVM e RVM. Este último modelo foi avaliado com menos dados devido à limitação de recursos computacionais disponíveis, pois o mesmo consome muita memória. Em decorrência da ausência das bases de dados utilizadas pelos autores, além da descrição limitada dos valores dos parâmetros utilizados, não foi possível comparar o modelo proposto com os outros trabalhos relacionados. Além disso, foram utilizadas as bases de dados empregadas neste trabalho para comparação de todos os modelos.

Nas Figuras 5.17 e 5.18, estão ilustrados os resultados do *ensemble* proposto comparado com os modelos RLi, RLo e SVM[7] com os dados da Cidade A e da cidade de Curitiba, respectivamente, sendo a luminosidade da cor representando os modelos avaliados. Como é possível visualizar, o *ensemble* supera todos os competidores avaliados. Possivelmente isso

[6]Considerando o uso dos parâmetros obtidos com busca manual.

[7]O SVM não foi avaliado com os dados de Curitiba devido ao seu elevado tempo de processamento: aproximadamente 13 dias com os dados da Cidade A (apenas 11 dias de dados).

acontece porque o *ensemble* proposto é composto por algoritmos de aprendizagem baseados em árvore de decisão, que são capazes de se adaptar a relacionamentos complexos e não lineares [29], como a ocorrência de aglomerados de ônibus. Além disso, RLi, RLo e SVM são considerados modelos eficazes em problemas com relações lineares e são mais adequados quando empregados em bases de dados balanceadas.

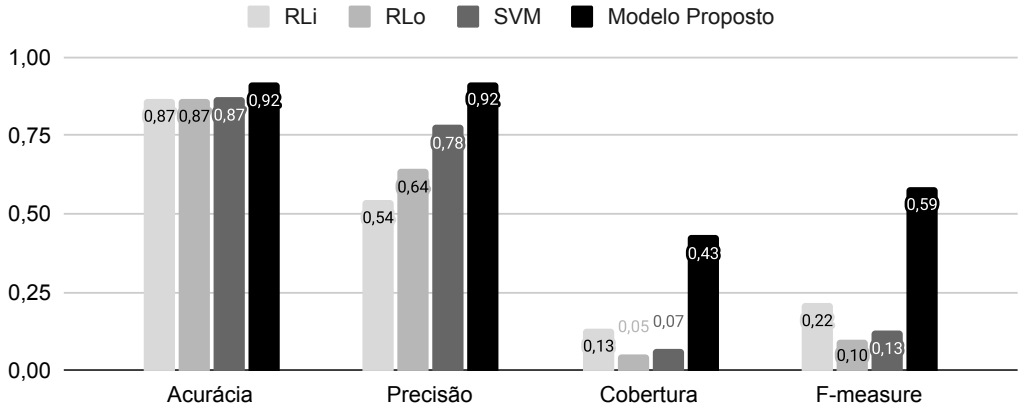

Figura 5.17: Modelo proposto comparado com os competidores, ambos utilizando os dados da Cidade A e ajuste de parâmetros com *grid search*.

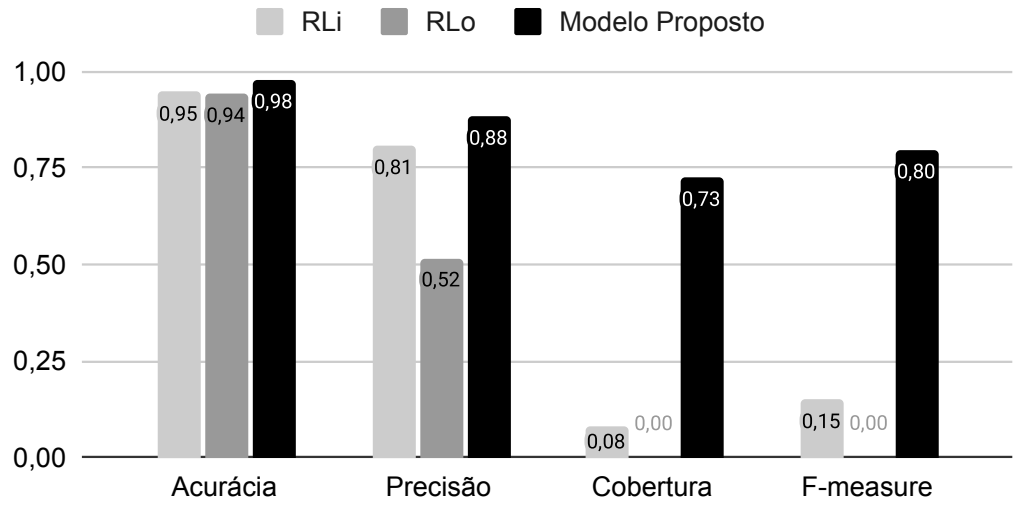

Figura 5.18: Modelo proposto comparado com os competidores ambos utilizando os dados de Curitiba e ajuste de parâmetros com *grid search*.

O teste de Friedman, aplicado no conjunto das predições de cada modelo avaliado na Figura 5.17, apresentou *estatística* $= 818003,9$ e *p-value* ≈ 0.0, aceitando, assim, a hipótese

alternativa de que há diferença nas distribuições das predições de cada modelo. Da mesma forma, no experimento da Figura 5.18, o mesmo teste foi aplicado e apresentou *estatística* $= 2448374,8$ e *p-value* ≈ 0.0, indicando conclusão similar: aceitação da hipótese alternativa de que há diferença nas distribuições das predições de cada modelo.

Na Figura 5.19, são mostrados os resultados do modelo proposto em comparação com o RVM, usando três valores diferentes do parâmetro do kernel: função linear, polinomial e radial (RBF). O eixo horizontal refere-se à quantidade de dados utilizada, o eixo vertical representa os valores de $F_{measure}$ e a luminosidade da cor representa os modelos avaliados.

Assim como ocorreu com os outros modelos competidores, a eficácia do modelo proposto superou a do RVM, exceto para bases de dados menores (5.000 e 15.000 instâncias), nos quais o RVM supera o modelo proposto. Como mostrado na Figura 5.7, isso provavelmente acontece porque o modelo proposto alcança maior eficácia quando a quantidade de dados de treinamento é aumentada. O RVM apresentou melhor resultado com o kernel polinomial, mas de acordo com o aumento da quantidade de dados de treinamento utilizada, houve diminuição na eficácia, apresentando $46\% \leq F_{measure} \leq 51\%$.

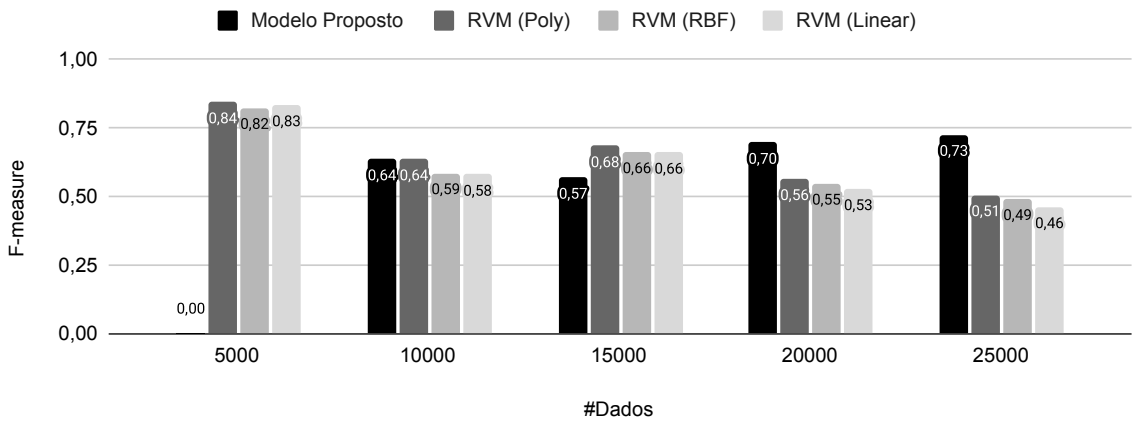

Figura 5.19: Modelo proposto comparado com o modelo competidor RVM, ambos utilizando os dados da Cidade A e ajuste de parâmetros com busca manual.

