UNIVERSIDADE FEDERAL DO PARANÁ

ISADORA BERGAMI

PREVISÃO DA CONCENTRAÇÃO DE MATERIAL PARTICULADO MP_{10} EM AMBIENTE URBANO UTILIZANDO APRENDIZADO DE MÁQUINA

CURITIBA

2021

ISADORA BERGAMI

PREVISÃO DA CONCENTRAÇÃO DE MATERIAL PARTICULADO MP_{10} EM AMBIENTE URBANO UTILIZANDO APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso de Graduação em Engenharia Ambiental da Universidade Federal do Paraná, apresentado como requisito parcial para obtenção de Bacharel em Engenharia Ambiental

Orientador: Prof. Dr. Emílio G. F. Mercuri

CURITIBA

2021

Agradecimentos

À Deus, por minha vida, por ter me permitido que eu tivesse determinação e saúde e por fazer com que meus objetivos fossem alcançados durante os anos de estudo.

Ao meu orientador, o Prof. Emílio G. F. Mercuri, pelo acompanhamento, orientação e apoio neste trabalho de conclusão de curso, com muita dedicação.

À mestre Leatrice Rodrigues, pela dedicação ao projeto CWBreathe, do qual foram obtidos os dados para a realização do presente trabalho, e pelo apoio prestado a mim.

Aos meus pais, Rita Helena Nesi Bergami e Roberto João Bosco Bergami, pelo suporte que me deram durante toda a minha vida.

À minha irmã, Janaina Bergami, Engenheira de Produção pela UFPR, pela inspiração para escolher a Engenharia e apoio nos momentos mais difíceis.

Ao meu namorado, Gabriel Melo, pela parceria e incentivo durante todo o curso.

À empresa Perkons, pela disponibilização de dados para o estudo e todo auxílio prestado.

Também quero agradecer à Universidade Federal do Paraná e a todos os professores do meu curso pela elevada qualidade de ensino.

RESUMO

O presente estudo trata da previsão da concentração de material particulado com diâmetro aerodinâmico de 10 μm (MP_{10}) em ambiente urbano. Para isso, foram utilizados dados meteorológicos de duas estações do INMET (Instituto Nacional de Meteorologia) e dados de fluxo de veículos de uma via urbana no bairro Jardim das Américas, em Curitiba (PR). A qualidade do ar foi analisada em dois pontos de monitoramento, denominados Politécnico e Perkons, em que foram instalados sensores ópticos SDS011. A metodologia baseou-se na aplicação do algoritmo *Random Forest* (RF) e do método *baseline*, com uso de código em *Python*. O RF foi introduzido por Breiman em 2001 (GENUER; POGGI; TULEAU-MALOT, 2010) e é usado para solução de problemas de classificação e regressão. No estudo, ele foi utilizado para construir 1000 árvores de decisão simples. Foi incluído no RF o método *baseline*, que usa heurística, estatística simples e aleatoriedade para criar previsões de uma certa variável. O *baseline* considerado no trabalho foi um conjunto de dados de uma base histórica de medições de material particulado de estações de monitoramento de Curitiba pertencentes ao projeto de extensão da UFPR intitulado CWBreathe. Diversos cenários foram testados e concluiu-se que a escala temporal diária apresenta melhor desempenho na previsão do MP_{10}, com acurácia de 77,95%, utilizando o método *baseline* e MP_{10} Perkons como descritores. Além disso, as variáveis mais revelantes para o algoritmo, quando não aplicado o *baseline*, foram a temperatura instantânea (°C), velocidade do vento (m/s) e rajada de vento (m/s). Os resultados também mostraram que ao longo do dia ocorreram dois picos com grande quantidade de poluentes no ar, próximo das 8:00 horas e das 18:00 horas, horários em que há os maiores fluxos de veículos circulando na via. Notou-se ainda que há concordância entre as medições de material particulado realizadas nos dois pontos de monitoramento, localizados a 1000 metros de distância.

Palavras-chaves: MP_{10}, SDS011, monitoramento da qualidade do ar, dispersão atmosférica, *Random Forest*.

ABSTRACT

The present study is about the prediction of the concentration of particulate matter with aerodynamic diameter of 10 µm (MP_{10}) in an urban environment. For this purpose, meteorological data from two stations of INMET (National Institute of Meteorology) and vehicle flow data on an urban road in the Jardim das Américas neighborhood in Curitiba (PR) were used. The air quality was analyzed in two monitoring points, named Politécnico and Perkons, where SDS011 optical sensors were installed. The methodology was based on the application of the *Random Forest* (RF) and the *baseline* method, using code in *Python*. RF was introduced by Breiman in 2001 (GENUER; POGGI; TULEAU-MALOT, 2010) and is used for solving classification and regression problems. In the study, it was used to build 1000 simple decision trees. Included in the RF was the *baseline* method, which uses heuristics, simple statistics and randomness to create predictions of a certain variable. The *baseline* considered in the work was a dataset from a historical database of particulate matter measurements from monitoring stations in Curitiba belonging to the UFPR extension project entitled CWBreathe. Several scenarios were tested and it was concluded that the daily time scale presents the best performance in the prediction of MP_{10}, with accuracy of 77.95%, using the *baseline* method and MP_{10} Perkons as descriptors. In addition, the most revealing variables for the algorithm, when *baseline* was not applied, were instantaneous temperature (°C), wind speed (m/s), and wind gust (m/s). The results also showed that throughout the day there were two peaks with large amounts of pollutants in the air, near 8:00 am and 6:00 pm, times when there are the largest flows of vehicles circulating on the road. It was also noted that there is agreement between the particulate matter measurements taken at the two monitoring points, located 1000 meters apart.

Key-words: PM_{10}, SDS011, air quality monitoring, atmospheric dispersion, Random Forest.

Lista de ilustrações

Sumário

1 Introdução

A poluição atmosférica tem sido uma preocupação crescente ao longo dos últimos anos, devido a alta concentração de poluentes no ar em muitas cidades do mundo todo. É uma questão ambiental presente praticamente em todos os países, afetando diretamente a qualidade de vida das pessoas e a economia. Trata-se de um problema de saúde pública (World Health Organization, 2016).

Segundo a Organização para a Cooperação e Desenvolvimento Econômico, em 2050 a exposição ao material particulado se tornará a principal causa ambiental de mortalidade prematura, chegando a 3,6 milhões de mortes por ano no mundo, número duas vezes maior em relação à situação atual (POMÁZI, 2012). A maioria das cidades da América Latina já apresenta níveis de material particulado que excedem os padrões recomendados pela Organização Mundial de Saúde (GOUVEIA et al., 2021). A principal fonte de emissão desse tipo de poluente atmosférico em áreas urbanas provém de partículas emitidas por veículos automotores (KAM et al., 2012).

Há quatro décadas, os avanços tecnológicos e regulatórios têm trazido significativas melhorias na qualidade do ar, e ainda há espaço para aprimoramentos. Alguns países, como a Inglaterra, já apresentam uma rede de monitoramento que realiza a medição de poluentes atmosféricos com o objetivo de mitigar esse problema. Em Londres, o controle dos níveis de poluição do ar é feito por meio de estimativas da forma com que as partículas poluidoras são dispersadas no tempo e espaço. São analisados os poluentes NO_2 (dióxido de nitrogênio) e MP (material particulado), pois estão ligados à problemas de saúde no local (London Govern UK, 2021). Já em Curitiba, atualmente, existem quatro estações de monitoramento localizadas em praças e outros logradouros municipais.

Neste estudo, serão utilizados dados meteorológicos de Curitiba e região metropolitana, dados de fluxo de veículos e uma base histórica de medições de material particulado. Os dados meteorológicos foram obtidos pelo INMET (Instituto de Meteorologia do Paraná), enquanto os dados de concentração de material particulado e os dados de contagem de veículos foram fornecidos pelo projeto de extensão da Universidade Federal do Paraná intitulado “CWBreathe – Curitiba, o ar que você respira” e pela empresa Perkons S.A – Mobilidade e Segurança no Trânsito.

A Perkons é uma empresa especializada em segurança e gestão integrada de tráfego. Sua atuação é baseada no desenvolvimento de estruturas tecnológicas para o acompanhamento do trânsito em tempo real, fiscalização de infrações, contagem de fluxo e identificação de frotas. E o CWBreathe tem como objetivo desenvolver boletins mensais com dados de medição de material particulado disponíveis online (LACTEA, 2021). Suas atividades tem como fundamento o conceito Internet das coisas (IoT, *Internet of Things*), em que uma rede de “objetos físicos” incorporados a sensores, softwares e outras tecnologias é capaz de

reunir e transmitir dados.

O presente trabalho utiliza um método de aprendizado de máquina, o conceito de Internet das Coisas e sensores ópticos de baixo custo para prever a concentração de material particulado em Curitiba, a partir de variáveis meteorológicas, dados de fluxo de veículos e dados de material particulado. A medição de material particulado foi feita por sensores SDS011, que são Para isso, será utilizado o algoritmo *Random Forest*, de forma a identificar as correlações entre as variáveis que norteiam o problema, ou descritores, e encontrar as previsões da concentração de partículas poluidoras.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral deste estudo é estabelecer um sistema que permita previsão da concentração de material particulado (MP_{10}) próximo à uma via urbana de Curitiba utilizando o algoritmo *Random Forest*.

1.1.2 Objetivos Específicos

1. Desenvolver um código embasado em Python para o modelo de aprendizado de máquina;

2. Estabelecer rede de monitoramento utilizando sensores ópticos de baixo custo para medição de material particulado;

3. Identificar a escala temporal em que o algoritmo *Random Forest* apresenta melhor desempenho. Serão testadas as escalas diária e horária;

4. Identificar quais variáveis tem maior importância na previsão da concentração de material particulado em diferentes cenários;

5. Investigar a relação entre fluxo de veículos e a concentração de material particulado, em escala horária.

2 Revisão Bibliográfica

2.1 Qualidade do ar

Segundo Jacobson (2012), desde as primeiras comunidades humanas a poluição do ar causada por atividades antropogênicas têm afetado o meio ambiente. Alguns exemplos são os incêndios florestais e a queima de madeira, vegetação, carvão, óleo, gasolina, diesel, resíduos, produtos químicos, etc. Até o século XX, problemas relativos à poluição atmosférica raramente eram mitigados, já que legislações a respeito do tema eram inexistentes ou pouco impositivas. Os efeitos dos poluentes na saúde humana eram mal compreendidos (JACOBSON, 2012).

De acordo com estudos epidemiológicos, a poluição do ar está diretamente ligada a efeitos negativos na saúde devido à alta concentração de poluentes no meio atmosférico (PARK et al., 2019). Segundo a Resolução CONAMA nº 491, que dispõe sobre padrões de qualidade do ar, poluente atmosférico é definido como "qualquer forma de matéria em quantidade, concentração, tempo ou outras características, que tornem ou possam tornar o ar impróprio ou nocivo à saúde, inconveniente ao bem-estar público, danoso aos materiais, à fauna e flora" (BRASIL, 2018).

O maior efeito da poluição atmosférica na saúde humana ocorre nos pulmões. Pessoas com asma, bronquite, pneumonia e outras doenças respiratórias sofrem mais quando estão expostas a níveis altos de dióxido de enxofre, ozônio ou material particulado, por exemplo (BAIRD, 2018). Várias cidades industrializadas presenciaram o fenômeno de *smog* urbano, onde há alta concentração de fuligem, presente principalmente em emissões veiculares, e de enxofre no ar, causando taxa de mortalidade expressiva. Em Londres, na Inglaterra, no ano de 1952 cerca de 4000 pessoas morreram em alguns dias e 8000 nos meses seguintes devido à alta concentração desses poluentes, pois houve acúmulo deles em uma massa de ar presa próxima à superfície (BAIRD, 2018).

2.2 Material particulado

De acordo com a Resolução CONAMA nº 491 de 2018, o material particulado consiste em partículas de material sólido ou líquido suspensas no ar, na forma de poeira, neblina, aerossol, fuligem, entre outros. O índice MP representa a quantidade de material particulado presente em um dado volume de ar, medido geralmente em microgramas por metros cúbicos ($\mu g/m^3$). Mesmo que as partículas não tenham formato exatamente esférico, seu diâmetro médio é considerado como a propriedade mais importante.

Os menores materiais particulados possuem diâmetro de 0,002 micrômetros (µm) e o

limite superior é de 100 micrômetros (µm). O material particulado grosso possui diâmetro aerodinâmico médio no intervalo de 2,5 a 10 μm e o fino possui diâmetro aerodinâmico médio inferior a 2,5 μm (BAIRD, 2018). As partículas encontradas no ambiente podem ser diferenciadas em grossas e finas (Figura 1).

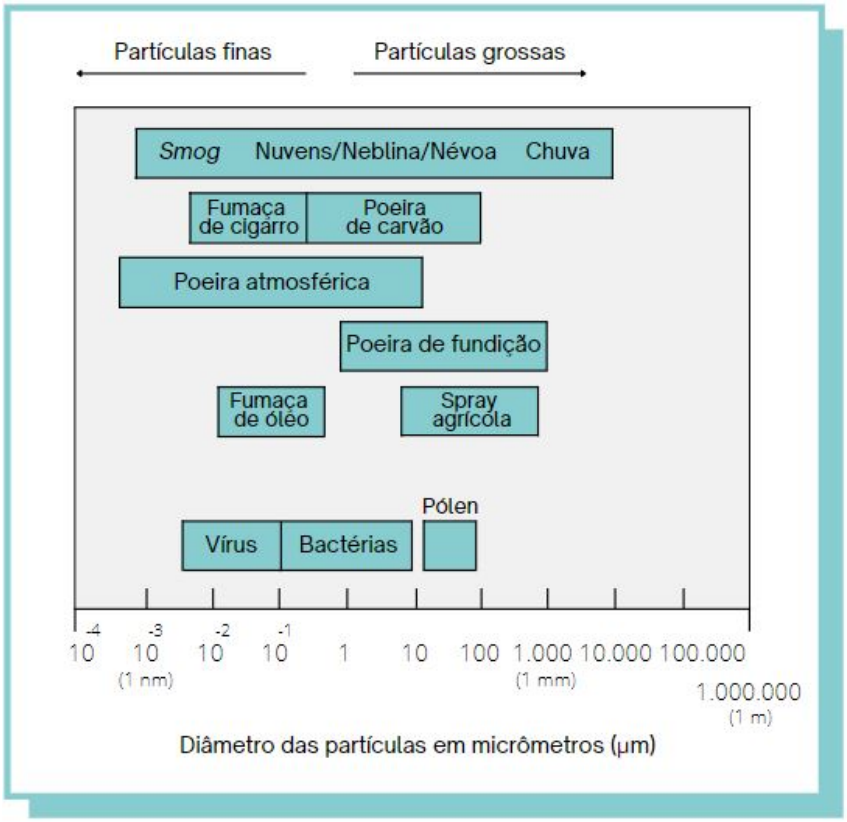

Figura 1 – Tamanhos de partículas comuns finas e grossas. Fonte: Adaptada de (BAIRD, 2018).

As partículas grossas, como pólen e aerossóis marinhos, são provenientes de fontes naturais. No entanto, atividades humanas como moagem de rochas nas pedreiras e cultivo de terra, também geram poluição atmosférica devido ao levantamento de partículas pelo vento. Já as partículas finas têm como origem principal o uso de veículos e a geração de poeira na fundição de metais. Além disso, a combustão incompleta de combustíveis, como gasolina, diesel e carvão, gera grande quantidade de fuligem, classificada também como partícula fina (BAIRD, 2018).

Baird (2018) afirma também que "respirar o ar que contém material particulado é reconhecidamente perigoso para a saúde humana". As agências governamentais em muitos países têm monitorado a concentração de MP_{10}, que compreende a faixa de todas as partículas finas mais uma parte das grossas e são chamadas de partículas inaláveis, pois podem ser aspiradas. O $MP_{2,5}$ corresponde às respiráveis, que chegam ao fundo dos pulmões. A CONAMA nº 491 estabeleceu padrões de qualidade do ar final com limite de 25 $\mu g/m^3$ para o $MP_{2,5}$ e 50 $\mu g/m^3$ para o MP_{10}, considerando a média para o período de referência de 24 horas (BRASIL, 2018).