Em resumo, o *ensemble* proposto superou os modelos competidores (QP7) possivelmente devido aos seguintes motivos: os modelos-base não são sensíveis a dados desequilibrados, o *ensemble* apresenta considerável generalização devido à sua característica intrínseca e, por fim, os modelos-base são capazes de lidar com problemas de relacionamentos não lineares.

5.7 Discussão dos Resultados

A seguir, será apresentado um resumo das conclusões obtidas em cada questão de pesquisa que guiou os experimentos.

QP1: Ao avaliar o modelo proposto com dados de duas cidades brasileiras, os resultados indicaram que o mesmo pode ser usado para predizer aglomerados de ônibus para todas as rotas de ônibus de diferentes cidades. Os experimentos nas duas cidades mostraram $F_{measure} \geq 80\%$ ao utilizar os parâmetros obtidos empiricamente com a busca manual. Os resultados obtidos com os parâmetros do *grid search* apresentaram perda de quase 30% na eficácia com os dados da Cidade A, onde a aplicação do teste de Mann-Whitney indicou diferença significativa nos resultados obtidos. Assim, conclui-se que o ajuste de parâmetros representa uma relevante etapa para a configuração do modelo em cada cidade.

QP2: O ensemble proposto produziu eficácia superior a dois modelos-base utilizados (XGBoost e CatBoost) e produziu eficácia igual a um dos modelo-base (RF), quando avaliado com os valores de parâmetros da busca manual. Quando utilizado os valores de parâmetros do *grid search* nos dados da Cidade A, o desempenho dos modelos-base foi inferior; enquanto que, com os dados de Curitiba, o resultado de eficácia geral do *ensemble* foi similar. O teste de Friedman aplicado a cada cenário, indicou diferença entre as distribuições das predições. Dessa forma, conclui-se que, com a utilização da busca manual no ajuste de parâmetros do modelo com os dados da Cidade A, o *ensemble* pode apresentar resultado similar ou superior aos modelos-base, do mesmo modo que com a utilização do *grid search* para os dados de Curitiba.

QP3: Os resultados dos experimentos indicaram que a eficácia do modelo proposto tende a ter aumento conforme também é aumentada a quantidade de dados utilizada para treinamento do modelo. A exceção ocorreu para a proporção de 75% dos dados, que apresentou resultado similar ao da base completa nos dados da Cidade A, e resultado superior nos dados da cidade de Curitiba. Isto pode ter acontecido devido à proporção maior de ruídos ou desbalanceamento nos 25% dos dados adicionados, ou mesmo à outro fator ainda desconhecido.

QP4: As fontes de dados que apresentaram eficácia maior na predição dos aglomerados de ônibus, quando analisadas individualmente, foram GPS+GTFS, em ambas as técnicas de ajuste de parâmetros empregadas. Ainda assim, o teste de Friedman indicou diferença na

distribuição dos resultados das predições em ambas as cidades. Como citado, para o caso em que houver uma menor quantidade de dados de treinamento disponível, apenas os dados de GPS+GTFS poderia ser suficiente para atingir os mesmos resultados de eficácia, o que economiza o tempo de coleta de dados das demais fontes.

QP5: Com os experimentos conduzidos, observou-se que, dependendo da janela de dados utilizada, é possível manter ou melhorar a eficácia do modelo proposto com a aplicação da aprendizagem incremental. Neste trabalho, a eficácia do modelo aumentou ao utilizar janelas de incrementos superiores a algumas horas de intervalo de atualização, representando o acúmulo de mais dados. O teste de Friedman também indicou diferença significativa nos resultados obtidos. Porém, é possível que outras técnicas de aprendizagem incremental realize a atualização com mais eficácia em um intervalo de tempo menor.

QP6: Segundo os resultados de eficiência, o tempo de execução de uma predição do modelo proposto foi menor que meio segundo, indicando a possibilidade de aplicação do mesmo em tempo real. Além disso, o modelo proposto apresentou desempenho relevante na predição da ocorrência dos aglomerados até dez paradas de ônibus antes, sem prejuízo significativo na eficácia ao obter $73\% \leq F_{measure} \leq 80\%$.

QP7: Não-disponibilização das fontes de dados empregadas e parâmetros dos modelos empregados foram os principais impedimentos encontrados para comparar o *ensemble* apresentado com os modelos já propostos na literatura. Ainda assim, comparando o modelo proposto com alguns modelos competidores, observou-se eficácia superior do *ensemble*, considerando as fontes de dados deste trabalho. A diferença significativa nos resultados deve-se ao fato do *ensemble* representar um conjunto de modelos robustos e não-sensíveis ao desbalanceamento da base de dados, diferente dos modelos avaliados, que tendem a ter boa performance em problemas lineares e com dados de treinamento balanceados.

Como pode-se observar na seção anterior, o resultado de eficácia do modelo proposto, ao utilizar os valores obtidos pelo *grid search*, apresentou-se inferior ao resultado obtido com os valores de parâmetros da busca manual. Isto porque não foi possível avaliar a abordagem *grid search* com a base de dados completa devido ao tempo de treinamento de alguns modelos-base: algumas semanas[8] para o XGBoost, por exemplo; que apresentaria, portanto, os valores de parâmetros ideais para serem conduzidos com a quantidade de dados forne-

[8]Utilizando a configuração de máquina descrita.

cida. Um parâmetro diretamente afetado por esta abordagem é o número de árvores (ou *n_estimators*) que tende a ser maior de acordo com a quantidade de dados de treinamento fornecida. Entretanto, tais avaliações servem para mostrar tendências de resultados de acordo com os diferentes cenários avaliados.

5.8 Ameaças à Validade

O modelo de predição de aglomerados de ônibus considera o *headway* programado (h_k^s) disponível no GTFS para calcular o limiar $\alpha = h_k^s/4$. No entanto, o *headway* programado não está diretamente disponível no GTFS, mas sim, é calculado a partir dos horários de chegada programados dos ônibus em cada parada. Em geral, quando h_k^s não pode ser calculado ou não está disponível, o que geralmente ocorre quando o GTFS está desatualizado, é utilizado um limiar α definido pelo usuário. Embora este trabalho tenha considerado $\alpha = 5$, é recomendada uma personalização cuidadosa do parâmetro para cada cidade.

Todas as bases de dados foram integradas de acordo com medidas de distância entre os pontos de geolocalização de cada base. No entanto, atenção especial deve ser dada à integração dos dados climáticos. Se a estação mais próxima estiver muito longe do ponto analisado, é possível que a precipitação coletada não reflita a situação real do local.

Os experimentos relacionados ao tempo de execução apresentam valores aproximados, visto que tais valores dependem da capacidade de processamento da máquina e da concorrência de processos por recursos. Nesta pesquisa, somente cada experimento avaliado estava sendo executado por máquina.

Por fim, a aprendizagem incremental aplicada nos modelos-base considerou apenas a adição de novas árvores ao modelo-base treinado. No entanto, este tipo de abordagem pode prejudicar a execução das predições em tempo real, caso o tempo de atualização seja custoso. Além disso, intervalos de tempo de atualização $\geq 1h$ podem não recuperar de forma rápida os eventos acontecendo no trânsito.

5.9 Considerações Finais

Neste capítulo, foram apresentadas as avaliações realizadas no modelo proposto. Para realizar os experimentos, sete questões de pesquisa foram definidas e motivaram a execução da avaliação experimental, a qual foi dividida em três cenários de avaliação: eficácia, eficiência e comparação com os modelos competidores Além disso, foram apresentadas as fontes de dados utilizadas nesta pesquisa, o processo de integração dos dados, a discussão dos resultados e, por fim, algumas ameaças à validade dos resultados apresentados.

No capítulo a seguir, são apresentadas as conclusões gerais desta pesquisa e as perspectivas para trabalhos futuros.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste capítulo, serão apresentadas as conclusões gerais deste trabalho na Seção 6.1 e as perspectivas para trabalhos futuros na Seção 6.2.

6.1 Conclusões

Este trabalho propôs um modelo de predição de aglomerados de ônibus usando uma combinação de fontes de dados comumente disponíveis nas cidades: GPS, GTFS, situação climática e situação do trânsito. O modelo é baseado na abordagem *ensemble*, combinando modelos baseados em árvores de decisão (RF, XGBoost e CatBoost) com uma abordagem de votação. O modelo proposto é indicado para aplicação em cidades cujos conjuntos de dados utilizados são desbalanceados, escassos e disponíveis em tempo real. Além disso, um volume de dados considerável ajuda a aumentar a eficácia do modelo de predição de aglomerados de ônibus.

Considerando os experimentos realizados neste trabalho com dados de duas cidades brasileiras, os resultados indicaram algumas características do modelo proposto de predição dos aglomerados: a adaptabilidade do modelo para diferentes cidades e tipos de rotas; o *ensemble* proposto alcança pelo menos o mesmo resultado que o melhor modelo-base aplicado; quanto mais dados, melhor a eficácia do modelo proposto; e as quatro fontes de dados utilizadas apresentam relevância semelhante quando avaliados individualmente, com destaque para as fontes de dados GPS e GTFS. Além disso, a aplicação de uma técnica de aprendizagem incremental para impedir que o modelo se torne obsoleto indicou que, com a adição

de mais incrementos, o desempenho do modelo aumenta discretamente, pois o mesmo será atualizado com dados recentes.

Em resumo, estas predições podem ajudar diretamente os agentes de trânsito, prevendo em tempo real os aglomerados de ônibus pelo menos $n = 10$ paradas antes de sua ocorrência e obtendo $F_{measure} \geq 73\%$. O modelo proposto também obteve melhor desempenho quando comparado com quatro modelos competidores disponibilizados na literatura: Regressão Linear, Regressão Logística, Support Vector Machine e Relevance Vector Machine.

6.2 Trabalhos Futuros

Nesta seção, são apresentadas as perspectivas de extensão do trabalho desenvolvido nesta dissertação, como detalhado a seguir:

- **Criação de uma aplicação para monitoramento de aglomerados de ônibus.** Para os agentes de transporte público, seria ainda mais produtivo possuir uma aplicação completa de monitoramento de aglomerados de ônibus em tempo real, envolvendo desde a análise de dados históricos, indicando padrões da ocorrência dos eventos e a exibição de estatísticas, a própria predição considerando os dados em tempo real, até as sugestões de ações preventivas e corretivas a serem tomadas;

- **Aplicação de outras técnicas de *ensemble*.** Outras técnicas de *ensemble* para combinar os modelos-base ainda podem ser avaliadas, como a *bagging*. Além disso, é necessário avaliar o desempenho da adição de outros modelos-base ao *ensemble* proposto, por exemplo, o LightGBM;

- **Aplicação de outras técnicas de aprendizagem incremental.** Para reduzir o intervalo de atualização do modelo, outras técnicas de aprendizagem incremental podem ser avaliadas, a exemplo de uma que atualize as árvores já criadas na fase de treinamento, em vez de criar e adicionar novas árvores, conforme foi utilizado neste trabalho;

- **Avaliação da predição para todas as paradas da rota.** A fim de garantir o maior horizonte de predição possível para o modelo proposto, é necessário avaliar a predição dos aglomerados de ônibus para todas as paradas em cada rota. Para isso, pode-se

ajustar os dados de treinamento, adicionando o rótulo de todas as paradas seguintes à cada instância analisada e aplicar o modelo aqui proposto.

- **Aplicação de técnicas que adicionam interpretabilidade às predições.** Visando facilitar a tomada de decisão dos agentes de transporte público, pode-se adicionar interpretabilidade às predições da ocorrência dos aglomerados de ônibus, ou seja, indicar quais valores de variáveis estão influenciando no resultado da predição. As árvores de decisão já são modelos interpretáveis, porém a utilização de muitas árvores, como no *ensemble*, dificulta a interpretação dos resultados. Para superar este desafio, pode-se aplicar abordagens para explicar os resultados destes modelos, como o LIME (*Local Interpretable Model-Agnostic Explanations*) [47], cuja intuição é variar os dados de entrada e avaliar como as predições mudam. Entretanto, será necessário avaliar o tempo de execução destas abordagens para verificar se as predições com interpretabilidade ainda serão realizadas em tempo real.

Bibliografia

[1] Matthias Andres. A predictive-control framework to eliminate bus bunching. Master's thesis, University of Kaiserslautern, Erwin-Schrödinger-Straße 52, 67663 Kaiserslautern, Germany, 2016.

[2] Matthias Andres and Rahul Nair. A predictive-control framework to address bus bunching. *Transportation Research Part B: Methodological*, 104:123–148, 2017.

[3] ANTP. Entidades lançam 5 programas para o transporte público e a mobilidade sustentável no brasil. Disponível em: http://www.antp.org.br/noticias/destaques/entidades-lancam-5-programas-para-o-transporte-publico-e-a-mobilidade-sustentavel-no-brasil.html, 2018. Acesso em 29 de dezembro de 2018.

[4] Jacqueline Arriagada, Antonio Gschwender, Marcela A Munizaga, and Martin Trépanier. Modeling bus bunching using massive location and fare collection data. *Journal of Intelligent Transportation Systems*, 23(4):332–344, 2019.

[5] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.

[6] Aloïs Bissuel. Hyper-parameter optimization algorithms: a short review. Disponível em: https://medium.com/criteo-labs/hyper-parameter-optimization-algorithms-2fe447525903, 2019. Acesso em 14 de janeiro de 2021.

[7] Tarciso Braz, Matheus Maciel, Demetrio Gomes Mestre, Nazareno Andrade, Carlos Eduardo Pires, Andreza Raquel Queiroz, and Veruska Borges Santos. Estimating inefficiency in bus trip choices from a user perspective with schedule, positioning, and ticketing data. *IEEE Transactions on Intelligent Transportation Systems*, 19(11):3630–3641, 2018.