O monitoramento de material particulado do projeto CWBreathe em Curitiba e região metropolitana é desenvolvido pelo Laboratório de Computação e Tecnologia em Engenharia Ambiental (LACTEA), que está vinculado ao departamento de Engenharia Ambiental do Setor de Tecnologia da Universidade Federal do Paraná (UFPR). Os responsáveis pela instalação dos sensores, transmissão e armazenamento dos dados em um servidor, análise dos resultados e produção deste relatório são alunos de Engenharia Ambiental e professores do Departamento de Engenharia Ambiental (DEA). A elaboração deste produto técnico integra as atividades projeto de extensão universitária intitulado "Curitiba, o ar que voce respira". Esse projeto de extensão está ligado ao projeto de pesquisa da UFPR/CNPq intitulado "Monitoramento e estudo de relações entre material particulado e variaveis meteorológicas em Curitiba", que tem como objetivo criar uma rede de monitoramento da qualidade do ar na capital do estado do Parana. Os boletins do monitoramento estao disponíveis na pagina do laboratório (http://www.lactea.ufpr.br/pesquisa/quali-ar/mp) e sao publicados mensalmente (LACTEA, 2021).

2.3 Fontes móveis

A urbanização vem crescendo rapidamente em todo o mundo, atualmente quatro bilhões de pessoas vivem em áreas urbanas. A estimativa é de que em 2050 aproximadamente dois terços da população mundial viverá em áreas urbanas (RITCHIE; ROSER, 2018). Com este aumento de pessoas vivendo em cidades, a poluição do ar, originada de fontes móveis, também aumentará.

Conforme Cacciola et al. (2002) mais de 600 milhões de pessoas que vivem hoje em áreas urbanas estão expostas a altos níveis de poluentes atmosféricos gerados pelo tráfego, apud (SRIMURUGANANDAM; NAGENDRA, 2010). Segundo Onursal (1997), uma importante questão que intensifica a geração de poluentes em centros urbanos é a ocorrência de congestionamentos de grandes extensões em horários de pico, pois causam a redução da velocidade dos veículos e, consequentemente, maior gasto de combustíveis apud (TEIXEIRA; FELTES; SANTANA, 2008). Além disso, Pant et al. (2017) afirmam que a intensidade com que o material particulado é emitido por fontes móveis depende de características específicas, como o tipo de combustível usado, motor, idade e conservação das condições do veículo.

Existem diversas formas de emissão de MP veicular, como resuspensão, exaustão, partida a frio, desgaste de pneu e motor. Modelos desenvolvidos antes dos anos 80 geralmente só previam emissões de funcionamento a quente para alguns poluentes de automóveis à gasolina e apenas com base em dados de testes de certificação. Em contrapartida, os modelos atuais tendem a incluir todos os tipos de veículos e combustíveis relevantes, poluentes regulamentados e não regulamentados, gases com efeito de estufa e combustível consumo, e são baseadas em testes de emissão (SMIT; NTZIACHRISTOS; BOULTER,

2010). As emissões veiculares podem ser medidas em condições controladas em laboratório e realizadas em chassis ou instalações com dinamômetros de motor. Os operadores de ensaio têm controle sobre as condições ambientais e outros parâmetros, possibilitando a repetibilidade dos resultados (FRANCO et al., 2013).

Segundo Wrobel *et al.* (2000), a emissão gerada no tráfego de veículos corresponde a mais de 50% do total de material particulado emitido. Em Londres, mais de 80% das emissões de MP são provenientes de tráfego urbano, de acordo com DoT (2002). E em Atenas, essa taxa é de 66,5%, conforme Economopoulou and Economopoulos (2002) apud (SRIMURUGANANDAM; NAGENDRA, 2010). Tendo em vista que a principal fonte de MP é o tráfego de veículos, é importante e necessário quantificar o nível de emissão dessas partículas e determinar seu comportamento, desde a emissão até o transporte para longe da fonte – estradas movimentadas e rodovias (ZHU et al., 2002).

Em Curitiba, os sensores SDS011 a serem analisados encontram-se no bairro Jardim das Américas. Segundo a Prefeitura de Curitiba (2021), o bairro possui edificações de tipologia horizontal, de uso predominantemente habitacional, é considerado de padrão baixo/médio e possui 27% dos lotes vagos. A área do bairro é formada principalmente por comércios e serviços, cerca de 36%, e residências, cerca de 41%. O Jardim das Américas está no setor Norte da Linha Verde.

Os dados da pesquisa Origem Destino do IPPUC (Instituto de Pesquisa e Planejamento Urbano de Curitiba) estão apresentados na Figura 2. No bairro Jardim das Américas há 1,3 carros/domicílio e 0,14 motocicletas/domicílio, o que totalizam 7.678 automóveis e 366 motocicletas (IPPUC, 2017b).

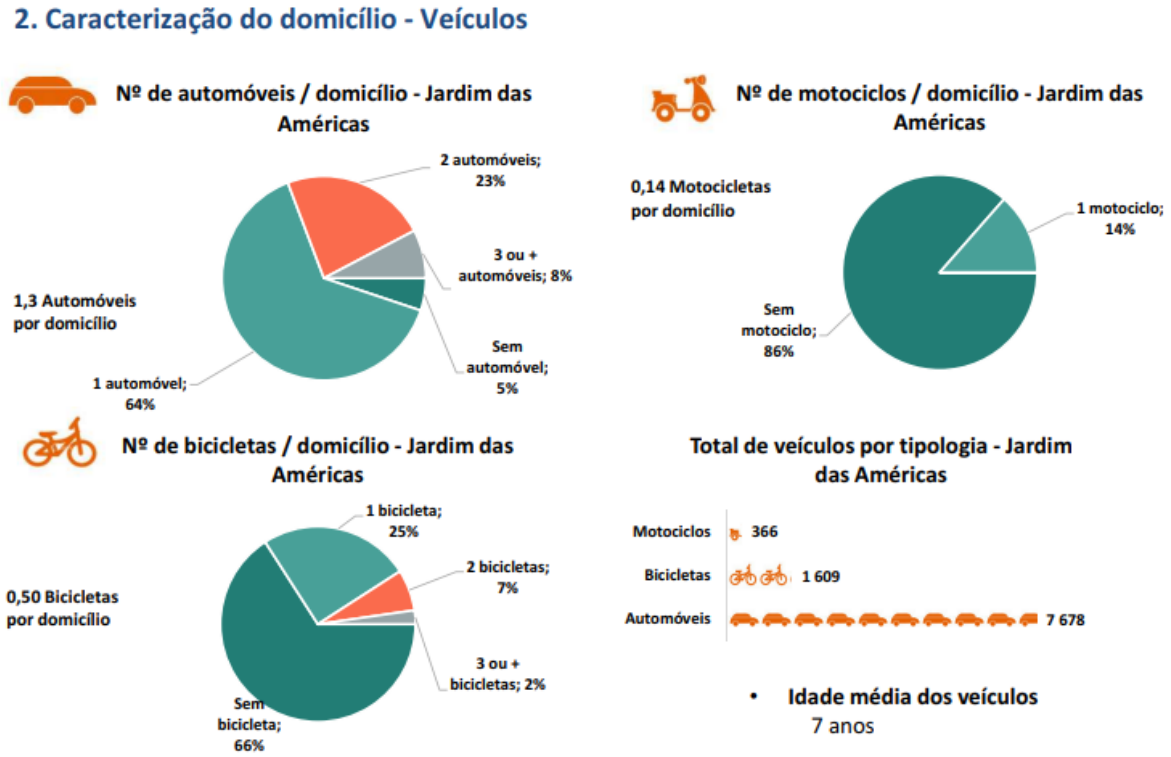

Figura 2 – Pesquisa Origem Destino Jardim das Américas. Fonte: IPPUC (2017b).

Essa quantidade de veículos representa apenas a frota dos residentes no bairro Jardim das Américas, mas o fluxo de veículos é proveniente de outros bairros também. Um ponto importante a ser lembrado é que neste bairro se encontra o centro Politécnico da UFPR, destino de milhares de estudantes todos os dias da semana.

2.4 Dispersão atmosférica

Segundo Stull et al. (2015), dispersão atmosférica é o nome dado ao espalhamento e movimento de poluentes. Depende da velocidade e direção do vento, do levantamento da pluma, da região de ar poluído e da turbulência atmosférica. A velocidade do vento médio e a turbulência também podem influenciar na concentração de poluentes.

A turbulência é definida como o efeito de um conjunto de turbilhões ou estruturas turbulentas de diferentes tamanhos. Uma pluma de emissão de poluentes pode se elevar ou se rebaixar de acordo com as condições do ambiente. A forçante térmica, gerada devido à radiação solar, e a forçante mecânica, gerada devido à velocidade do vento, influenciam na dinâmica atmosférica e também na dispersão de poluentes (STULL et al., 2015).

O aumento na estabilidade do ar diminui a velocidade do vento próximo à superfície e aumenta o vento acima da superfície e, consequentemente, menos ar sobe e desce de maneira flutuante (JACOBSON, 2012). Os estudiosos Pasquill e Gifford criaram uma

classificação para denotar os diferentes tipos de turbulência que variam de A até G, em que A representa uma atmosfera bem instável e G, bem estável (Tabela 1).

Tipos de turbulência de Pasquill-Gifford durante o dia				**Tipos de turbulência de Pasquill-Gifford para a noite**		
Velocidade	Radiação solar			Velocidade	Cobertura de nuvens	
do vento (m/s)	Forte	Moderada	Fraca	do vento (m/s)	$\geq 4/8$	$\leq 3/8$
<2	A	A a B	B	<2	G	G
2 a 3	A a B	B	C	2 a 3	E	F
3 a 4	B	B a C	C	3 a 4	D	E
4 a 6	C	C a D	D	4 a 6	D	D
>6	C	D	D	>6	D	D

Tabela 1 – Classificação de Pasquill-Gifford. Fonte: Adaptada de (STULL, 2015).

As condições favoráveis à dispersão de poluentes são as instáveis, representadas pelas classes A, B e C. A condição neutra é representada por D e as condições desfavoráveis, estáveis, são as indicadas por E, F e G. "As classes de estabilidade de Pasquill-Gifford são obtidas a partir de grandezas meteorológicas das médias horárias, como a velocidade do vento e a radiação solar ou cobertura de nuvens, medidas a poucos metros da superfície. Elas fornecem apenas uma ideia aproximada da estabilidade da subcamada superficial da camada limite atmosférica" (IAP, 2011). Para a condição A ocorre turbulência rigorosa e para a condição G a turbulência é suprimida (STULL et al., 2015).

O modelo CFD (em inglês, Computational Fluid Dynamics) é uma das ferramentas mais avançadas disponíveis para os investigadores compreenderem quais são os fatores que afetam a dispersão de poluentes nas ruas. Segundo Jeanjean et al. (2017), o modelo proporciona uma visão estável da realidade e corresponde a uma imagem fixa do fluxo do vento e das concentrações de poluentes. Essas aproximações são feitas pois, na realidade, o vento apresenta oscilações em magnitude e direção e as concentrações de poluentes são variáveis.

De acordo com o estudo de (JEANJEAN et al., 2017), as árvores podem ser uma estratégia rentável para promover a deposição de partículas e aumentar a dispersão aerodinâmica das concentrações de NO_2 em uma avenida. Barreiras sólidas podem melhorar a qualidade do ar, mas têm efeitos prejudiciais aos ciclistas e condutores que circulam pelo local, devido à alta concentração de poluentes nas ruas. Os autores citam ainda que a melhoria do desempenho de uma barreira sólida pode ser feita com a aplicação de tinta fotocatalítica nesse material para promover a deposição em superfícies.

2.5 Random Forest

Random Forest (RF) é um algoritmo popular e muito eficiente, baseado em ideias de agregação de modelos, tanto para problemas de classificação como de regressão, introduzido

por Breiman em 2001 (GENUER; POGGI; TULEAU-MALOT, 2010). É um método de aprendizado de máquina que opera através da construção de uma infinidade de árvores de decisão. Já uma floresta aleatória, *Random Forest* em português, consiste em um número definido de árvores de decisão simples. Cada um dos componentes que formam uma RF usa um subconjunto dos dados disponíveis, independentes entre si. A Figura 3 mostra um exemplo de uma árvore de decisão simplificada. Para cada árvore, descritores são selecionados com igual probabilidade. Esses descritores são variáveis que influenciam o problema.

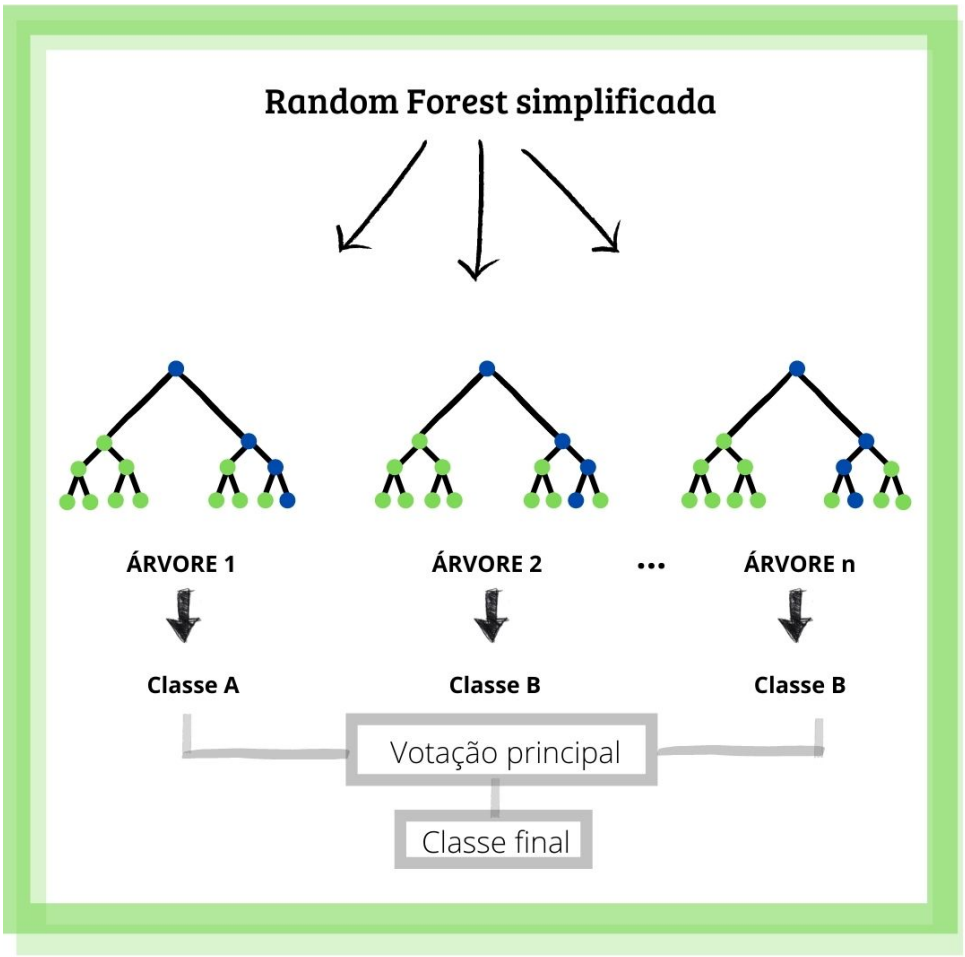

Figura 3 – Árvore de decisão simplificada. Fonte: Adaptada de (KOEHRSEN, 2017b).

A saída prevista é obtida agregando e calculando a média de previsões individuais de todas essas árvores de componentes. A maior importância é dada à variável com a maior soma de re-substituição (KAMIŃSKA, 2018). O RF retorna no fim do processamento a acurácia do algoritmo, em %. Essa medida é o grau em que algo é verdadeiro ou exato. A elevada acurácia implica em um pequeno erro. Se tanto a classe prevista como a real coincidirem, então a previsão é considerada como exata (KUMAR et al., 2018).

A obtenção de um bom desempenho de uma *Random Forest* depende da escolha de características relevantes como descritores, ou seja, variáveis que realmente estejam

relacionadas ao problema, e também de baixa correlação entre as árvores de decisão. Dessa forma, de uma maneira geral, a árvore se moverá para a direção correta. A *Random Forest* garante que as árvores de decisão individuais não se corrrelacionem entre si.

Para isso, utiliza-se o conceito *Bagging*, que provém de *Bootstrap Aggregation*. *Bootstraps* são amostras diferentes da base de dados utilizadas para aprender hipóteses diferentes. O *Bagging* trata-se de uma reamostragem com reposição, é o agrupamento de *Bootstraps*. Consiste em uma técnica de agrupamento utilizada para reduzir a variância de uma função de previsão estimada. Enquanto para a regressão simplesmente encaixamos a mesma árvore de regressão muitas vezes e calculamos a média do resultado, para a classificação, cada comissão de árvores vota individualmente, a favor da classe prevista (FRIEDMAN, 2017).

A árvore de decisão simples utiliza como possibilidade todas as características fornecidas, a *Random Forest* usa um subconjunto aleatório de *features*. Isso resulta em uma menor correlação e maior diversificação entre as árvores. Esse método é conhecido por *Feature Randomness*. Essas duas metodologias, *Bagging* e *Feature Randomness*, fazem com que o algoritmo tenha árvores de decisão diversas (FRIEDMAN, 2017).