[8] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. Classification and regression trees. belmont, ca: Wadsworth. *International Group*, 432:151–166, 1984.

[9] Jason Brownlee. 4 strategies for multi-step time series forecasting. Disponível em: https://machinelearningmastery.com/multi-step-time-series-forecasting/, 2019. Acesso em 19 de julho de 2020.

[10] Jason Brownleel. How to calculate nonparametric statistical hypothesis tests in python. Disponível em: https://machinelearningmastery.com/nonparametric-statistical-significance-tests-in-python/, 2018. Acesso em 16 de janeiro de 2021.

[11] Francesco Calabrese, Rahul Nair, and Fabio Pinelli. Real-time prediction and correction of scheduled service bunching, January 2 2018. US Patent 9,858,542.

[12] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[13] Haibin Cheng, Pang-Ning Tan, Jing Gao, and Jerry Scripps. Multistep-ahead time series prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 765–774. Springer, 2006.

[14] Mario Cools, Elke Moons, and Geert Wets. Assessing the impact of weather on traffic intensity. *Weather, Climate, and Society*, 2(1):60–68, 2010.

[15] Gregory W Corder and Dale I Foreman. *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014.

[16] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. In *arXiv preprint arXiv:1810.11363*, 2018.

[17] Marcos Bicalho dos Santos. Financiamento do custeio do transporte público coletivo urbano. Disponível em: https://www.ntu.org.br/novo/upload/Publicacao/Pub63668720 3994198126.pdf, 2018. Acesso em 30 de dezembro de 2018.

[18] John Dudovskiy. Snowball sampling. Disponível em: https://research-methodology.net/sampling-in-primary-data-collection/snowball-sampling/. Acesso em 18 de janeiro de 2021.

[19] Wei Feng and Miguel Figliozzi. Empirical findings of bus bunching distributions and attributes using archived avl/apc bus data. In *ICCTP 2011: Towards Sustainable Transportation Systems*, pages 4330–4341. 2011.

[20] Maria Teresa Francoso and Wilson Aparecido Sedano Filho. Influência das chuvas no uso de transporte público: Um estudo baseado no município de são paulo. *Gestão do Ambiente Construído*, pages 48–59.

[21] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[22] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[23] Google. View places, traffic, terrain, biking, and transit. Disponível em: https://support.google.com/maps/answer/3092439?co=GENIE.Platform%3DDesktophl=en: :text=Traffic%20colorstext=Green%3A%20No%20traffic%20delays.,of%20traffic%20on%20the%20road., 2020. Acesso em 27 de julho de 2020.

[24] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[25] Markus Hofmann and Margaret O'Mahony. The impact of adverse weather conditions on urban bus performance measures. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pages 84–89. IEEE, 2005.

[26] Syed Saqib Ali Kazmi, Mehreen Ahmed, Rafia Mumtaz, and Zahid Anwar. Spatio-temporal clustering and analysis of road accident hotspots by exploiting gis technology and kernel density estimation. *The Computer Journal*, 00, 2020.

[27] John D Kelleher, Brian Mac Namee, and Aoife D'arcy. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.

[28] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[29] Christoph Kern, Thomas Klausch, and Frauke Kreuter. Tree-based machine learning methods for survey research. In *Survey Research Methods*, volume 13, pages 73–93, 2019.

[30] M Venkateswararao Koppisetti, Veeraruna Kavitha, and Urtzi Ayesta. Bus schedule for optimal bus bunching and waiting times. In *Communication Systems & Networks (COMSNETS), 2018 10th International Conference on*, pages 607–612. IEEE, 2018.

[31] Vijay Kotu and Bala Deshpande. *Predictive Analytics and Data Mining*. Morgan Kaufmann, Boston, MA, 2015.

[32] Vijay Kotu and Bala Deshpande. *Data Science (Second Edition)*. Morgan Kaufmann, Boston, MA, 2019.

[33] Viktor Losing, Barbara Hammer, and Heiko Wersing. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274, 2018.

[34] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[35] Demetrio Gomes Mestre. *Leveraging the entity matching performance through adaptive indexing and efficient parallelization*. PhD thesis, Universidade Federal de Campina Grande, R. Aprígio Veloso, 882 - Universitário, Campina Grande - PB, 58428-830, 2018.

[36] Luís Moreira-Matias, Oded Cats, João Gama, João Mendes-Moreira, and Jorge Freire de Sousa. An online learning approach to eliminate bus bunching in real-time. *Applied Soft Computing*, 47:460–482, 2016.

[37] Luís Moreira-Matias, João Gama, João Mendes-Moreira, and Jorge Freire de Sousa. An incremental probabilistic model to predict bus bunching in real-time. In *International Symposium on Intelligent Data Analysis*, pages 227–238. Springer, 2014.

[38] Rahul Nair, Eric Bouillet, Yiannis Gkoufas, Olivier Verscheure, Magda Mourad, Farzin Yashar, Rosie Perez, Joel Perez, Gerald Bryant, and Miami Dade Transit. Data as a resource: real-time predictive analytics for bus bunching. In *Proceedings of the Annual Meeting of the Transportation Research Board, Washington, DC*, 2014.

[39] Taewoo Nam and Theresa A Pardo. Smart city as urban innovation: Focusing on management, policy, and context. In *Proceedings of the 5th international conference on theory and practice of electronic governance*, pages 185–194, 2011.

[40] Gordon Frank Newell and Renfrey Burnard Potts. Maintaining a bus schedule. In *Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne*, volume 2, 1964.

[41] NTU. Ônibus urbano perde três milhões de passageiros por dia. Disponível em: https://www.ntu.org.br/novo/NoticiaCompleta.aspx?idArea=10&idNoticia=850, 2017. Acesso em 29 de dezembro de 2018.

[42] Zhenchao Ouyang, Jianwei Niu, Yu Liu, and Xue Liu. An ensemble learning-based vehicle steering detector using smartphones. *IEEE Transactions on Intelligent Transportation Systems*, 21:1964–1975, 2019.

[43] Charlie Parker, Sam Scott, and Alistair Geddes. Snowball sampling. *SAGE Research Methods Foundations*, pages 1–13, 2019.

[44] Antonio Rafael Sabino Parmezan, Gustavo Enrique de Almeida Prado Alves Batista, et al. Descrição de modelos estatísticos e de aprendizado de máquina para predição de séries temporais. *São Carlos, SP, Brasil.*, 2016.

[45] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[46] Soroush Rashidi, Prakash Ranjitkar, Orosz Csaba, and Andy Hooper. Using automatic vehicle location data to model and identify determinants of bus bunching. *Transportation research procedia*, 25:1444–1456, 2017.

[47] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[48] Irum Sanaullah, Mohammed Quddus, and Marcus Enoch. Developing travel time estimation methods using sparse gps data. *Journal of Intelligent Transportation Systems*, 20(6):532–544, 2016.

[49] Jan-Dirk Schmöcker, Wenzhe Sun, Achille Fonzone, and Ronghui Liu. Bus bunching along a corridor served by two lines. *Transportation Research Part B: Methodological*, 93:300–317, 2016.