A limitação do *Random Forest* é que a sua capacidade de prever novos resultados está limitada ao intervalo do conjunto de dados de formação, pois não prevê dados com parâmetros variáveis fora do intervalo de formação (sem extrapolação) (ZIMMERMAN et al., 2018). Apesar da limitação, Archer e Kimes (2008) afirmaram que este método de construção capaz de combinar os conceitos de amostragem e seleção aleatória de características tem demonstrado melhoria no desempenho em relação a outros algoritmos de aprendizagem de máquinas e modelos de regressão linear apud (KAMIŃSKA, 2018).

Um estudo feito na capital do Irã, Teerão, mostrou a comparação entre três métodos de *machine learning*, o *Random Forest*, o Gradiente Descendente Estocástico (*Stochastic Gradient Descent*) e o *AdaBoost*) para realizar a modelagem espaço-temporal da concentração de $MP_{2,5}$. A maior acurácia encontrada entre os métodos aplicados foi a do RF, com valores de 0,926 (92,6%), 0,94 (94%) e 0,949 (94,9%) para diferentes cenários (SHOGRKHODAEI; RAZAVI-TERMEH; FATHNIA, 2021).

Outro estudo sobre previsão da qualidade do ar realizado na China utilizando *Random Forest* também mostrou resultados com alta acurácia. Em Shenyang, o algoritmo resultou em 81% de precisão do AQI, indicador usado pelo governo para comunicar publicamente quão poluído o ar se encontra no local (YU et al., 2016).

3 Métodos

3.1 Localização dos sensores

Há dois sensores de monitoramento, identificados como SDS011 Politécnico e SDS011 Perkons. Ambos fornecem dados de material particulado no local e um deles, o Perkons, fornece também dados de contagem de veículos. Os equipamentos de medição estão localizados no bairro Jardim das Américas, em Curitiba (PR), a 1000 metros de distância entre eles.

Figura 4 – Lombada eletrônica da Perkons. Fonte: O Autor (2021).

O sensor Politécnico está localizado na Universidade Federal do Paraná (UFPR) – Campus Centro Politécnico, em um container de pesquisa que tem acesso à energia elétrica e internet. Já o sensor Perkons está localizado na lombada eletrônica da Av. Cel. Francisco H. dos Santos (Figura 4), em frente ao mercado Casa Fiesta.

Os fluxos de veículos medidos pela empresa Perkons são diferentes de acordo com os

dois sentidos da Avenida. Os sentidos são: BR277, direção Nordeste e Av. Comendador Franco (Av. das Torres), direção Sudoeste (Figura 5). Há um semáforo próximo ao ponto de monitoramento na avenida, portanto esse trecho geralmente fica congestionado e as emissões ocorrem com maior intensidade.

Figura 5 – Localização dos dois sensores SDS011. Fonte: O Autor (2021).

3.2 Medição de material particulado

Nos pontos Politécnico e Perkons foram instalados sensores ópticos de baixo custo para medir a concentração de $MP_{2,5}$ e MP_{10}, juntamente com um *Rasperry Pi*, computador de placa única de tamanho reduzido. O sensor utilizado é denominado SDS011 (Figura 6) e seu princípio de funcionamento é o espalhamento da luz. São fornecidos por esse equipamento dados a cada 5 segundos, enviados pelo *Raspberry Pi* à uma pasta do *Dropbox*. O objetivo do sensor é captar diferenças de concentração de MP ao longo do dia. A acurácia média deste equipamento é de 88,58% (CHAVES, 2021).

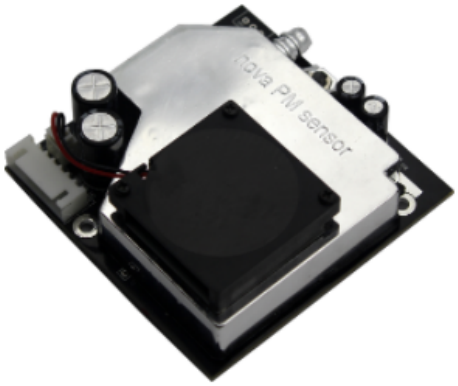

Figura 6 – Sensor SDS011. Fonte: AQICN (2021)

O sensor óptico SDS011 Politécnico foi instalado pela equipe do projeto CWBreathe. E o sensor de monitoramento SDS011 Perkons foi instalado no interior de lombada eletrônica pela equipe técnica da empresa (Figura 7).

Figura 7 – Instalação do SDS011 na lombada eletrônica. Fonte: O Autor (2021).

3.3 Dados de material particulado

No total, há dois conjuntos de dados de material particulado, referente aos sensores de monitoramento. Cada conjunto tem dois arquivos em formato csv (*comma-separated values*) e correspondem ao material particulado de diâmetro aerodinâmico inferiores a 2,5

μm ($MP_{2,5}$) e de diâmetro aerodinâmico no intervalo de 2,5 a 10μm (MP_{10}). O sensor do Centro Politécnico possui medições desde o dia 04/10/2019 e o administrado pela empresa, desde 16/10/2020.

As informações foram disponibilizadas pela empresa Perkons através de uma plataforma chamada Grafana, onde é possível visualizar e analisar métricas de dados por meio de gráficos. Essa plataforma pode ser acessada em qualquer sistema operacional e nela é possível criar *dashboards* dinâmicos para facilitar a análise das informações.

3.4 Dados do fluxo de veículos

Segundo um dos relatórios da pesquisa Origem Destino de Curitiba e região metropolitana desenvolvido pelo IPPUC, os períodos com maior intensidade de tráfego ocorrem entre as 7:00 e 8:00 horas e entre as 17:30 e 18:30 horas. A variação horária do tráfego Foi analisada em 16 pontos de contagens (IPPUC, 2017a).

O mesmo comportamento é encontrado no perfil diário dos dados de veículos deste trabalho (Figura 8), onde há maior circulação de veículos próximo às 7:00 horas e às 18:00 horas. Perto das 12:00 horas, principalmente no sentido da BR-277, houve um outro período com valores altos. Espera-se que a origem da emissão desses poluentes sejam os carros, veículos mais presentes na rua analisada, e que seja encontrada uma relação proporcional entre o número de veículos circulando na avenida e a concentração de MP.

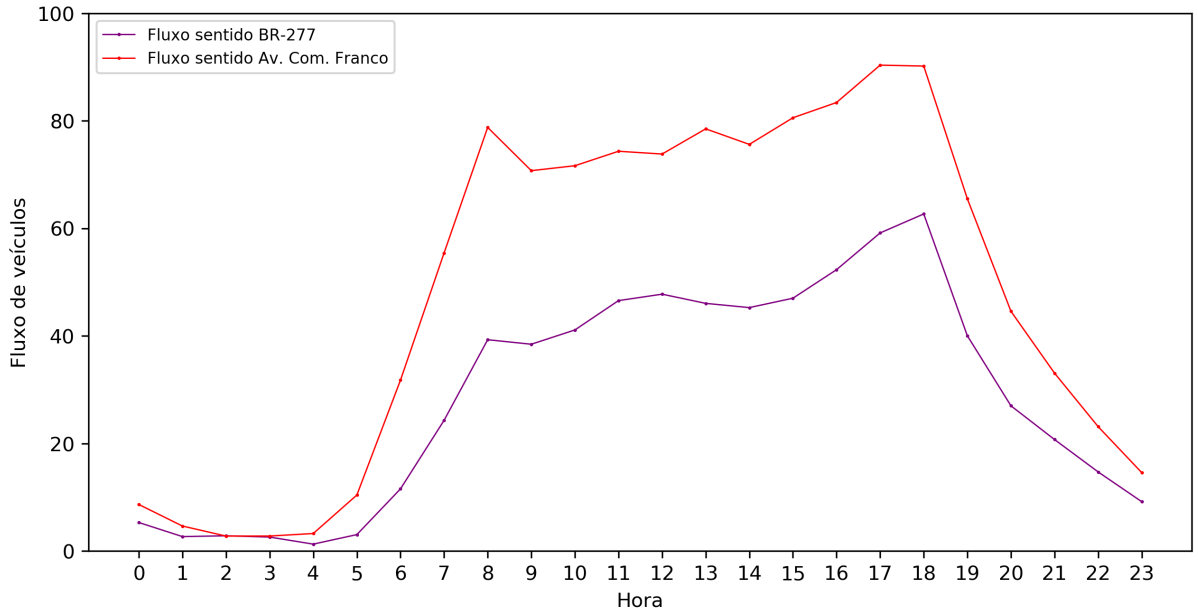

Figura 8 – Perfil diário do fluxo total de veículos na BR277 e Av. Com. Franco. Fonte: O Autor (2021).

Nota-se também que a média horária do fluxo de veículos sentido Av. Comendador Franco é maior do que a média horária da BR277, com valor médio de 80 e 50 veículos

por hora respectivamente, no período em que há radiação solar. Por dia, passam cerca de 1.920 veículos em direção à BR e 1.200 em direção à avenida.

Em alguns períodos no início do ano de 2021 houveram restrições nos horários de atendimento de restaurantes, comércio e shoppings devido à determinação de bandeira vermelha para combate à pandemia do Coronavírus (COVID-19). Além disso, a maior parte das escolas permaneceram fechadas, portanto, o fluxo de veículos em condições normais provavelmente é maior do que o medido nesse período.

A empresa Perkons disponibilizou dados de contagem de veículos nos dois sentidos, BR277 e Avenida Comendador Franco, com diferenciação entre tipos de veículos. Foram fornecidos: o número de veículos pequenos, médios e grandes que circulam na via e também o número total do fluxo de veículos. A diferenciação citada considera o comprimento do automóvel, sendo considerado pequeno o comprimento entre 1 e 6 metros, médio entre 7 a 10 metros e grande maior do que 11 metros. A Figura 9 mostra que os veículos pequenos representam a maior parte do fluxo total circulando na via.

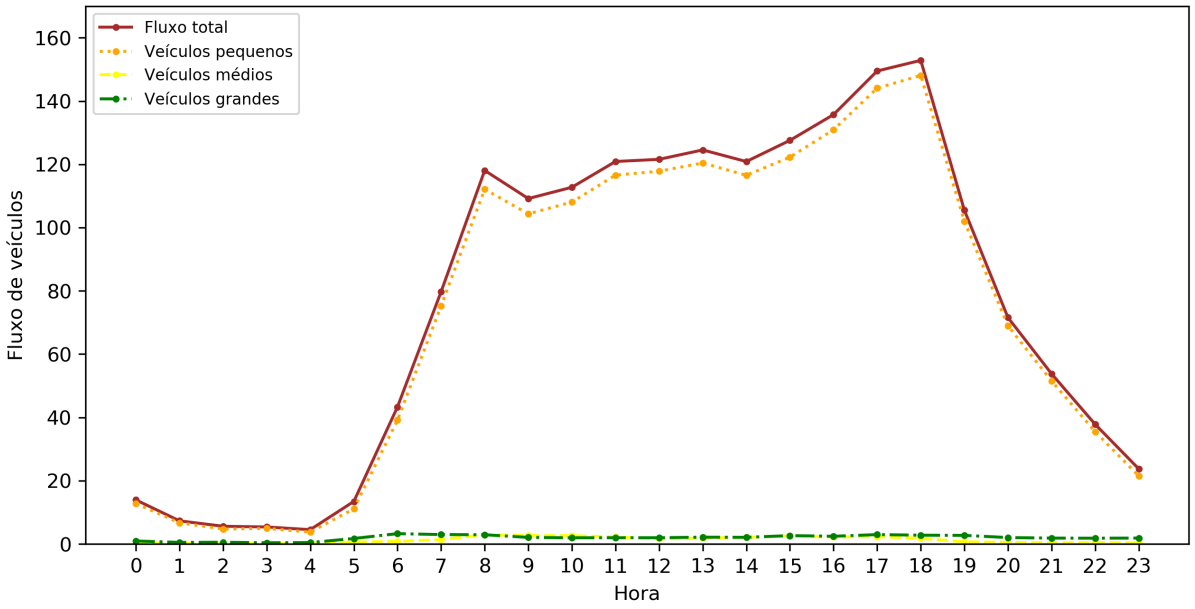

Figura 9 – Perfil diário do fluxo total de veículos na BR277 e Av. Com. Franco com diferenciação entre tipos de veículos. Fonte: O Autor (2021).

Apesar de em menor quantidade, os veículos grandes emitem alta concentração de poluentes e portanto, também são uma preocupação ambiental que causa poluição atmosférica do local. A figura 10 mostra o perfil diário somente do fluxo de veículos grandes na via.

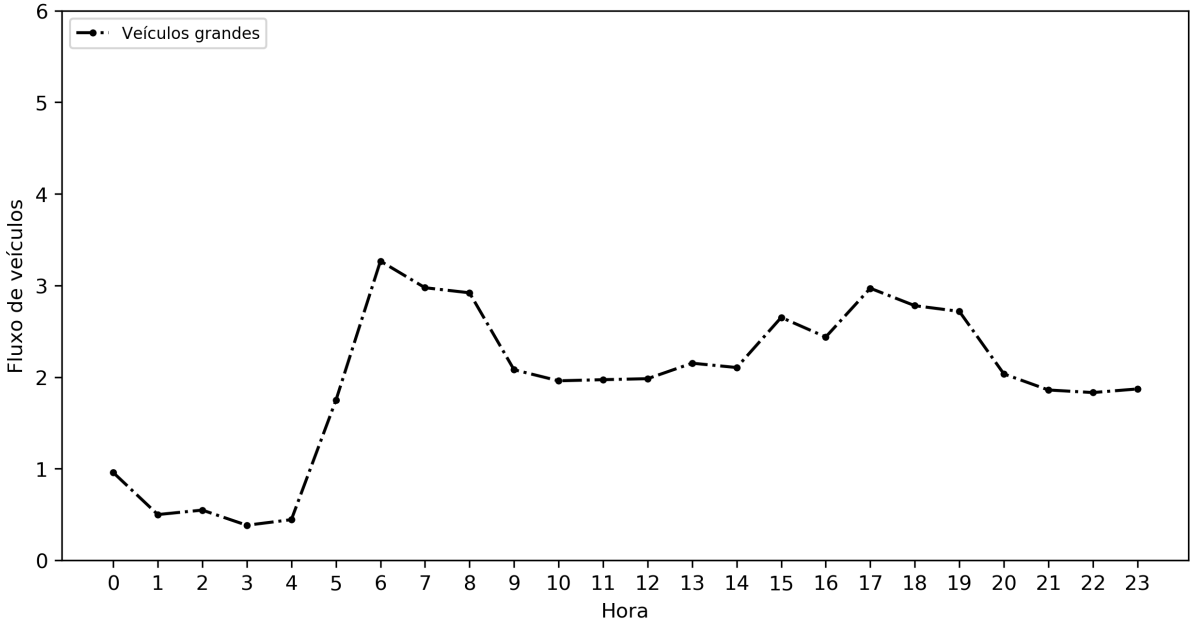

Figura 10 – Perfil diário do fluxo de veículos grandes na BR277 e Av. Com. Franco com diferenciação entre tipos de veículos. Fonte: O Autor (2021).

3.5 Dados meteorológicos

Para entender a dinâmica de dispersão de poluentes na atmosfera, foram incluídas no estudo algumas variáveis meteorológicas, já que o fluxo de veículos no local não é o único fator ligado à concentração de poluentes. Essas variáveis têm relação direta com a concentração de material particulado, pois influenciam no seu movimento no ar e podem auxiliar no entendimento de resultados obtidos no estudo.

Os dados meteorológicos foram obtidos pelo site do INMET (Instituto Nacional de Meteorologia). Na página, é possível encontrar dados de várias estações meteorológicas em formato de tabela com o intervalo de datas desejado. Foram escolhidas duas estações meteorológicas automáticas, uma em Curitiba (INMET, 2021b) e outra em Colombo (INMET, 2021a).

A estação automática de Curitiba é identificada por CURITIBA (A807) e está localizada no Centro Politécnico, próxima aos pontos de monitoramento do estudo. A estação automática em Colombo é identificada por COLOMBO (B806) e está localizada próximo à BR-476. As tabelas geradas estão em formato *csv* e apresentam dados de temperatura instantânea (°C), umidade instantânea (%), ponto de orvalho instantâneo (°C), pressão instantânea (hPa), radiação (kJ h^{-1} m^{-2}), velocidade do vento (m/s), rajada de vento (m/s), entre outros.

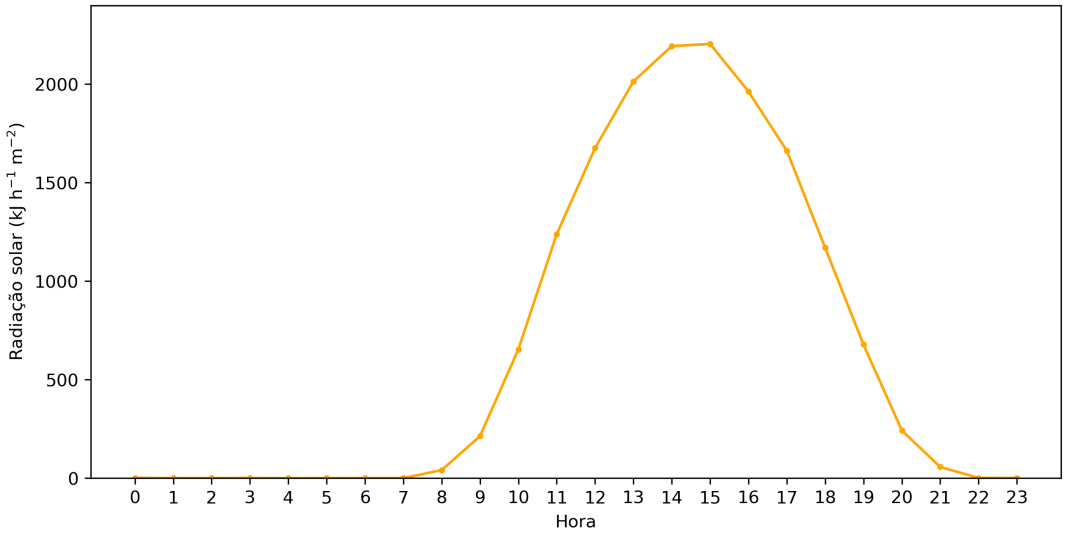

Figura 11 – Perfil diário da radiação solar. Fonte: O Autor (2021).

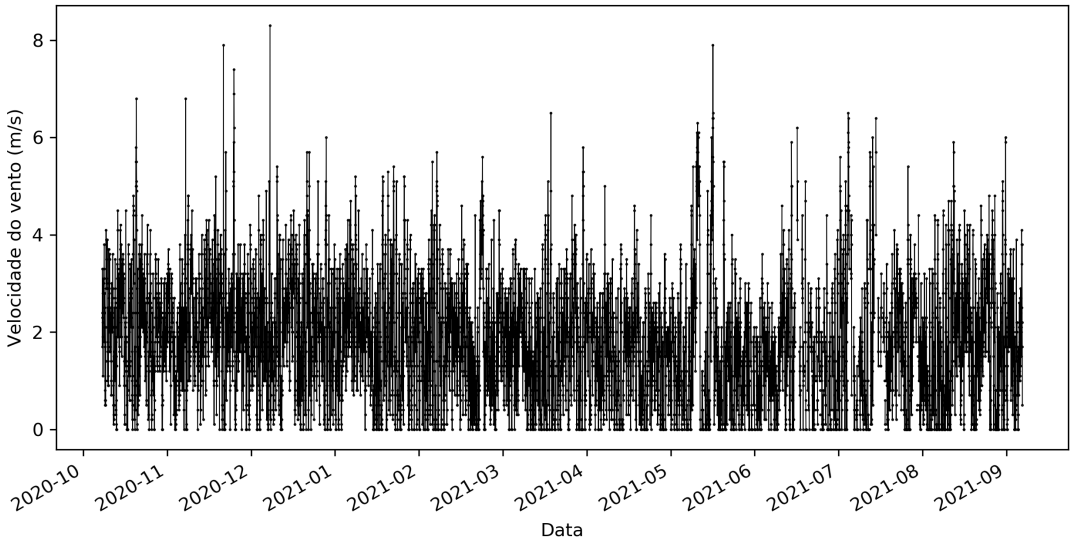

Figura 12 – Série temporal com médias horárias da velocidade do vento. Fonte: O Autor (2021).

O perfil diário da radiação solar foi demonstrado na Figura 11 e a série temporal da velocidade do vento foi demonstrada na Figura 12. Nota-se que a incidência de luz solar tem variações varia ao longo do dia e tem seu valor máximo próximo das 14:00 horas (Figura 11). A série completa dos dados de velocidade do vento, em m/s, mostra que há variações expressivas ao longo dos dias analisados, com valores entre 0 e 8 m/s e média aproximada de 2,5 m/s. (Figura 12).

3.6 Análise de dados

O código de programação do sensor foi feito em *Python*, configurado dentro do computador conectado ao sensor localizado do lado de fora dos locais escolhidos, para receber

o ar do ambiente. Todos os arquivos de entrada estão no formato *csv*. Para realizar as leituras dos arquivos também foi usado código em *Python* em outro computador e foi usada a biblioteca *Pandas*, pelo comando *pd.read_csv*. Podem ser obtidos alguns dados falhos, quando o equipamento não mediu corretamente. Esses valores não aparecem nos resultados. Durante o desenvolvimento do código, foram utilizadas também as bibliotecas *Numpy* e *Datetime*. Especificamente para a plotagem dos dados, foram usadas a *matplotlib*, *matplotlib.pyplot* e *matplotlib.dates*.

Após as leituras e médias dos dados, foram formados seis *dataframes*, quatro de MP, diferenciados entre $MP_{2,5}$ e MP_{10}, e dois de fluxo total de veículos, nos dois sentidos da avenida, com diferenciação entre veículos pequenos, médios e grandes. Esses *dataframes* foram unidos em uma só tabela para facilitar as análises. Foram incluídas nessa tabela a radiação solar e velocidade do vento. Com os dados devidamente organizados por nomes de variáveis, foi possível fazer comparações diversas. Uma delas foi a comparação entre as concentrações encontradas pelos dois sensores, Politécnico e Perkons, por meio de um diagrama de dispersão.

Foram feitos também os perfis diários do $MP_{2,5}$ e MP_{10} para compreender o comportamento dessas partículas ao longo do dia. A média horária dos dados foi feita através do comando *resample('H').mean()*. Outra análise do trabalho tratou da relação entre o fluxo de veículos e o material particulado. Além disso, foi analisada a relação entre o MP e a radiação e entre o MP e a velocidade do vento, em gráficos separados. O fluxo de veículos nos dois sentidos da avenida também foi explorado. O período de dados estudado foi de 17/10/2020 à 31/07/2021.

3.7 Aplicação do método Random Forest

A metolodogia para aplicação do *Random Forest* foi baseada em um curso do site *Towards data science* escrito por Will Koehrsen, cientista de dados. O curso apresenta uma explicação de como a árvore de decisão funciona utilizando um exemplo de previsão da temperatura máxima para o dia de amanhã usando um ano de dados históricos de temperatura para a cidade de Seattle, em Washington (USA) (KOEHRSEN, 2017b). O curso ainda mostra o código em *Python* utilizado para realizar essa previsão por meio do algoritmo *RF* e alguns meios de apresentação dos resultados finais encontrados (KOEHRSEN, 2017a).

O *Random Forest* utiliza vários descritores como dados de entrada para alcançar a solução do problema. Um descritor é uma variável relacionada ao problema que pode ter uma relação positiva, diretamente proporcional, ou negativa, inversamente proporcional, à variável de interesse. Para a escolha dos descritores, foram analisados vários cenários com diferentes dados. As configurações foram testadas e então determinado que seria prevista a concentração de MP_{10} do Politécnico, pois engloba também as concentrações de $MP_{2,5}$

do local. Ao final do processamento, o algoritmo fornece a lista com porcentagens das importâncias de cada descritor usado no problema. A partir dessas importâncias, o melhor conjunto de descritores foi escolhido.

No estudo, foram utilizados os seguintes descritores: fluxo de veículos sentido BR277, fluxo de veículos sentido Av. Comendador Franco, temperatura instantânea (°C), velocidade do vento (m/s), direção do vento (graus) e radiação (kJ h^{-1} m^{-2}) referentes à estação do INMET de Curitiba, rajada máxima de vento (m/s) e umidade relativa do ar (%) da estação meteorológica Colombo, MP_{10} medido no Politécnico e um conjunto de dados conhecido por *baseline*. Além disso, foi incluída a hora do dia como descritor, utilizando o comando *features.index.hour*.

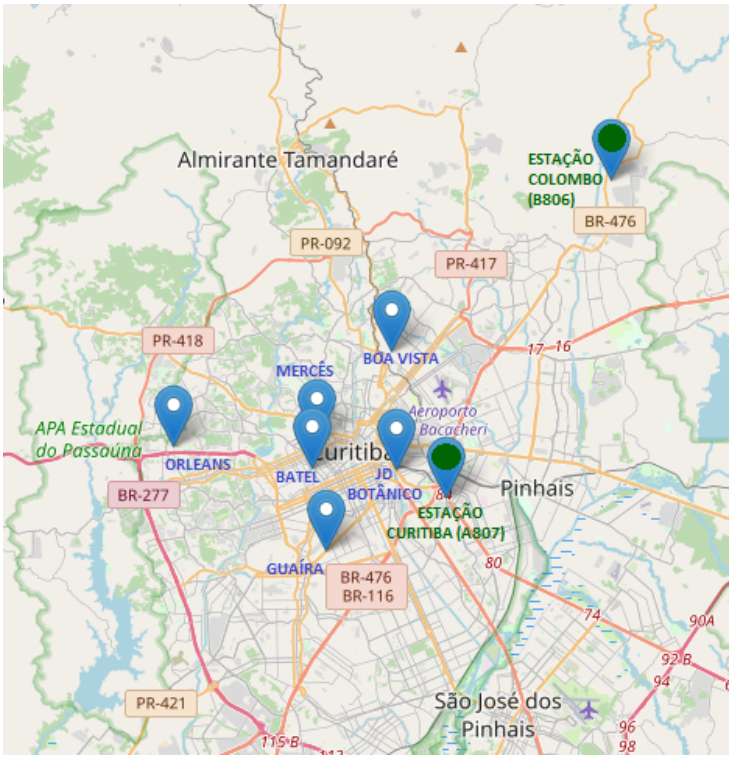

Figura 13 – Mapa das estações meteorológicas do INMET e dos pontos usados no *baseline*. Fonte: O Autor (2021).

O *baseline* é um método que usa heurística, estatística simples e aleatoriedade para criar previsões de uma certa variável. É possível medir o desempenho do *baseline* usando métricas de erro como a acurácia. A eficiência da previsão com o *baseline* é comparada com a eficiência do algoritmo de aprendizado de máquina do algoritmo RF. O *baseline* considerado corresponde às concentrações médias de uma base histórica de medições de material particulado de seis estações de monitoramento de Curitiba pertencentes ao projeto CWBreathe. Estão localizadas nos seguintes bairros: Batel, Jardim Botânico, Boa Vista, Guaíra, Mercês e Orleans (Figura 13). As estações meteorológicas de Curitiba e Colombo também estão ilustradas na Figura 13.

Mesmo essas estações de monitoramento sendo distantes uma das outras, entende-se que a dinâmica da atmosfera da cidade segue um comportamento semelhante em todos os pontos medidos (Figura 14).

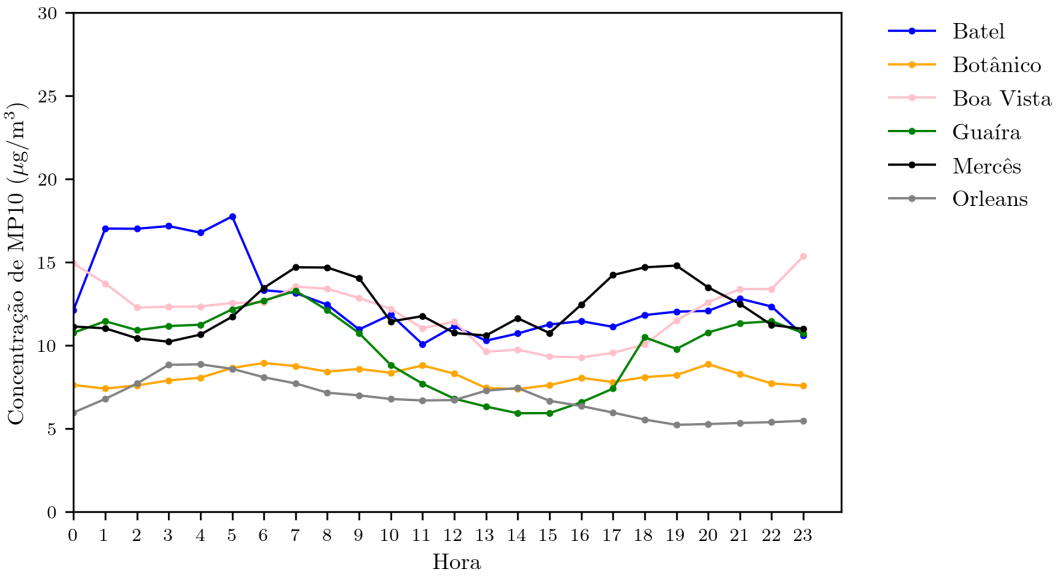

Figura 14 – Perfil diário das estações que formam o *baseline*. Fonte: O Autor (2021).

O *RF* foi aplicado utilizando também código em *Python*. Primeiramente, foi feita a leitura dos dados de material particulado, do fluxo de veículos, dos dados meteorológicos fornecidos pelo INMET e também da série histórica que formará o *baseline* do problema. Todos os arquivos lidos estavam em formatos csv (*comma separated value*). Após a leitura dos dados, foi criado um *dataframe* com os descritores e medição de MP (Figura 15).

datas	Total_CF	Total_BR	Temp. Ins. (C)	Umi. Ins. (%)	Vel. Vento (m/s)	Dir. Vento (graus)	baselineCWBreathe	Rajada colombo (m/s)	MP10_FHS_PER	Radiacao (KJ/m²)	hora do dia
2020-10-17 00:00:00	15.083333	9.333333	12.3	85.0	1.7	87.0	10.307793	6.5	11.755000	0.0	0
2020-10-17 01:00:00	8.583333	7.166667	11.6	88.0	0.6	92.0	10.920555	2.7	9.371667	0.0	1
2020-10-17 02:00:00	7.333333	7.416667	11.9	91.0	0.5	57.0	12.319794	2.7	7.261667	0.0	2
2020-10-17 03:00:00	2.750000	2.916667	11.6	95.0	0.8	100.0	12.444399	2.3	6.542373	0.0	3
2020-10-17 04:00:00	4.833333	2.750000	11.8	95.0	0.6	97.0	15.352023	2.3	6.563333	0.0	4

Figura 15 – *Dataframe features*. Fonte: O Autor (2021).

Em seguida, a média horária do *dataframe* completo que contém os descritores foi calculada por meio do comando *resample.mean('d')*. Após essa etapa, foi realizada a limpeza dos dados faltantes (*NaN*) utilizando o comando *dropna()*.

Na sequência, foram determinadas as porcentagens de dados de treinamento e de teste do algoritmo. O conjunto de dados foi dividido entre 73% para dados de treinamento e

27% para dados de teste. Essas porcentagens foram escolhidas de acordo com tentativas realizadas pela análise dos resultados finais.

Seis configurações diferentes foram desenvolvidas para análise da performance do *Random Forest*:

1. Base horária sem *baseline* como descritor
2. Base horária com *baseline* como descritor
3. Base horária com *baseline* e MP_{10} Perkons como descritores
4. Base diária sem *baseline* como descritor
5. Base diária com *baseline* como descritor
6. Base diária com *baseline*w e MP_{10} Perkons como descritores

4 Resultados e Discussão

O diagrama de dispersão indica a existência de correlação entre variáveis e o tipo de correlação – pode ser positiva, negativa, ou não haver relação. Por meio da realização desse diagrama, demonstra-se que há correlação positiva entre a concentração medida pelo SDS011 no Politécnico e a concentração medida pelo sensor SDS011 na lombada da empresa Perkons (Figura 16). Isso ocorre possivelmente pela proximidade entre os pontos de monitoramento, 1000 metros. Esse resultado mostra também que esse tipo de sensor pode ser satisfatório para medição dessas partículas poluidoras.