[50] Leão Serva. Após crescerem dez vezes em 70 anos, cidades têm de melhorar mobilidade. Disponível em: http://temas.folha.uol.com.br/e-agora-brasil-transporte-urbano/falta-de-planejamento/apos-crescerem-dez-vezes-em-70-anos-cidades-tem-de-melhorar-mobilidade.shtml, 2018. Acesso em 29 de dezembro de 2018.

[51] Wenzhe Sun. *Bus Bunching Prediction and Transit Route Demand Estimation Using Automatic Vehicle Location Data*. PhD thesis, Kyoto University, Yoshidahonmachi, Sakyo Ward, Kyoto, 606-8501, Japan, 2020.

[52] Wenzhe Sun, Jan-Dirk Schmöcker, and Toshiyuki Nakamura. On the tradeoff between sensitivity and specificity in bus bunching prediction. pages 1–17. Taylor & Francis, 2020.

[53] Yanshuo Sun, Jungang Shi, and Paul M Schonfeld. Identifying passenger flow characteristics and evaluating travel time reliability by visualizing afc data: a case study of shanghai metro. *Public Transport*, 8(3):341–363, 2016.

[54] Alexandre SG Vianna, Michael O Cruz, Luciano Barbosa, and Kiev Gama. Análise do impacto de chuvas na velocidade média do transporte público coletivo de ônibus em recife. In *Anais do I Workshop Brasileiro de Cidades Inteligentes*. SBC, 2018.

[55] Waze. Use traffic view. Disponível em: https://support.google.com/waze/partners/answer/7246755?hl=en, 2020. Acesso em 27 de julho de 2020.

[56] Ian H Witten, Eibe Frank, and Mark A Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Cambridge, MA, 2005.

[57] Junjian Yang, Hang Zhou, Xuewu Chen, and Long Cheng. Applying the support vector machine to predicting headway-based bus bunching. In *CICTP 2019*, pages 1542–1553. 2019.

[58] Haiyang Yu, Dongwei Chen, Zhihai Wu, Xiaolei Ma, and Yunpeng Wang. Headway-based bus bunching prediction using transit smart card data. *Transportation Research Part C: Emerging Technologies*, 72:45–59, 2016.

[59] Haiyang Yu, Zhihai Wu, Dongwei Chen, and Xiaolei Ma. Probabilistic prediction of bus headway using relevance vector machine regression. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1772–1781, 2017.

[60] Jinhua Zhao, Adam Rahbee, and Nigel HM Wilson. Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. *Computer-Aided Civil and Infrastructure Engineering*, 22(5):376–387, 2007.

Apêndice A

Exemplos de Arquivos do GTFS

Neste apêndice, são exibidas amostras de dados dos arquivos de GTFS utilizados neste trabalho. Os dados são referentes à cidade de Curitiba - Brasil.

route_id	agency_id	route_short_name	route_long_name	route_desc	route_type	route_url	route_color	route_text_color
1	1	532	JD. PARANAENSE		3		F98700	0
6	1	632	QUARTEL GENERAL		3		F98700	0
7	1	657	XAXIM / CAPÃO RASO		3		F98700	0
12	1	666	NOVO MUNDO		3		EFDC07	0
16	1	169	JD. KOSMOS		3		EFDC07	0
17	1	902	STA. FELICIDADE L.D.		3		FF0000	0
20	1	171	PRIMAVERA		3		EFDC07	0
21	1	534	PARIGOT DE SOUZA		3		F98700	0
22	1	533	E. VERÍSSIMO / PANTANAL		3		F98700	0
25	1	622	RONDON		3		F98700	0
27	1	654	CAMPO ALEGRE		3		F98700	0
29	1	331	MERCÚRIO		3		F98700	0
30	1	212	SOLAR		3		F98700	0
31	1	213	SÃO JOÃO		3		F98700	0
32	1	225	BOA VISTA / BARREIRINHA		3		F98700	0
33	1	224	CASSIOPÉIA		3		F98700	0
34	1	226	ABAETÉ		3		F98700	0
35	1	242	V. LEONICE		3		F98700	0
39	1	822	GABINETO		3		F98700	0

Figura A.1: Amostra de dados do arquivo *routes* do GTFS.

route_id	service_id	trip_id	trip_headsign	trip_short_name	direction_id	block_id	shape_id
34	1	4380739	Terminal Boa Vista		1		1800
34	1	4380740	Abaeté		0		1799
34	1	4380741	Terminal Boa Vista		1		1800
34	1	4380742	Abaeté		0		1799
34	1	4380743	Terminal Boa Vista		1		1800
34	1	4380744	Abaeté		0		1799
34	1	4380745	Terminal Boa Vista		1		1800
34	1	4380746	Abaeté		0		1799
34	1	4380747	Terminal Boa Vista		1		1800
34	1	4380748	Abaeté		0		1799
34	1	4380749	Terminal Boa Vista		1		1800
34	1	4380750	Abaeté		0		1799
34	1	4380751	Terminal Boa Vista		1		1800
34	1	4380752	Abaeté		0		1799
34	1	4380753	Terminal Boa Vista		1		1800
34	1	4380754	Abaeté		0		1799
34	1	4380755	Terminal Boa Vista		1		1800
34	1	4380756	Abaeté		0		1799
34	1	4380757	Terminal Boa Vista		1		1800

Figura A.2: Amostra de dados do arquivo *trips* do GTFS.

trip_id	arrival_time	departure_time	stop_id	stop_sequence	stop_headsign	pickup_type	drop_off_type
4380739	05:00:00	05:00:00	1899	1		0	0
4380739	05:00:48	05:00:48	27220	2		0	0
4380739	05:01:44	05:01:44	27221	3		0	0
4380739	05:02:30	05:02:30	27222	4		0	0
4380739	05:03:32	05:03:32	34486	5		0	0
4380739	05:04:14	05:04:14	27223	6		0	0
4380739	05:05:13	05:05:13	27224	7		0	0
4380739	05:06:13	05:06:13	27179	8		0	0
4380739	05:07:10	05:07:10	27226	9		0	0
4380739	05:07:56	05:07:56	31888	10		0	0
4380739	05:08:55	05:08:55	27227	11		0	0
4380739	05:09:38	05:09:38	27228	12		0	0
4380739	05:10:42	05:10:42	31690	13		0	0
4380739	05:11:17	05:11:17	27229	14		0	0
4380739	05:12:01	05:12:01	27173	15		0	0
4380739	05:13:01	05:13:01	27213	16		0	0
4380739	05:13:43	05:13:43	27232	17		0	0
4380739	05:14:53	05:14:53	27163	18		0	0
4380739	05:15:53	05:15:53	27187	19		0	0

Figura A.3: Amostra de dados do arquivo *stop_times* do GTFS.