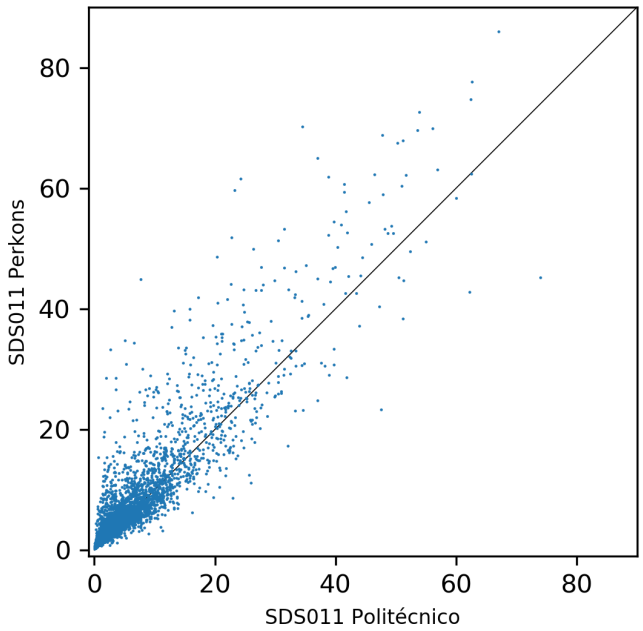

Figura 16 – Diagrama de dispersão. Fonte: O Autor (2021).

Como já mencionado, a concentração de material particulado varia ao longo do dia e está relacionada à diversos fatores. Para analisar seu comportamento, foram analisados os perfis diários do material particulado ($MP_{2,5}$ e MP_{10}) das duas estações - Politécnico e Perkons. Observa-se que no período do dia há dois picos com grande quantidade de poluentes, próximo das 8:00 horas e das 19:00 horas (Figura 17). Nota-se ainda que no período da madrugada, entre 0:00 e 5:00 horas, também há valores elevados de concentração, possivelmente pelo tráfego de trens que circulam próximo à região e emitem grande quantidade de MP.

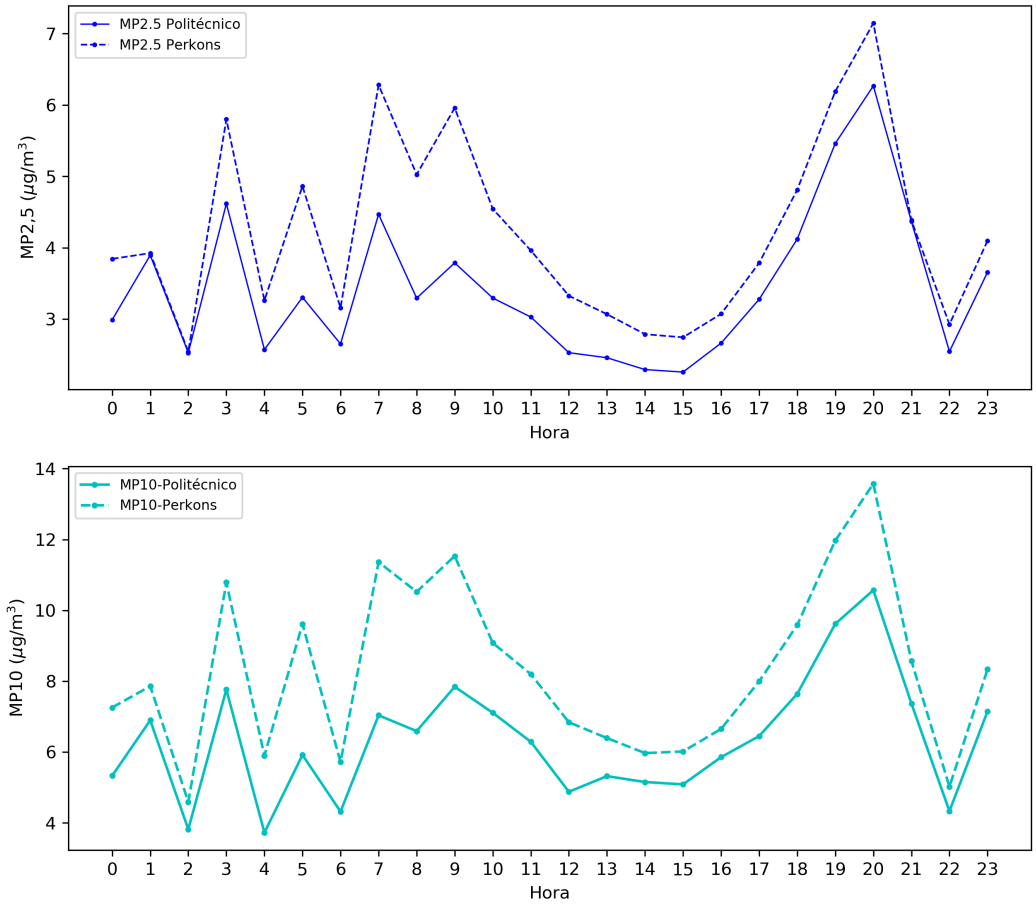

Figura 17 – Perfil diário material particulado. Fonte: O Autor (2021).

As variáveis meteorológicas possuem notável importância na análise da concentração de poluentes. A radiação solar e a velocidade do vento interferem diretamente na dispersão e dinâmica da atmosfera. Enquanto a radiação solar influencia na dispersão de poluentes por se tratar de uma forçante térmica, a velocidade do vento influencia na dispersão pela forçante mecânica.

Para analisar essa relação entre essas variáveis e o MP, a radiação solar e a velocidade do vento foram incluídas separadamente no perfil diário mostrado anteriormente. O pico da radiação solar (aproximadamente 2300 kJ h^{-1} m^{-2}) próximo de 14:00 horas, corresponde ao intervalo em que ocorrem as menores concentrações de MP, cerca de 3 μg/m^3 para o $MP_{2,5}$ e 6 μg/m^3 para o MP_{10} (Figura 18). E no intervalo em que houve maior intensidade do vento (aproximadamente 2,5 m/s), no período da tarde, é também onde ocorrem as menores concentrações de MP (Figura 19).

Portanto, essas variações na concentração de material particulado estão diretamente ligadas ao fenômeno de dispersão atmosférica. Próximo de 12:00 horas, nota-se que a concentração de MP é reduzida. Isso ocorre pois há maior radiação solar; pela forçante térmica ocorre a dispersão dos poluentes. Outra variável que influencia nesse fenômeno é a velocidade do vento que, pela forçante mecânica, também promove a dispersão. Mas é

preciso analisar também as possíveis fontes de poluentes para explicar esse comportamento do perfil diário.

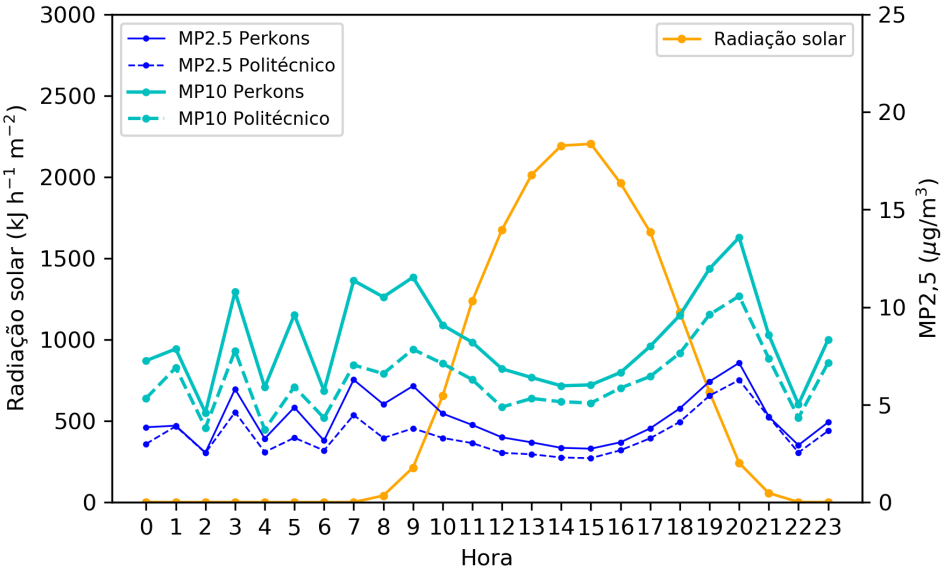

Figura 18 – Relação entre radiação solar e MP. Fonte: O Autor (2021).

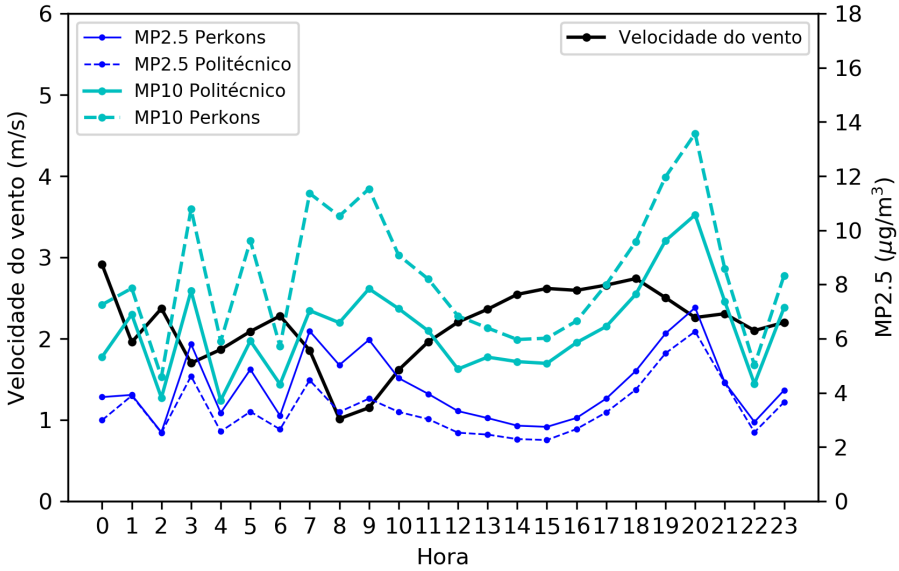

Figura 19 – Relação entre velocidade do vento e MP. Fonte: O Autor (2021).

Nota-se que nos períodos em que há maior número de veículos, cerca de 140, a concentração de MP também é maior. Às 8:00 horas, por exemplo, quando havia fluxo de 120 veículos aproximadamente, a concentração de $MP_{2,5}$ era de aproximadamente 6 $\mu g/m^3$ e a de MP_{10} de 11 $\mu g/m^3$. Às 18:00 horas, quando havia fluxo de 150 veículos

aproximadamente, a concentração de $MP_{2,5}$ era de 7 $\mu g/m^3$ e de MP_{10} era de 13 $\mu g/m^3$. De maneira geral, quando há menor quantidade de veículos, também ocorrem menores concentrações de material particulado (Figuras 20 e 21).

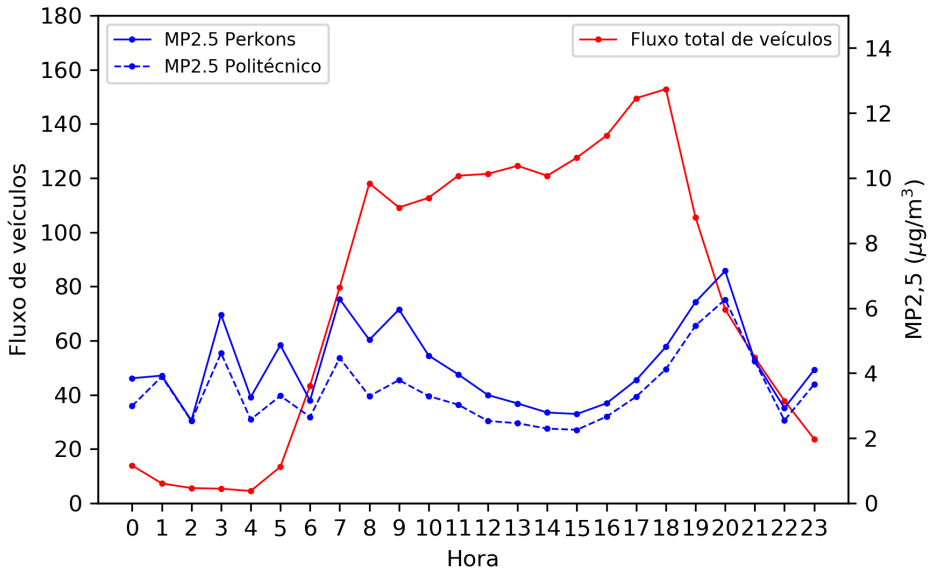

Figura 20 – Concentração de $MP_{2,5}$ e fluxo de veículos. Fonte: O Autor (2021).

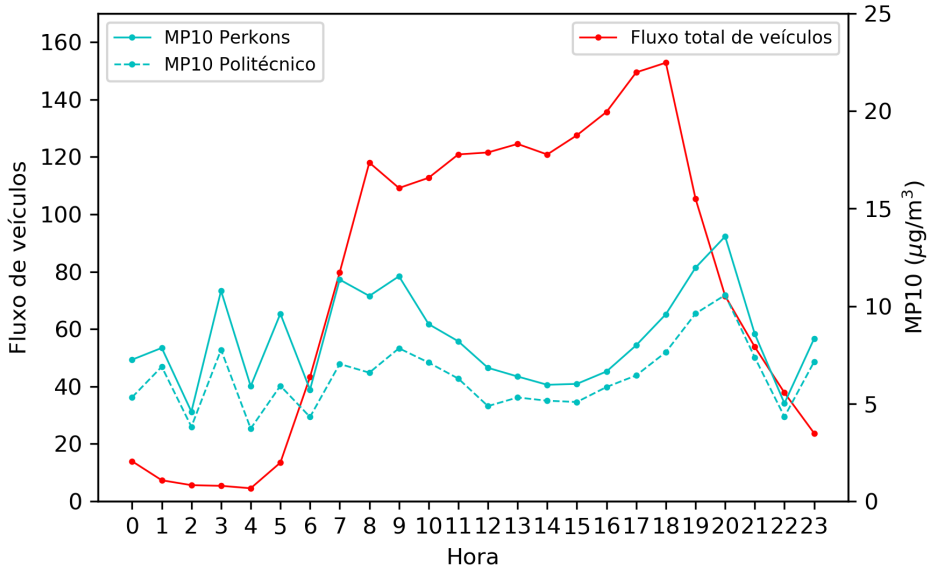

Figura 21 – Concentração de MP_{10} e fluxo de veículos. Fonte: O Autor (2021).

4.1 Resultados Random Forest

O algoritmo estudado fornece ao usuário vários dados: erro médio do *baseline* ($\mu g/m^3$), o erro médio absoluto ($\mu g/m^3$), acurária do algoritmo e importâncias de cada descritor. O erro médio do baseline ($\mu g/m^3$) representa a distância entre a previsão dos dados de MP_{10} e os dados históricos fornecidos. O erro médio absoluto ($\mu g/m^3$) representa a média das diferenças absolutas entre o valor previsto e o valor real. A acurácia representa a porcentagem de acerto da previsão e as importâncias representam o quanto cada variável foi necessária para encontrar a RF final. Para encontrar a acurácia é feito 100 menos a mape(*mean absolute percentage error*), onde a mape equivale à 100 menos a razão entre os erros e os dados de teste.

		BASE HORÁRIA	**BASE DIÁRIA**
sem baseline	Erro médio absoluto ($\mu g/m^3$)	4,09	4,40
	Acurácia (%)	18,69	28,96
com baseline	Erro médio do *baseline* ($\mu g/m^3$)	4,26	2,81
	Erro médio absoluto ($\mu g/m^3$)	3,12	2,60
	Acurácia (%)	44,40	58,47
com baseline e MP_{10} Perkons	Erro médio do *baseline* ($\mu g/m^3$)	4,26	2,81
	Erro médio absoluto ($\mu g/m^3$)	1,94	1,67
	Acurácia (%)	76,13	77,95

Tabela 2 – Cenários *Random Forest* e resultados. Fonte: O Autor (2021).

Diversas configurações foram testadas até serem encontradas as que apresentavam melhor acurácia. O modelo ainda pode ser modificado, utilizando refinamentos no código e novos conjuntos de dados para aperfeiçoar o resultado obtido. Os resultados das seis configurações feitas para analisar a performance do *Random Forest* estão apresentados na Tabela 2.

Os cenários foram enumerados para facilitar a identificação:

1. Base horária sem *baseline* como descritor
2. Base diária sem *baseline* como descritor
3. Base horária com *baseline* como descritor
4. Base diária com *baseline* como descritor
5. Base horária com *baseline* e MP10 Perkons como descritores
6. Base diária com *baseline* e MP10 Perkons como descritores

A ordem crescente dos cenários considerando a acurácia é 1 (18,69%)- 2 (28,96%) - 3 (44,40%) - 4 (58,47%) - 5 (76,13%) - 6 (77,95%). Portanto, o cenário que apresentou a melhor previsão do MP_{10} do Politécnico foi o 6, que utiliza a base diária com *baseline* e MP_{10} da Perkons como descritores. O pior cenário encontrado foi o 1, que utiliza a

base horária sem *baseline* como descritor. Em todos os cenários, o erro médio absoluto encontrado foi menor do que o erro médio do *baseline*, em $\mu g/m^3$, portanto, a previsão feita pelo algoritmo é melhor do que a base histórica fornecida como descritor.

À medida que a acurácia do modelo foi aumentando, o número de variáveis com maior importância foi diminuindo, até chegar no cenário 6, em que o descritor mais importante para o RF foi o MP_{10} medido pela Perkons, com 80%. Esse resultado era o esperado, já que, como mencionado anteriormente, os dados entre esses dois sensores de monitoramento se assemelham.

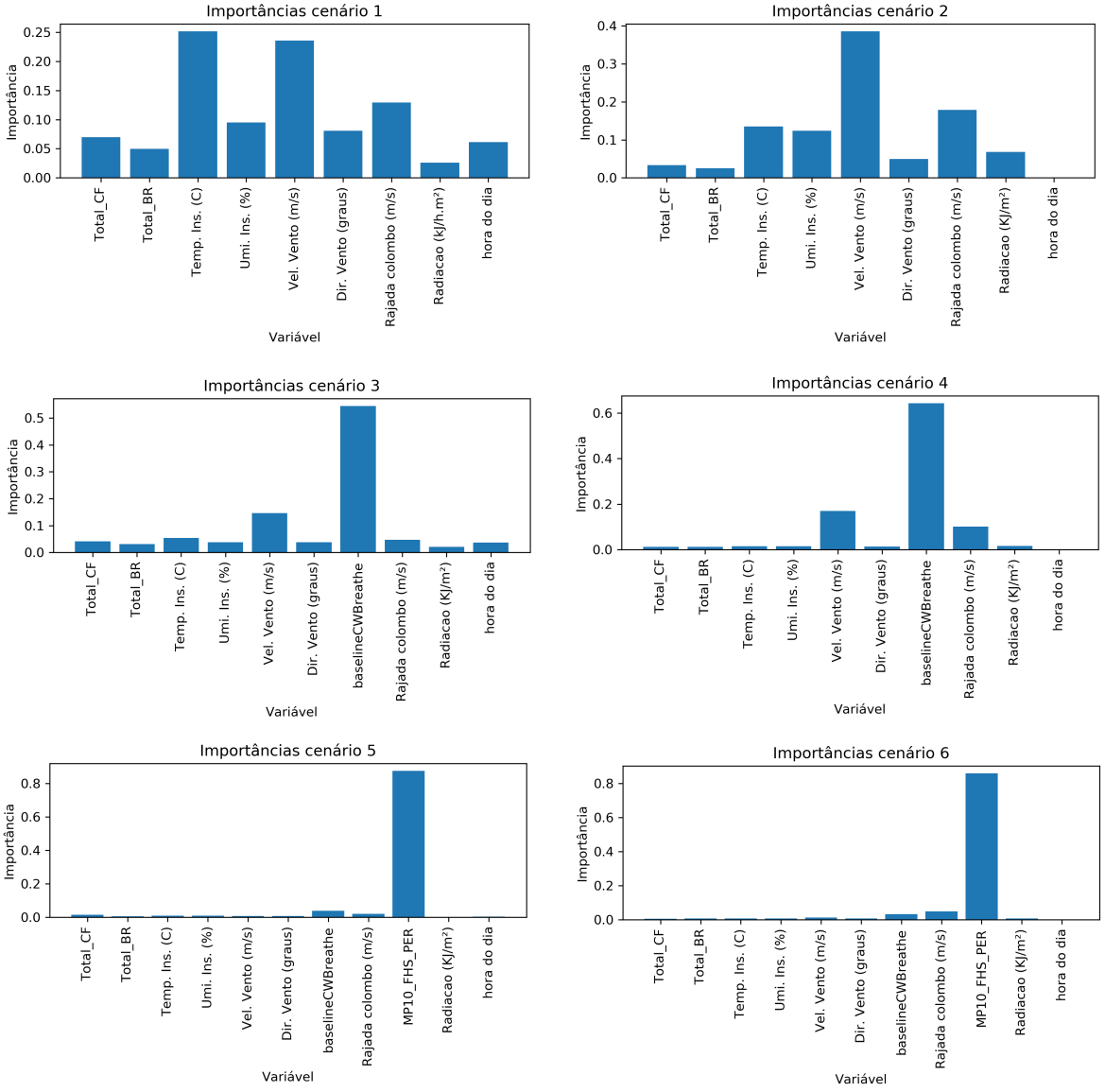

Figura 22 – Cenários do *Random Forest* e importâncias. Fonte: O Autor (2021).

Na Figura 22 nota-se que, quando não foi usado o *baseline*, as variáveis com maior importância foram a temperatura instantênea (°C), velocidade do vento (m/s) e rajada de vento medida na estão de Colombo, em m/s (cenários 1 e 2). Nos cenários 3 e 4, utilizando

o *baseline* sem o MP_{10} da Perkons, o descritor com maior importância (aproximadamente 55%) foi o *baseline*. Nos cenários 5 e 6, em que foi usado o *baseline* e o MP10 da Perkons como descritores, a variável que mais impactou no resultado (aproximadamente 60%) foi o MP_{10} da Perkons.

As previsões de MP_{10} obtidas no cenário 6, pontos em vermelho, estão bem próximas dos dados reais de MP_{10} no local, linha azul tracejada. Os comportamentos dos dois conjuntos de dados, medidos e previstos, seguem o mesmo padrão; os picos e vales se encontram em intervalos de tempo bem próximos (Figura 23). No mês de janeiro de 2021, onde foram medidas baixas concentrações de MP (cerca de 3 $\mu g/m^3$), também foram previstas baixas concentrações de MP pelo RF. E em junho e meados de julho, onde foram medidas altas concentrações de MP (cerca de 45 $\mu g/m^3$), também foram encontradas altas previsões de concentração dessas partículas. Isso mostra que o algoritmo RF é considerado satisfatório para prever a concentração de MP_{10} em um ambiente urbano.

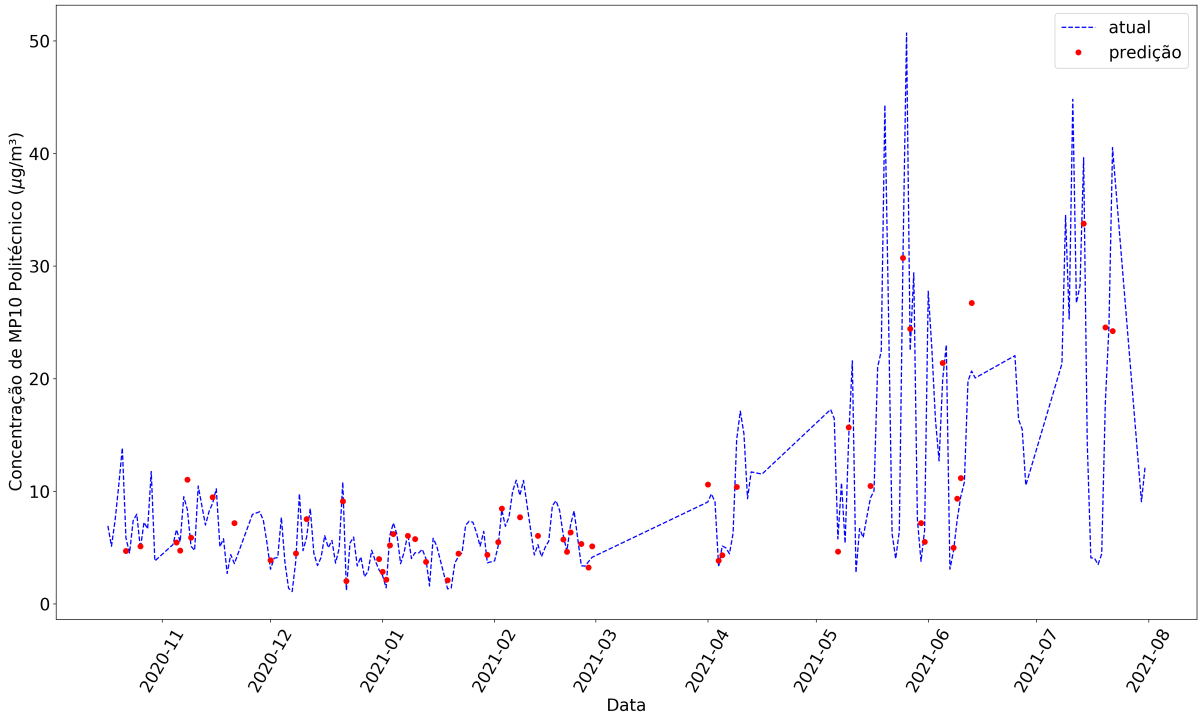

Figura 23 – Gráfico com valores atuais e predições da concentração de MP_{10} no Politécnico. Fonte: O Autor (2021).

5 Conclusão

Este trabalho teve como objetivo a previsão da concentração de MP_{10} em ambiente urbano por meio do algoritmo *Random Forest* e o método *baseline*. Seis configurações distintas foram criadas para analisar o desempenho do algoritmo. A configuração que contou com o melhor resultado (77,95%) foi a de escala temporal diária, com *baseline* e MP_{10} Perkons como descritores. Além disso, o cenário que contou com acurácia mais baixa foi o que considerou a escala temporal horária e sem o *baseline* ou MP_{10} Perkons como descritores. Portanto, a escala diária apresentou melhor desempenho em comparação com a escala horária. Quanto às importâncias dadas à cada variável, quando não foi usado o *baseline*, as variáveis com maior importância foram a temperatura instantênea (°C), velocidade do vento (m/s) e rajada de vento medida na estão de Colombo, em m/s (cenários 1 e 2). Nos cenários 3 e 4, utilizando o *baseline* sem o MP_{10} da Perkons, o descritor com maior importância (aproximadamente 55%) foi o *baseline*. Nos cenários 5 e 6, em que foi usado o *baseline* e o MP10 da Perkons como descritores, a variável que mais impactou no resultado (aproximadamente 60%) foi o MP_{10} da Perkons. Verificou-se ainda que há correlação positiva entre a concentração medida pelo SDS011 no Politécnico e na lombada eletrônica da empresa Perkons, portanto o sensor SDS011 pode ser considerado satisfatório para medição dessas partículas. Em relação ao fluxo de veículos, notou-se que este está diretamente relacionado à concentração de material particulado em via urbana. Nos períodos em que havia maior número de veículos, a concentração de MP também era maior. Às 8:00 horas, por exemplo, quando havia fluxo de 120 veículos por hora aproximadamente, a concentração horária de $MP_{2,5}$ era de aproximadamente 6 $\mu g/m^3$ e a de MP_{10} de 11 $\mu g/m^3$.

5.1 Recomendações Futuras

O algoritmo *Random Forest* pode ser usado não apenas para prever a concentração de MP em um ponto de monitoramento da qualidade do ar, mas para prever essa concentração em um bairro inteiro de Curitiba ou até mesmo em toda a cidade. Dessa forma, os tomadores de decisão terão acesso aos dados de poluição atmosférica antecipadamente e assim poderão pensar em soluções para esse problema ambiental antes mesmo de ocorrer. Outro estudo relacionado que pode ser desenvolvido é o de regressão linear múltipla. Na regressão simples, usa-se apenas uma variável preditora e na múltipla podem ser usadas diversas variáveis.

Referências

AQICN. *THE SDS011 Air Quality Sensor experiment*. 2021. Disponível em: <https://aqicn.org/sensor/sds011/>.

BAIRD, C. *Química ambiental*. [S.l.]: Reverté, 2018.

BRASIL. *Resolução nº 491, de 19 de novembro de 2018. Dispõe sobre padrões de qualidade do ar*. [S.l.]: Diário Oficial da União Brasília, 2018.

CHAVES, M. G. D. *Viabilidade do sensor de baixo custo SDS011 para monitoramento de material particulado*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2021.

FRANCO, V. et al. Road vehicle emission factors development: A review. *Atmospheric Environment*, Elsevier, v. 70, p. 84–97, 2013.

FRIEDMAN, J. H. *The elements of statistical learning: Data mining, inference, and prediction*. [S.l.]: springer open, 2017.

GENUER, R.; POGGI, J.-M.; TULEAU-MALOT, C. Variable selection using random forests. *Pattern recognition letters*, Elsevier, v. 31, n. 14, p. 2225–2236, 2010.

GOUVEIA, N. et al. Ambient fine particulate matter in latin american cities: Levels, population exposure, and associated urban factors. *Science of The Total Environment*, Elsevier, v. 772, p. 145035, 2021.

IAP. *Relatório Anual da Qualidade do Ar na Região Metropolitana de Curitiba Ano de 2011*. 2011. Disponível em: <http://www.iat.pr.gov.br/sites/agua-terra/arquivos_restritos/files/documento/2020-08/rel_anual_2011_qualidade_ar_v7_iap.pdf>.

INMET. *Estação: COLOMBO (B806)*. 2021. Disponível em: <https://tempo.inmet.gov.br/TabelaEstacoes/B806>.

INMET. *Estação: CURITIBA (A807)*. 2021. Disponível em: <https://tempo.inmet.gov.br/TabelaEstacoes/A807>.

IPPUC. *Consolidação de dados de oferta, demanda, sistema viário e zoneamento: relatório 3 - Pesquisas de tráfego da cordon line e da screen line*. 2017. Disponível em: <https://ippuc.org.br/visualizar.php?doc=https://admsite2013.ippuc.org.br/arquivos/documentos/D536/D536_014_BR.pdf>.

IPPUC. *Inquéritos domiciliares - bairros Curitiba*. 2017. Disponível em: <https://ippuc.org.br/visualizar.php?doc=https://admsite2013.ippuc.org.br/arquivos/documentos/D536/D536_001_BR.pdf>.

JACOBSON, M. Z. *Air pollution and global warming: history, science, and solutions*. [S.l.]: Cambridge University Press, 2012.

JEANJEAN, A. P. R. et al. Ranking current and prospective no2 pollution mitigation strategies: An environmental and economic modelling investigation in oxford street, london. *Environmental pollution*, Elsevier, v. 225, p. 587–597, 2017.

KAM, W. et al. Size-segregated composition of particulate matter (pm) in major roadways and surface streets. *Atmospheric environment*, Elsevier, v. 55, p. 90–97, 2012.

KAMIŃSKA, J. A. The use of random forests in modelling short-term air pollution effects based on traffic and meteorological conditions: a case study in wrocław. *Journal of environmental management*, Elsevier, v. 217, p. 164–174, 2018.

KOEHRSEN, W. *Random Forest in Python*. 2017. Disponível em: <https://towardsdatascience.com/random-forest-in-python-24d0893d51c0>.

KOEHRSEN, W. *Random Forest Simple Explanation*. 2017. Disponível em: <https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d>.

KUMAR, D. et al. Evolving differential evolution method with random forest for prediction of air pollution. *Procedia computer science*, Elsevier, v. 132, p. 824–833, 2018.

LACTEA. *Relatório técnico do Monitoramento da Qualidade do Ar em Curitiba e região metropolitana*. 2021. Disponível em: <http://www.lactea.ufpr.br/relatorios/Relatorio-LACTEA-Ar-2021-05.pdf>.

London Govern UK. *Monitoring and predicting air pollution*. 2021. Disponível em: <https://www.london.gov.uk/what-we-do/environment/pollution-and-air-quality/monitoring-and-predicting-air-pollution>.

PANT, P. et al. Characterization of traffic-related particulate matter emissions in a road tunnel in birmingham, uk: Trace metals and organic molecular markers. *Aerosol and Air Quality Research*, Taiwan Association for Aerosol Research, v. 17, n. 1, p. 117–130, 2017.

PARK, S. et al. Estimation of ground-level particulate matter concentrations through the synergistic use of satellite observations and process-based models over south korea. *Atmospheric Chemistry and Physics*, Copernicus GmbH, v. 19, n. 2, p. 1097–1113, 2019.

POMÁZI, I. Oecd environmental outlook to 2050. the consequences of inaction. *Hungarian Geographical Bulletin*, v. 61, n. 4, p. 343–345, 2012.

Prefeitura de Curitiba. *CEPAC – Características da área*. 2021. Disponível em: <https://www.curitiba.pr.gov.br/conteudo/cepac-caracteristicas-da-area/578>.

RITCHIE, H.; ROSER, M. *Our World in Data*. 2018. Disponível em: <https://ourworldindata.org/urbanization?source=:so:li:or:awr:ohcm>.

SHOGRKHODAEI, S. Z.; RAZAVI-TERMEH, S. V.; FATHNIA, A. Spatio-temporal modeling of pm2. 5 risk mapping using three machine learning algorithms. *Environmental Pollution*, Elsevier, v. 289, p. 117859, 2021.