stop_id	stop_code	stop_name	stop_desc	stop_lat	stop_lon
70	104505	Terminal Campina do Siqueira - 303 - Centenário / Campo Comprido	Terminal Campina do Siqueira - Campo Comprido	-25.435723602407	-49.30699836494
270	104905	Terminal Carmo - 030 - Interbairros III	Terminal Carmo 030 - Interbairros III (Sentido Oficinas)	-25.501341193047	-49.23759683498
276	105606	Terminal Oficinas - 030 - Interbairros III	Terminal Oficinas 030 - Interbairros III (Sentido Bairro Alto)	-25.451549913966	-49.214917373611
299	105603	Terminal Oficinas - 030 - Interbairros III	Terminal Oficinas 030 - Interbairros III (Sentido Carmo)	-25.451665262751	-49.215085830689
308	104907	Terminal Carmo - 030 - Interbairros III	Terminal Carmo 030 - Interbairros III (Sentido Capão Raso)	-25.501310905969	-49.237825154596
568	190836	R. Dep. José Hoffmann, 80 - Vista Alegre	150 - C. Música / V.Alegre (Ponto Final)	-25.4086089	-49.29986
581	110312	Praça Santos Andrade - 150 - C. da Música / Vista Alegre	Praça Santos Andrade 150 - C.Música / V. Alegre .	-25.428209687138	-49.265846178539
597	190896	R. Eng. Agro. Lauro Klas, 106 - Pilarzinho	160 - Jd. Mercês / Guanabara (Ponto Final)	-25.398409680381	-49.293254561972
616	150689	Rua Rio de Janeiro, 1293 - Água Verde	Ponto Final 160 - Jd. Mercês / Guanabara (Sentido Jd. Mercês)	-25.462635199023	-49.277920309193
662	190600	Rua São Francisco Xavier, 132 - Pilarzinho	166 - Vila Nori (Ponto Final) 167 - Fredolin Wolf (Sentido Nestor de Castro)	-25.389211409326	-49.302757242328
690	120607	Rua José Ribeiro de Cristo, 1-119 - Pilarzinho	Ponto de Parada 170 - Bracatinga (Sentido Bracatinga) 188 - Madrugueiro Pilarzinho / Uberaba (Sentido Pilarzinho)	-25.382745007551	-49.293668590178
728	190638	Rua Tomaz Tessari, 95 - Lamenha Pequena	168 - Raposo Tavares (Ponto Final)	-25.357505925976	-49.333239863824
775	120350	R. Jorn. Geraldo Russe, 1-79 - Pilarzinho	169 - Jd. Kosmos (Ponto Final)	-25.391328615541	-49.294544653373
806	120317	Rua Otalino Amado de Souza, 1 - Pilarzinho	170 - Bracatinga (Ponto Final) 188 - Madrugueiro Pilarzinho / Uberaba (Sentido Pilarzinho)	-25.381670814559	-49.300806196924
842	120342	Rua Domingos Antônio Moro, 1267 - Pilarzinho	171 - Primavera (Ponto Final)	-25.378955722683	-49.292597304299
867	120948	R. Rolfe Mertens, 42 - Pilarzinho	175 - Bom Retiro / PUC (Ponto Final -Bom Retiro)	-25.399735159139	-49.278406019643

Figura A.4: Amostra de dados do arquivo *stops* do GTFS.

shape_id	shape_pt_lat	shape_pt_lon	shape_pt_sequence	shape_dist_traveled
1708	-25.416432732579345	-49.268802216214	7166054	0
1708	-25.41647121277954	-49.26860454212067	7166055	20339
1708	-25.416512115585526	-49.26842497293825	7166056	38965
1708	-25.416543599787797	-49.26842834076837	7166057	42469
1708	-25.416572661387978	-49.26842097976231	7166058	45772
1708	-25.416604816916784	-49.26840751216139	7166059	49583
1708	-25.416626070744645	-49.26838868014249	7166060	52605
1708	-25.41664314111133	-49.26836912101544	7166061	55334
1708	-25.416654267153575	-49.26835011932894	7166062	57609
1708	-25.41666514922686	-49.268321061388065	7166063	60771
1708	-25.41667179175138	-49.268296697313076	7166064	63330
1708	-25.41667175250458	-49.268269426835104	7166065	66074
1708	-25.416671068360618	-49.26824170796931	7166066	68864
1708	-25.416661824427674	-49.268203430976996	7166067	72849
1708	-25.416647512056958	-49.26818060444339	7166068	75640
1708	-25.416624306148965	-49.268153768474804	7166069	79368
1708	-25.41660412849299	-49.268141684655745	7166070	81912
1708	-25.41657671505156	-49.26812553588786	7166071	85356
1708	-25.416613568229756	-49.2678836516057	7166072	110031

Figura A.5: Amostra de dados do arquivo *shapes* do GTFS.

service_id	monday	tuesday	wednesday	thursday	friday	saturday	sunday	start_date	end_date
1	1	1	1	1	1	0	0	20130401	20190630
2	0	0	0	0	0	1	0	20130401	20190630
3	0	0	0	0	0	0	1	20130401	20190630
4	0	0	1	0	0	0	0	20130401	20190630

Figura A.6: Amostra de dados do arquivo *calendar* do GTFS.

Apêndice B

Variáveis Utilizadas

Neste apêndice, serão descritas as 119 variáveis[1] de cada conjunto de dados usado nos experimentos após a integração de dados e a engenharia de variáveis.

Variáveis de GPS

- *route*: representa o código da rota do ônibus, por exemplo, 242.
- *tripNum*: número sequencial da viagem de ônibus[2], por exemplo, 1.
- *busCode*: identificador do ônibus, por exemplo, $BA001$.
- *gpsPointId*: identificador do ponto de geolocalização do GPS, por exemplo, 574.
- *gpsLat*: latitude do ponto de GPS, por exemplo, -25.367081.
- *gpsLon*: longitude do ponto de GPS, por exemplo, -49.259845.
- *distanceToShapePoint*: distância (em metros) entre o ponto do *shape* e o ponto de GPS, por exemplo, 28.726088.
- *problem*: status da viagem identificado pelo algoritmo BULMA. Em alguns casos, as viagens não são identificadas corretamente devido à escassez de pontos de GPS, por exemplo, $NO\_PROBLEM$.

[1] A cidade A possui apenas 111 colunas porque as variáveis com valores ausentes no intervalo de tempo dos dados considerados foram removidas.

[2] Viagem de ônibus é o caminho percorrido pelo ônibus de um ponto inicial predefinido até um ponto final. É representado por um conjunto de pontos de GPS ordenados.

- *headway*: diferença do horário de chegada (em minutos) na parada de ônibus entre dois ônibus consecutivos da mesma rota, por exemplo, 21.
- *busBunching*: ocorrência (ou não) de aglomerado de ônibus, por exemplo, $False$.
- *GPShour*: hora em que os dados foram coletados. É extraído da data do GPS, por exemplo, 6.
- *DAY(gps_datetime)*: dia em que os dados foram coletados. É extraído da data do GPS, por exemplo, 20.
- *YEAR(gps_datetime)*: ano em que os dados foram coletados. É extraído da data do GPS, por exemplo, 2019.
- *MONTH(gps_datetime)*: número do mês em que os dados foram coletados. É extraído da data do GPS. Por exemplo, janeiro = 1, ... , dezembro = 12.
- *WEEKDAY(gps_datetime)*: número do dia da semana em que os dados foram coletados. É extraído da data do GPS. Por exemplo, segunda-feira = 0, ... , domingo = 6.

Variáveis do GTFS

- *shapeId*: representa o identificador do *shape* do ônibus, ou seja, o caminho predefinido do ônibus, por exemplo, 1815. Cada *shape* é composto por pontos de geolocalização (latitude e longitude) e sua sequência.
- *routeFrequency*: frequência da rota, com base no *headway* mediano da cidade (h_median), ou seja, $high\_frequency$ se a média do *headway* da rota for menor que a mediana do *headway* da cidade; $low\_frequency$, caso contrário.
- *shapeSequence*: sequência do ponto do *shape*. É um número sequencial, por exemplo, 7136773.
- *shapeLat*: latitude do ponto do *shape*, por exemplo, -25.36733924283432.
- *shapeLon*: longitude do ponto da *shape*, por exemplo, -49.259852836954.