SMIT, R.; NTZIACHRISTOS, L.; BOULTER, P. Validation of road vehicle and traffic emission models–a review and meta-analysis. *Atmospheric environment*, Elsevier, v. 44, n. 25, p. 2943–2953, 2010.

SRIMURUGANANDAM, B.; NAGENDRA, S. M. S. Analysis and interpretation of particulate matter–pm10, pm2. 5 and pm1 emissions from the heterogeneous traffic near an urban roadway. *Atmospheric Pollution Research*, Elsevier, v. 1, n. 3, p. 184–194, 2010.

STULL, R. B. et al. Practical meteorology: an algebra-based survey of atmospheric science. 2015.

TEIXEIRA, E. C.; FELTES, S.; SANTANA, E. R. R. d. Estudo das emissões de fontes móveis na região metropolitana de porto alegre, rio grande do sul. *Química Nova*, SciELO Brasil, v. 31, p. 244–248, 2008.

World Health Organization. Ambient air pollution: A global assessment of exposure and burden of disease. World Health Organization, 2016.

YU, R. et al. Raq–a random forest approach for predicting air quality in urban sensing systems. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 16, n. 1, p. 86, 2016.

ZHU, Y. et al. Concentration and size distribution of ultrafine particles near a major highway. *Journal of the air & waste management association*, Taylor & Francis, v. 52, n. 9, p. 1032–1042, 2002.

ZIMMERMAN, N. et al. A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring. *Atmospheric Measurement Techniques*, Copernicus GmbH, v. 11, n. 1, p. 291–313, 2018.

6 Códigos em Python

6.1 Perfis diários e séries temporais dos dados

```
import numpy as np
from os import system
from urllib.request import urlretrieve
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib import cm
from matplotlib import rc
import matplotlib
from matplotlib.colors import ListedColormap, LinearSegmentedColormap
import datetime
from datetime import datetime
from matplotlib.dates import DayLocator, HourLocator, DateFormatter, drange
import matplotlib.dates as mdates
from IPython.display import display, Markdown
# Pandas is used for data manipulation
import pandas as pd

# Read in data and display first 5 rows '2020-10-17 00:00:00'
dateparse2 = lambda x: datetime.strptime(x, '\%Y-\%m-\%d \%H:\%M:\%S')
COMPLETO = pd.read_csv('completo3.csv', sep=',', index_col = 0,
parse_dates=['Date'], date_parser=dateparse2)

dateparse = lambda x: datetime.strptime(x, '\%d/\%m/\%Y \%H')
INMET = pd.read_csv('CURITIBA (A807)_atualizado.aux', sep=',',
 index_col = 0, parse_dates=['Data Hora (UTC)'], date_parser=dateparse)
df1 = pd.read_csv('novos_dados_INMET/
dados_83842_H_2020-01-01_2021-07-24.csv',sep=';',skiprows=9)
curitiba = df1.copy()

df2 = pd.read_csv('novos_dados_INMET/
dados_B806_H_2020-01-01_2021-07-31.csv',sep=';',skiprows=9)
colombo = df2.copy()
```

```
datamedicao1 = curitiba['Data Medicao'].values
horamedicao1 = curitiba['Hora Medicao'].values

datamedicao2 = colombo['Data Medicao'].values
horamedicao2 = colombo['Hora Medicao'].values

datahorario1 = []
for i in range(len(datamedicao1)):
    datahorario1.append(datamedicao1[i] + ' ' + str(int(horamedicao1[i]/100)))

datahorario2 = []
for i in range(len(datamedicao2)):
    datahorario2.append(datamedicao2[i] + ' ' + str(int(horamedicao2[i]/100)))

lista = []
for i in range(len(datahorario1)):
    lista.append(datetime.strptime(datahorario1[i], '%Y-%m-%d %H'))

lista2 = []
for i in range(len(datahorario2)):
    lista2.append(datetime.strptime(datahorario2[i], '%Y-%m-%d %H'))

curitiba['data'] = lista
del curitiba['Data Medicao']
del curitiba['Hora Medicao']
curitiba = curitiba.set_index('data')

colombo['data'] = lista2
del colombo['Data Medicao']
del colombo['Hora Medicao']
colombo = colombo.set_index('data')

INMET.index=pd.to_datetime(INMET.index)

COMPLETO['Vel. Vento (m/s)'] = INMET['Vel. Vento (m/s)']
COMPLETO['Radiacao (KJ/m²)'] = INMET['Radiacao (KJ/m²)']

COMPLETO['TOTAL_VEICULOS'] = COMPLETO['TOTAL_FLUXO_BR'] +
```

```
COMPLETO['TOTAL_FLUXO_CF']
COMPLETO['PEQUENOS_VEICULOS'] = COMPLETO['Pequenos_BR'] +
COMPLETO['Pequenos_CF']
COMPLETO['MEDIOS_VEICULOS'] = COMPLETO['Medios_BR'] +
COMPLETO['Medios_CF']
COMPLETO['GRANDES_VEICULOS'] = COMPLETO['Grandes_BR'] +
 COMPLETO['Grandes_CF']

COMPLETO = COMPLETO.dropna()
COMPLETO.to_csv('COMPLETO.csv')

TOTAL_VEICULOS = COMPLETO['TOTAL_VEICULOS']
PEQUENOS_VEICULOS = COMPLETO['PEQUENOS_VEICULOS']
MEDIOS_VEICULOS = COMPLETO['MEDIOS_VEICULOS']
GRANDES_VEICULOS = COMPLETO['GRANDES_VEICULOS']
FLUXO_BR = COMPLETO['TOTAL_FLUXO_BR']
FLUXO_CF = COMPLETO['TOTAL_FLUXO_CF']
PEQUENOS_CF = COMPLETO['Pequenos_CF']
MEDIOS_CF = COMPLETO['Medios_CF']
GRANDES_CF = COMPLETO['Grandes_CF']
PEQUENOS_BR = COMPLETO['Pequenos_BR']
MEDIOS_BR = COMPLETO['Medios_BR']
GRANDES_BR = COMPLETO['Grandes_BR']
MP25_PER = COMPLETO['MP25_PER']
MP10_PER = COMPLETO['MP10_PER']
MP25_POLI = COMPLETO['MP25_POLI']
MP10_POLI = COMPLETO['MP10_POLI']

plt.figure(figsize = (10, 5))

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('\%Y-\%m'))

plt.tick_params(axis='both', labelsize=8) #increase font size for ticks

plt.gcf().autofmt_xdate()

plt.plot(INMET['Vel. Vento (m/s)'],color='black', marker='o',linewidth=0.5,
markersize=0.5)
```

```
plt.tick_params(axis='both', labelsize=10) #increase font size for ticks

ax2 = plt.axes()
ax2.xaxis.set_major_locator(plt.MultipleLocator(31))
plt.xlabel('Data', fontsize=10)
plt.ylabel('Velocidade do vento (m/s)', fontsize=10) #y label
plt.savefig('graficos/vel.vento.png', dpi = 300)
plt.show()
selecao = INMET['2021-01-20 00:00:00':'2021-01-24 00:00:00']

plt.figure(figsize=(10,5))
plt.plot(selecao['Radiacao (KJ/m²)'],color='orange', marker='o',
linewidth=0.8, markersize=1.0)
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
#plt.plot(selecao['MP10_POLI'],color='GREY', marker='o',linewidth=0.5,
markersize=0.5, label="MP10- CWBREATHE")
#plt.xticks(rotation=30)
plt.ylim(-150,3100)
ax2 = plt.axes()
ax2.xaxis.set_major_locator(plt.MultipleLocator(1))
plt.xticks(rotation=30)
plt.xlabel('Dia', fontsize=10) #x label
plt.ylabel('Radiação solar (kJh^{-1}.m^{-2})', fontsize=10) #y label
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0.,
frameon=False)
plt.savefig('graficos/radsolardias.png', dpi = 300, bbox_inches='tight')
plt.show()

df=pd.read_csv('COMPLETO.csv',sep=',')
result = pd.DataFrame()
df['Date'] = pd.to_datetime(df['Date'])
hour = pd.to_timedelta(df['Date'].dt.hour, unit='H')
#print(hour)
perfil_diario = df.groupby(hour).mean()
result = pd.concat([result, perfil_diario], axis=1, sort=False)
print('Data/horário início : ',df.index[0])
print('Data/horário final  : ',df.index[-1])
horario = np.linspace(0, 23, 24)
print(horario)
```

```
plt.figure(figsize=(10,5))
plt.plot(horario,result['TOTAL_FLUXO_BR'],color='purple', marker='o',
linewidth=0.7, markersize=0.7, label="Fluxo sentido BR-277")
plt.plot(horario,result['TOTAL_FLUXO_CF'],color='red', marker='o',
linewidth=0.7, markersize=0.7, label="Fluxo sentido Av. Com. Franco")
plt.legend(loc='best')
plt.xticks(horario)
plt.ylim(0,100)
plt.xlabel('Hora', fontsize=10) #x label
plt.ylabel(r'Fluxo de veículos', fontsize=10) #y label
plt.legend(loc ='upper left',fontsize=8)
plt.savefig('graficos/fluxometodos.png', dpi = 300, bbox_inches='tight')
plt.show()

plt.figure(figsize=(10,5))
plt.plot(horario,result['TOTAL_VEICULOS'],color='brown', marker='o',
linewidth=1.5, markersize=2.5, label="Fluxo total")
plt.plot(horario,result['PEQUENOS_VEICULOS'],color='orange', marker='o',
linestyle='dotted',linewidth=1.5, markersize=2.5, label="Veículos pequenos")
plt.plot(horario,result['MEDIOS_VEICULOS'],color='yellow', marker='o',
linestyle='dashed',linewidth=1.5, markersize=2.5, label="Veículos médios")
plt.plot(horario,result['GRANDES_VEICULOS'],color='black', marker='o',
linestyle='dashdot',linewidth=1.5, markersize=2.5, label="Veículos grandes")
plt.legend(loc='upper left',fontsize=8)
plt.xticks(horario)
plt.ylim(0,170)
plt.xlabel('Hora', fontsize=10) #x label
plt.ylabel(r'Fluxo de veículos', fontsize=10) #y label
plt.savefig('graficos/fluxocomtipos.png', dpi = 300, bbox_inches='tight')
plt.show()

plt.figure(figsize=(10,5))
plt.plot(horario,result['GRANDES_VEICULOS'],color='black', marker='o',
linestyle='dashdot',linewidth=1.5, markersize=2.5, label="Veículos grandes")
plt.legend(loc='upper left',fontsize=8)
plt.xticks(horario)
plt.ylim(0,6)
plt.xlabel('Hora', fontsize=10) #x label
```

```
plt.ylabel(r'Fluxo de veículos', fontsize=10)
plt.savefig('graficos/fluxograndes.png', dpi = 300, bbox_inches='tight')
plt.show()

#### gráfico MP25_PERKONS
plt.figure(figsize = (10,10))

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))

plt.tick_params(axis='both', labelsize=8) #increase font size for ticks

plt.gcf().autofmt_xdate()

plt.subplot(2,1,1)
plt.xticks(horario)
plt.xlabel('Hora', fontsize=10)
plt.ylabel(r'MP2,5 ($\mu$g/m$^3$)', fontsize=10)
plt.plot(horario, result['MP25_POLI'],color='blue', marker='o',linewidth=0.8,
markersize=1.8, label="MP2.5 Politécnico")
plt.plot(horario, result['MP25_PER'],marker='o', markersize = 1.8, linewidth=1.0,
linestyle ='--', color='blue',label='MP2.5 Perkons')
plt.legend(fontsize=8)
plt.subplot(2,1,2)
plt.xticks(horario)
plt.xlabel('Hora', fontsize=10)
plt.ylabel(r'MP10 ($\mu$g/m$^3$)', fontsize=10)
plt.plot(horario, result['MP10_POLI'],color='c', marker='o',linewidth=1.5,
markersize=2.5, label="MP10-Politécnico")
plt.plot(horario, result['MP10_PER'],marker='o', markersize = 2.5, linewidth=1.5,
linestyle ='--', color='c',label='MP10-Perkons')
plt.legend(fontsize=8)

plt.savefig('graficos/2 MPs.png', dpi = 300)
plt.show()

x = COMPLETO['MP25_POLI'].values
y = COMPLETO['MP25_PER'].values

z = np.array([0, 90])
```

```
plt.plot(z,z,color='black',linewidth=0.4)
plt.gca().set_aspect('equal', adjustable='box')

plt.xlim(-1,90)
plt.ylim(-1,90)
plt.plot(x[1:],y[1:],'o',markersize = 1, markeredgewidth = 0.1)
plt.legend(loc=2, prop={'size': 8},frameon=False)
plt.xlabel('SDS011 CWBREATHE', fontsize=8)
plt.ylabel('SDS011 PERKONS', fontsize=8)
plt.savefig('graficos/CWBREATHExPER.png', dpi = 300)
plt.show()

#variáveis:
#MP25 do politecnico:           completo['MP25_POLI']
#MP10 do politecnico:           completo['MP10_POLI']
#MP25 do politecnico:           completo['MP25_FHS_PER']
#MP10 do politecnico:           completo['MP10_FHS_PER']
#fluxo total da CF:             completo['Total_CF']
#fluxo total da BR:             completo['Total_BR']
#fluxo veiculos por hora BR:    completo['fluxo_hora_BR']
#fluxo veiculos por hora CF:    completo['fluxo_hora_CF']

#radiação :                     completo['Radiacao (KJ/m²)']
# vento :                       completo['Vel. Vento (m/s)']

#fluxo veiculos pequenos da CF: completo['Pequenos_CF']
#fluxo veiculos medios da CF:   completo['Medios_CF']
#fluxo veiculos grandes da CF:  completo['Grandes_CF']
#fluxo veiculos pequenos da BR: completo['Pequenos_BR']
#fluxo veiculos medios da BR:   completo['Medios_BR']
#fluxo veiculos grandes da BR:  completo['Grandes_BR']

#### gráfico MP25_PERKONS
plt.figure(figsize = (15, 10))

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))

plt.tick_params(axis='both', labelsize=8) #increase font size for ticks
```

```
plt.gcf().autofmt_xdate()

plt.subplot(2,2,1)
plt.xticks(horario)
plt.ylabel(r'MP2,5 - CWBREATHE ($\mu$g/m$^3$)', fontsize=10)
plt.plot(horario, result['MP25_POLI'],color='blue', marker='o',linewidth=0.8,
markersize=1.8, label="MP2.5 Cwbreathe")
plt.subplot(2,2,2)
plt.xticks(horario)
plt.ylabel(r'MP2,5 - PERKONS ($\mu$g/m$^3$)', fontsize=10)
plt.plot(horario, result['MP25_PER'],marker='o', markersize = 1.8,
linewidth=0.8, color='blue',label='MP2.5 Perkons')
plt.subplot(2,2,3)
plt.xticks(horario)
plt.ylabel(r'MP10 - CWBREATHE ($\mu$g/m$^3$)', fontsize=10)
plt.plot(horario, result['MP10_POLI'],color='c', marker='o',
linewidth=0.8, markersize=1.8, label="MP10 Cwbreathe")
plt.subplot(2,2,4)
plt.xticks(horario)
plt.ylabel(r'MP10 - PERKONS ($\mu$g/m$^3$)', fontsize=10)
plt.plot(horario, result['MP10_PER'],marker='o', markersize = 1.8,
 linewidth=0.8, color='c',label='MP10 Perkons')

plt.savefig('graficos/4 MPs.png', dpi = 300)
plt.show()

figsize = (15, 10)
fig, ax1 = plt.subplots()

ax1.set_xlabel('Hora')
ax1.set_ylabel('Fluxo de veículos', color='black')
ax1.plot(horario, result['TOTAL_VEICULOS'],
label = 'Fluxo total de veículos', color='red', marker='o',
markersize = 1.8, linewidth=0.8)
ax1.tick_params(axis='y', labelcolor='black')
ax1.set_ylim(0,180)
plt.legend(loc ='upper right',fontsize=8)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis
```

```
ax2.plot(horario, result['MP25_PER'], label = 'MP2.5 Perkons', color='blue',
marker='o', markersize = 1.8, linewidth=0.8)
ax2.plot(horario, result['MP25_POLI'], label = 'MP2.5 Politécnico', color='blue',
marker='o', markersize = 1.8, linestyle ='--',linewidth=0.8)
ax2.tick_params(axis='y', labelcolor='black')
ax2.set_ylim(0,15)
plt.xticks(horario)
plt.xlabel('Hora', fontsize=10)
plt.ylabel(r'MP2,5 ($\mu$g/m$^3$)', color='black',fontsize=10)

plt.legend(loc ='upper left',fontsize=8)
plt.savefig('graficos/fluxo e MP2,5.png', dpi = 300)
fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()

plt.figure(figsize = (15, 10))
fig, ax1 = plt.subplots()

ax1.set_xlabel('Hora')
ax1.set_ylabel('Fluxo de veículos', color='black')
ax1.plot(horario, result['TOTAL_VEICULOS'], label = 'Fluxo total de veículos',
color='red', marker='o', markersize = 1.8, linewidth=0.8)
ax1.tick_params(axis='y', labelcolor='black')
ax1.set_ylim(0,170)
plt.legend(loc ='upper right',fontsize=8)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

ax2.plot(horario, result['MP10_PER'], label = 'MP10 Perkons',
color='c', marker='o', markersize = 1.8, linewidth=0.8)
ax2.plot(horario, result['MP10_POLI'], label = 'MP10 Politécnico',
color='c', marker='o', markersize = 1.8, linestyle ='--',linewidth=0.8)
ax2.tick_params(axis='y', labelcolor='black')
ax2.set_ylim(0,25)
plt.xticks(horario)
plt.xlabel('Hora', fontsize=10)
plt.ylabel(r'MP10 ($\mu$g/m$^3$)', color='black',fontsize=10)
```

```
plt.legend(loc ='upper left',fontsize=8)
plt.savefig('graficos/fluxo e MP10.png', dpi = 300)
fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()

fig, ax1 = plt.subplots()

ax1.set_xlabel('Hora')
ax1.set_ylabel('Radiação solar (kJ h$^{-1}$ m$^{-2}$)', color='black')
ax1.plot(horario, result['Radiacao (KJ/m²)'], label = 'Radiação solar',
color='orange', marker='o', markersize = 2.8, linewidth=1.2)
ax1.tick_params(axis='y', labelcolor='black')
ax1.set_ylim(0,3000)
plt.legend(loc ='upper right',fontsize=8)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

ax2.plot(horario, result['MP25_PER'], label = 'MP2.5 Perkons', color='blue',
marker='o', markersize = 1.8, linewidth=0.8)
ax2.plot(horario, result['MP25_POLI'], label = 'MP2.5 Politécnico', color='blue',
marker='o', markersize = 1.8, linestyle ='--',linewidth=0.8)
plt.plot(horario, result['MP10_PER'],marker='o', markersize = 2.5, linewidth=1.5,
color='c',label='MP10 Perkons')
plt.plot(horario, result['MP10_POLI'],color='c', marker='o', linestyle ='--',
linewidth=1.5, markersize=2.5, label="MP10 Politécnico")
ax2.tick_params(axis='y', labelcolor='black')
ax2.set_ylim(0,25)

plt.xticks(horario)
plt.xlabel('Hora', fontsize=10)
plt.ylabel(r'MP ($\mu$g/m$^3$)', color='black',fontsize=10)

plt.legend(loc ='upper left',fontsize=8)
plt.savefig('graficos/radiação e MP.png', dpi = 300)
fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()

fig, ax1 = plt.subplots()
```

```
ax1.set_xlabel('Hora')
ax1.set_ylabel('Velocidade do vento (m/s)', color='black')
ax1.plot(horario, result['Vel. Vento (m/s)'], label = 'Velocidade do vento', color=
marker='o', markersize = 2.8, linewidth=1.5)
ax1.tick_params(axis='y', labelcolor='black')
ax1.set_ylim(0,6)
plt.legend(loc ='upper right',fontsize=8)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

ax2.plot(horario, result['MP25_PER'], label = 'MP2.5 Perkons', color='blue',
marker='o', markersize = 1.8, linewidth=0.8)
ax2.plot(horario, result['MP25_POLI'], label = 'MP2.5 Politécnico', color='blue',
marker='o', markersize = 1.8, linestyle ='--',linewidth=0.8)
plt.plot(horario, result['MP10_POLI'],color='c', marker='o',linewidth=1.5,
markersize=2.5, label="MP10 Politécnico")
plt.plot(horario, result['MP10_PER'],marker='o', markersize = 2.5, linewidth=1.5,
linestyle ='--', color='c',label='MP10 Perkons')
ax2.tick_params(axis='y', labelcolor='black')
ax2.set_ylim(0,18)

plt.xticks(horario)
plt.xlabel('Hora', fontsize=10)
plt.ylabel(r'MP ($\mu$g/m$^3$)', color='black',fontsize=10)

plt.legend(loc ='upper left', fontsize=8)
plt.savefig('graficos/vento e MP.png', dpi = 300)
fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()

plt.figure(figsize=(10,5))
plt.plot(horario,result['Radiacao (KJ/m²)'],color='orange', marker='o',
linewidth=1.5, markersize=2.5)
plt.xticks(horario)
plt.ylim(0,2400)
plt.xlabel('Hora', fontsize=10) #x label
plt.ylabel(r'Radiação solar (kJ h$^{-1}$ m$^{-2}$)', fontsize=10) #y label
plt.savefig('graficos/perfildiarioradsolar.png', dpi = 300, bbox_inches='tight')
plt.show()
```

```
import numpy as np
from os import system
from urllib.request import urlretrieve
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import matplotlib
import datetime
from datetime import datetime
from matplotlib.dates import DayLocator, HourLocator, DateFormatter, drange
import matplotlib.dates as mdates
from IPython.display import display, Markdown
# Pandas is used for data manipulation
import pandas as pd

dateparse2 = lambda x: datetime.strptime(x, '\%Y-\%m-\%d \%H:\%M:\%S')
COMPLETO = pd.read_csv('completo3.csv', sep=',', index_col = 0,
parse_dates=['Date'], date_parser=dateparse2)

dateparse = lambda x: datetime.strptime(x, '\%d/\%m/\%Y \%H')
INMET = pd.read_csv('CURITIBA (A807)_atualizado.aux', sep=',', index_col = 0,
parse_dates=['Data Hora (UTC)'], date_parser=dateparse)

df1 = pd.read_csv('novos_dados_INMET/
dados_83842_H_2020-01-01_2021-07-24.csv',sep=';',skiprows=9)
curitiba = df1.copy()

df2 = pd.read_csv('novos_dados_INMET/
dados_B806_H_2020-01-01_2021-07-31.csv',sep=';',skiprows=9)
colombo = df2.copy()

datamedicao1 = curitiba['Data Medicao'].values
horamedicao1 = curitiba['Hora Medicao'].values

datamedicao2 = colombo['Data Medicao'].values
horamedicao2 = colombo['Hora Medicao'].values

datahorario1 = []
for i in range(len(datamedicao1)):
```

```
    datahorario1.append(datamedicao1[i] + ' ' + str(int(horamedicao1[i]/100)))

datahorario2 = []
for i in range(len(datamedicao2)):
    datahorario2.append(datamedicao2[i] + ' ' + str(int(horamedicao2[i]/100)))

lista = []
for i in range(len(datahorario1)):
    lista.append(datetime.strptime(datahorario1[i], '\%Y-\%m-\%d \%H'))

lista2 = []
for i in range(len(datahorario2)):
    lista2.append(datetime.strptime(datahorario2[i], '\%Y-\%m-\%d \%H'))

curitiba['data'] = lista
del curitiba['Data Medicao']
del curitiba['Hora Medicao']
curitiba = curitiba.set_index('data')

colombo['data'] = lista2
del colombo['Data Medicao']
del colombo['Hora Medicao']
colombo = colombo.set_index('data')

arqs=['Dados_solicitados/BATEL.csv',
      'Dados_solicitados/BOTANICO.csv',
      'Dados_solicitados/BVISTA.csv',
      'Dados_solicitados/GUAIRA.csv',
      'Dados_solicitados/MERCES.csv',
      'Dados_solicitados/ORLEANS.csv']

colunas=['MP10',
         'MP10',
         'MP10',
         'MP10',
         'MP10',
         'MP10']

result = pd.DataFrame()
```

```
result2 = pd.DataFrame()
resultdiario = pd.DataFrame()

for k in range(len(arqs)):
    nome=arqs[k]
    nome=nome[:-4]
    print(nome)
    dt=pd.read_csv(nome+".csv",sep=';',index_col=0)

    dt.index=pd.to_datetime(dt.index)
    conc_media_hora=dt.resample('h').mean()
    result = pd.concat([result, conc_media_hora['MP10']], axis=1, sort=False)

    print('Data/horário início : ',dt[colunas[k]].index[0])
    print('Data/horário final  : ',dt[colunas[k]].index[-1])

baseline_mediasCWBreathe = result.mean(axis=1)

from datetime import date, timedelta

datas = pd.date_range('2020-10-17 00:00:00','2021-08-15 23:00:00',freq='h')

features = pd.DataFrame()
features['datas'] = datas

features.index=pd.to_datetime(features['datas'])
features

features['Total_CF'] = COMPLETO['TOTAL_FLUXO_CF']
features['Total_BR'] = COMPLETO['TOTAL_FLUXO_BR']
features['Temp. Ins. (C)'] = INMET['Temp. Ins. (C)']
features['Umi. Ins. (\%)']  = colombo['UMIDADE RELATIVA DO AR, HORARIA(\%)']
features['Vel. Vento (m/s)'] = INMET['Vel. Vento (m/s)']
features['Dir. Vento (graus)']  = INMET['Dir. Vento (m/s)']
features['Radiacao (KJ/m²)']  = INMET['Radiacao (KJ/m²)']
features['baselineCWBreathe'] = baseline_mediasCWBreathe
features['Rajada colombo (m/s)'] = colombo['VENTO, RAJADA MAXIMA(m/s)']
features['MP10_POLI'] = COMPLETO['MP10_POLI']
features['MP10_FHS_PER'] = COMPLETO['MP10_PER']
```

```
v = features.index.hour.values
mask1 = v < 8
mask1
mask2 = v > 21
mask2

lista_mask = []

for i in range(len(mask1)):
    #print(mask1[i] or mask2[i])
    lista_mask.append(mask1[i] or mask2[i])

mask3 = np.array(lista_mask)
mask3

novo = pd.DataFrame(np.where(mask3, 0, features['Radiacao (KJ/m²)']),
features.index)

novo.columns = ["Radiacao (KJ/m²)"]

del features['Radiacao (KJ/m²)']
del features['datas']

features['Radiacao (KJ/m²)'] = novo["Radiacao (KJ/m²)"]

dias_semana = features.index.day_name()
features['dia da semana'] = dias_semana

hora_do_dia = features.index.hour
features['hora do dia'] = hora_do_dia

del features['dia da semana']
print('The shape of our features is:', features.shape)

features = features.resample('d').mean()

dates = features.index
```

```
# Import matplotlib for plotting and use magic command for Jupyter Notebooks
import matplotlib.pyplot as plt

\%matplotlib inline

# Set up the plotting layout
fig, ((ax1, ax2,ax3)) = plt.subplots(nrows=3, ncols=1, figsize = (10,10))
fig.autofmt_xdate(rotation = 45)

# Actual temperature measurement
ax1.plot(dates, features['baselineCWBreathe'])
ax1.set_xlabel(''); ax1.set_ylabel('Temperature');
 ax1.set_title('Temperatura Instantânea')

ax2.plot(dates, features['MP10_FHS_PER'])
ax2.set_xlabel(''); ax2.set_ylabel('Fluxo para BR');
ax2.set_title('Fluxo de veículos para BR')

ax3.plot(dates, features['MP10_POLI'])
ax3.set_xlabel('Date'); ax3.set_ylabel('Concentração MP10 (micro g/m3)');
ax3.set_title('Concentração MP10')

plt.show()
plt.tight_layout(pad=2)

features = features.dropna()
features.describe()

# One-hot encode the data using pandas get_dummies
features = pd.get_dummies(features)

# Display the first 5 rows of the last 12 columns
features.iloc[:,5:].head(5)

features.to_csv('features_diario_combaseline_MP10P.csv')
copia_features = features

# Use numpy to convert to arrays
import numpy as np
```

```
# Labels are the values we want to predict
labels = np.array(features['MP10_POLI'])
datas_ok = features['MP10_POLI'].index

features= features.drop('MP10_POLI', axis = 1)

# Saving feature names for later use
feature_list = list(features.columns)

# Convert to numpy array
features_array = np.array(features)

# Using Skicit-learn to split data into training and testing sets
from sklearn.model_selection import train_test_split

# Split the data into training and testing sets
train_features, test_features, train_labels, test_labels =
train_test_split(features_array, labels, test_size = 0.27, random_state = 42)
datas_treinamento, datas_testes = train_test_split(features.index,
test_size = 0.27, random_state = 42)

print('Training Features Shape:', train_features.shape)
print('Training Labels Shape:', train_labels.shape)
print('Testing Features Shape:', test_features.shape)
print('Testing Labels Shape:', test_labels.shape)

# The baseline predictions are the historical averages
baseline_preds = test_features[:, feature_list.index('baselineCWBreathe')]

# Baseline errors, and display average baseline error
baseline_errors = abs(baseline_preds - test_labels)

print('Average baseline error: ', round(np.mean(baseline_errors), 2), 'micro g / m3

# Import the model we are using
from sklearn.ensemble import RandomForestRegressor

# Instantiate model with 1000 decision trees (aumentei para 1000)
```

```
rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)

# Train the model on training data
rf.fit(train_features, train_labels);

# Use the forest's predict method on the test data
predictions = rf.predict(test_features)

# Calculate the absolute errors
errors = abs(predictions - test_labels)

# Print out the mean absolute error (mae)
print('Mean Absolute Error:', round(np.mean(errors), 2), 'micro g / m3.')

# Calculate mean absolute percentage error (MAPE)
mape = 100 * (errors / test_labels)

# Calculate and display accuracy
accuracy = 100 - np.mean(mape)
print('Accuracy:', round(accuracy, 2), '\%.')

# Get numerical feature importances
importances = list(rf.feature_importances_)
# List of tuples with variable and importance
feature_importances = [(feature, round(importance, 2)) for feature,
importance in zip(feature_list, importances)]
# Sort the feature importances by most important first
feature_importances = sorted(feature_importances, key = lambda x: x[1],
reverse = True)
# Print out the feature and importances
[print('Variable: {:20} Importance: {}'.format(*pair)) for pair in
feature_importances];

# Import matplotlib for plotting and use magic command for Jupyter Notebooks
import matplotlib.pyplot as plt
\%matplotlib inline
x_values = list(range(len(importances)))
plt.bar(x_values, importances, orientation = 'vertical')
# Tick labels for x axis
```

```
plt.xticks(x_values, feature_list, rotation='vertical')
# Axis labels and title
plt.ylabel('Importância'); plt.xlabel('Variável'); plt.title('Importâncias cenário
plt.tight_layout()
plt.savefig('graficos/importanciacenario6.png', dpi = 300)

true_data = pd.DataFrame(data = {'date': datas_ok, 'actual': labels})
plt.figure(figsize=(20,8))
# Dataframe with predictions and dates
predictions_data = pd.DataFrame(data = {'date': datas_testes,
'prediction': predictions})
# Plot the actual values
#plt.plot(true_data['date'], true_data['actual'], 'b--', label = 'actual')
plt.plot(true_data['date'], true_data['actual'], 'b--', label = 'actual')
# Plot the predicted values
plt.plot(predictions_data['date'], predictions_data['prediction'], 'ro',
markersize=2, label = 'prediction')
plt.xticks(rotation = '60');
plt.legend()
# Graph labels
plt.xlabel('Date'); plt.ylabel('Concentração de MP ()');
plt.title('Actual and Predicted Values');
plt.savefig('graficos/serie_temporal_geral.png', dpi = 300)
```

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA AMBIENTAL

TERMO DE APROVAÇÃO DE PROJETO FINAL

ISADORA BERGAMI

PREVISÃO DA CONCENTRAÇÃO DE MATERIAL PARTICULADO MP10 EM AMBIENTE URBANO UTILIZANDO APRENDIZADO DE MÁQUINA

Projeto Final de Curso, aprovado como requisito parcial para a obtenção do Diploma de Bacharel em Engenharia Ambiental no Curso de Graduação em Engenharia Ambiental do Setor de Tecnologia da Universidade Federal do Paraná, com nota 93 , pela seguinte banca examinadora:

Prof. Emílio G. F. Mercuri
Dpto. de Engenharia Ambiental
Universidade Federal do Paraná
Matrícula: 203974

Orientador(a): _____
Emílio G. F. Mercuri
Departamento de Engenharia Ambiental - UFPR

Co-orientador(a): Não há.

Membro(a) 1): _____
João Pedro Bazzo
IPEA – Instituto de Pesquisa Econômica Aplicada

Membro(a) 2): _____
Marcelo Risso Errera
Departamento de Engenharia Ambiental - UFPR

Membro(a) 3): _____

Curitiba, 13 de dezembro de 2021