- *distanceTraveledShape*: comprimento/distância (em metros) do ponto inicial do *shape* até o ponto atual, por exemplo, 1269927.
- *stopPointId*: identificador do ponto da parada de ônibus, por exemplo, 31114.
- *headwayThreshold*: limiar baseado no *headway* para definir aglomerado de ônibus. É derivado do *headway* programado encontrado nos dados do GTFS (quando definido) ou em um limiar definido pelo usuário, por exemplo, 5.

Variáveis climáticas

- *precipitation*: representa a precipitação (em milímetros), por exemplo, $0,4$.
- *precipitationStationDistance*: distância (em metros) entre a estação meteorológica e o ponto do *shape*, por exemplo, $1595,623046875$.
- *DAY(precipitationTime)*: dia em que os dados foram coletados. É extraído da data da precipitação, por exemplo, 20.
- *YEAR(precipitationTime)*: ano em que os dados foram coletados. É extraído da data da precipitação, por exemplo, 2019.
- *MONTH(precipitationTime)*: número do mês em que os dados foram coletados. É extraído da data da precipitação. Por exemplo, janeiro = 1, ... , dezembro = 12.
- *WEEKDAY(precipitationTime)*: número do dia da semana em que os dados foram coletados. É extraído da data da precipitação. Por exemplo, segunda-feira = 0, ... , domingo = 6.

Variáveis de trânsito

- *alertSubtype*: representa o subtipo de alerta de trânsito, ou seja, a subcategoria do alerta, por exemplo, $NORMAL$.
- *alertType*: tipo de alerta de trânsito, ou seja, a categoria do alerta, por exemplo, $NORMAL$.

- *alertRoadType*: número representando o tipo da via, por exemplo, 2 (= via principal).
- *alertConfidence*: confiança $[-1, 5]$ no alerta com base nas reações (e.g. curtidas) de outros usuários, por exemplo, 5.
- *alertNComments*: número total de comentários nesse alerta publicado, por exemplo, 0.
- *alertNImages*: número total de imagens publicadas nesse alerta, por exemplo, 0.
- *alertNThumbsUp*: número total de curtidas nesse alerta publicado, por exemplo, 0.
- *alertReliability*: confiança $[0, 10]$ no alerta com base na entrada do usuário (e.g. curtidas) e classificação da publicação, por exemplo, 10.
- *alertReportMood*: nível de entusiasmo do alerta, por exemplo, 34.
- *alertReportRating*: classificação do usuário $[1, 6]$ (6 = usuário de classificação alta). Por exemplo, 5.0.
- *alertSpeed*: velocidade média atual em vias engarrafadas (em metros por segundo). Por exemplo, $0, 0$.
- *alertLatitude*: latitude de um ponto da via, por exemplo, $-25, 367081$.
- *alertLongitude*: longitude de um ponto da via, por exemplo, -49.259845.
- *alertDistanceToClosestShapePoint*: distância (em metros) entre o alerta e o ponto do *shape*, por exemplo, $28, 726088$.
- *alertIsJamUnifiedAlert*: define se o alerta de congestionamento está unificado ou não, por exemplo, $False$.
- *alertInScale*: define se o alerta está em escala, por exemplo, $False$.
- *jamBlockType*: tipo da via do congestionamento, por exemplo, $NORMAL$.
- *jamDelay*: atraso do congestionamento (em segundos) comparado à velocidade de fluxo livre, por exemplo, 0.

- *jamLength*: comprimento do congestionamento (em metros), por exemplo, 0.
- *jamLevel*: nível de congestionamento do tráfego $[0, 5]$ (0 = fluxo livre, 5 = bloqueado). Por exemplo, 0.
- *jamSeverity*: severidade do congestionamento $[0, 5]$ em comparação com a velocidade histórica (5 = mais grave). Por exemplo, 0.
- *jamSpeedKM*: velocidade média do congestionamento (em quilômetros por hora). Por exemplo, $5,47$.
- *DAY(alertDateTime)*: dia extraído da data do alerta de trânsito, por exemplo, 20.
- *DAY(jamUpdateDateTime)*: dia da última atualização do alerta de congestionamento. É extraído da data do congestionamento, por exemplo, 20.
- *DAY(jamExpirationDateTime)*: último dia de validade do alerta de congestionamento, por exemplo, 20.
- *YEAR(alertDateTime)*: ano em que o dado do alerta foi coletado. É extraído da data do alerta de trânsito, por exemplo, 2019.
- *YEAR(jamUpdateDateTime)*: ano da última atualização do alerta de congestionamento. É extraído da data do congestionamento, por exemplo, 2019.
- *YEAR(jamExpirationDateTime)*: ano de validade do alerta de congestionamento, por exemplo, 2019.
- *MONTH(alertDateTime)*: número do mês em que o dado foi coletado. É extraído da data do alerta de trânsito. Por exemplo, janeiro = 1, ... , dezembro = 12.
- *MONTH(jamUpdateDateTime)*: mês da última atualização do alerta de congestionamento. É extraído da data do congestionamento. Por exemplo, janeiro = 1, ... , dezembro = 12.
- *MONTH(jamExpirationDateTime)*: número do mês de validade do alerta de congestionamento. Por exemplo, janeiro = 1, ... , dezembro = 12.

- *WEEKDAY(alertDateTime)*: número do dia da semana extraído da data do alerta de trânsito. Por exemplo, segunda-feira = 0, ... , domingo = 6.

- *WEEKDAY(jamUpdateDateTime)*: dia da semana da última atualização de alerta de congestionamento. É extraído da data do congestionamento. Por exemplo, segunda-feira = 0, ... , domingo = 6.

- *WEEKDAY(jamExpirationDateTime)*: número do dia da semana de validade do alerta do congestionamento. Por exemplo, segunda-feira = 0, ..., domingo = 6.

Em cada linha do conjunto de dados, todas essas variáveis (exceto *route, shapeId, routeFrequency, headway, headwayThreshold, busBunching* e *GPShour*) são adicionadas ao ônibus consecutivo da mesma rota, ou seja, cada linha representa os dados de um ônibus e o seu consecutivo.

Apêndice C

Parâmetros dos Modelos

Neste Apêndice, são exibidos os valores dos parâmetros utilizados nos modelos-base do *ensemble* proposto para realização dos experimentos.

C.1 Valores dos Parâmetros após Busca Manual

Nas Tabelas C.1, C.2 e C.3 são mostrados os valores dos parâmetros de cada modelo-base. Tais valores foram encontrados após o ajuste dos hiper-parâmetros dos modelos considerando a técnica de busca manual e empírica. Os valores padrão de cada API foram utilizados para os demais parâmetros.

Tabela C.1: Valores dos parâmetros do modelo *Random Forest* para as duas cidades.

Parâmetros do *Random Forest*	Descrição	Curitiba	Cidade A
n_estimators	Número de árvores a serem consideradas.	150	100
min_samples_split	Número mínimo de amostras de dados necessárias para dividir um nó interno.	5	5
max_features	Número de variáveis a serem consideradas ao procurar a melhor divisão.	0.8	0.8

Tabela C.2: Valores dos parâmetros do modelo XGBoost para as duas cidades.

Parâmetros do XGBoost	Descrição	Valores
n_estimators	Número de árvores a serem consideradas.	120
learning_rate	A taxa de aprendizado usada para reduzir a etapa do gradiente.	0.1
max_depth	Profundidade máxima de uma árvore.	50
min_child_weight	Soma mínima do peso da instância necessária em um nó folha. Se a etapa de partição na árvore resultar em um nó folha com a soma do peso da instância menor que min_child_weight, o processo de criação desistirá de particionar mais.	1
gamma	Redução de perda mínima necessária para fazer uma partição adicional em um nó folha da árvore.	0
subsample	Proporção da subamostra das instâncias dos dados de treinamento a serem considerados a cada iteração.	0.8
colsample_bytree	Proporção de subamostra de colunas ao construir cada árvore. A subamostragem ocorre uma vez para cada árvore construída.	0.8
scale_pos_weight	Parâmetro para controlar o equilíbrio de pesos positivos e negativos, útil para classes desbalanceadas.	1

C.2 Valores dos Parâmetros após Grid Search

Nas Tabelas C.4, C.5 e C.6 são mostrados os valores dos parâmetros de cada modelo-base. Tais valores foram encontrados após o ajuste dos hiper-parâmetros dos modelos considerando a técnica de *grid search* aplicada em uma base menor com 5% dos dados de cada cidade. Os valores padrão de cada API foram utilizados para os demais parâmetros.

Um ponto importante a se observar é que boa parte dos valores dos parâmetros sofreram alteração, quando comparados os tipos de técnicas de ajuste de hiper-parâmetros aplicadas. Apesar dos valores da primeira técnica terem sido encontrados por meio de busca empí-

Tabela C.3: Valores dos parâmetros do modelo CatBoost para as duas cidades.

Parâmetros do CatBoost	Descrição	Valores
learning_rate	A taxa de aprendizado usada para reduzir a etapa do gradiente.	0.9
depth	Profundidade máxima da árvore.	8
iterations	O número máximo de árvores que podem ser construídas.	10000
l2_leaf_reg	Coeficiente no termo de regularização L2 da função de custo. Qualquer valor positivo é permitido.	5

Tabela C.4: Valores dos parâmetros do modelo *Random Forest* para as duas cidades.

Parâmetros do *Random Forest*	Descrição	Curitiba	Cidade A
n_estimators	Número de árvores a serem consideradas.	25	25
min_samples_split	Número mínimo de amostras de dados necessárias para dividir um nó interno.	5	15
max_features	Número de variáveis a serem consideradas ao procurar a melhor divisão.	0.5	0.9

rica e com base no conhecimento no domínio, os resultados obtidos foram superiores. Isso aconteceu porque a segunda técnica de ajustes (*grid search*) foi aplicada em uma base menor, devido ao seu custo de processamento, e então reutilizados os valores encontrados para os experimentos com a base completa. Este cenário pode não refletir o cenário real, dado que alguns parâmetros têm influência direta da quantidade de dados utilizada, a exemplo do $n\_estimators$ (número de árvores/estimadores).

Tabela C.5: Valores dos parâmetros do modelo XGBoost para as duas cidades.

Parâmetros do XGBoost	Descrição	Curitiba	Cidade A
n_estimators	Número de árvores a serem consideradas.	150	25
learning_rate	A taxa de aprendizado usada para reduzir a etapa do gradiente.	0.1	0.01
max_depth	Profundidade máxima de uma árvore.	50	5
min_child_weight	Soma mínima do peso da instância necessária em um nó folha. Se a etapa de partição na árvore resultar em um nó folha com a soma do peso da instância menor que min_child_weight, o processo de criação desistirá de particionar mais.	1	1
gamma	Redução de perda mínima necessária para fazer uma partição adicional em um nó folha da árvore.	0	0
subsample	Proporção da subamostra das instâncias dos dados de treinamento a serem considerados a cada iteração.	0.8	0.8
colsample_bytree	Proporção de subamostra de colunas ao construir cada árvore. A subamostragem ocorre uma vez para cada árvore construída.	0.8	0.8
scale_pos_weight	Parâmetro para controlar o equilíbrio de pesos positivos e negativos, útil para classes desbalanceadas.	1	1

Tabela C.6: Valores dos parâmetros do modelo CatBoost para as duas cidades.

Parâmetros do CatBoost	Descrição	Curitiba	Cidade A
learning_rate	A taxa de aprendizado usada para reduzir a etapa do gradiente.	0.1	0.01
depth	Profundidade máxima da árvore.	8	8
iterations	O número máximo de árvores que podem ser construídas.	10000	100000
l2_leaf_reg	Coeficiente no termo de regularização L2 da função de custo. Qualquer valor positivo é permitido.	5	1

Apêndice D

Tempo de Execução dos Experimentos

Neste Apêndice, são exibidos os tempos de treinamento/execução de cada modelo nos experimentos conduzidos, considerando uma execução e o uso dos valores dos parâmetros obtidos com o Grid Search. Vale ressaltar que, como citado na Seção 5.8 de Ameaças a Validade, os tempos de execução exibidos a seguir são valores aproximados, já que tais valores dependem da capacidade de processamento da máquina e da concorrência de processos por recursos.

Tabela D.1: Tempo de treinamento do *ensemble* relacionado à QP1.

Cidade	Tempo de Treinamento
Curitiba	2,7h
Cidade A	8,8h

Tabela D.2: Tempo de treinamento dos modelos-base e *ensemble* relacionado à QP2.

	Cidade A	Curitiba
RF	7,6min	13min
CatBoost	8,8h	2,7h
XGBoost	1min	3min
Ensemble	8,8h	2,7h

Tabela D.3: Tempo de treinamento do *ensemble* relacionado à QP3, considerando diferentes quantidades de dados.

	Cidade A	Curitiba
5%	2,2h	15,7min
25%	3,5h	47min
52%/50%	5,5h	1,4h
75%	7h	2,1h
100%	8,8h	2,7h

Tabela D.4: Tempo de treinamento do *ensemble* relacionado à QP4, considerando as diferentes combinações de fontes de dados.

	Cidade A	Curitiba
GPS+GTFS	7,8h	2,2h
GPS+GTFS+clima	8h	2,3h
GPS+GTFS+trânsito	8,6h	2,6h
GPS+GTFS+clima+trânsito	8,8h	2,7h

Tabela D.5: Tempo de treinamento do *ensemble* relacionado à QP5 - aprendizagem incremental.

	Cidade A	Curitiba
50%	2,9h	1,4h
50% + 20%	1,4h	38,8min
50% + 20% + 20%	1,4h	39min

Tabela D.6: Tempo de treinamento do *ensemble* relacionado à QP7 - comparação com modelos competidores.

	Cidade A	Curitiba
RLi	0,7min	0,9min
RLo	5,5h	6,8h
SVM	13 dias	-
Ensemble	8,8h	2,7h

Tabela D.7: Tempo de treinamento do *ensemble* relacionado à QP7 - comparação com o competidor RVM.

	Cidade A	RVM(Poly)	RVM(RBF)	RVM(Linear)
5000	16min	1,7min	9min	13min
25000	58,5min	17,6h	28,8h	1,2h