# Classificação da Qualidade do Ar Utilizando Técnicas de Aprendizado de Máquina


Bruno da Silva Macêdo
*Pós-Graduação em Engenharia de Sistemas e Automação*
*Universidade Federal de Lavras (UFLA)*
Divinópolis-MG, Brasil
bruno.macedo2@estudante.ufla.br

Rita de Cássia de Vasconcelos Pedrosa
*Departamento de Biometria e Estatística Aplicada*
*Universidade Federal Rural de Pernambuco (UFRPE)*
Recife-PE, Brasil
rita.cvpedrosa@gmail.com

Rafael Ávila dos Santos
*Pós-Graduação em Engenharia de Sistemas e Automação*
*Universidade Federal de Lavras (UFLA)*
Lavras-MG, Brasil
rafael.santos20@estudante.ufla.br

Camila Martins Saporetti
*Departamento de Modelagem Computacional*
*Universidade do Estado do Rio de Janeiro (UERJ)*
Nova Friburgo-RJ, Brasil
camila.saporetti@iprj.uerj.br

Bruno Henrique Groenner
*Departamento de Automática*
*Universidade Federal de Lavras (UFLA)*
Lavras-MG, Brasil
brunohb@ufla.br

Danton Diego Ferreira
*Departamento de Automática*
*Universidade Federal de Lavras (UFLA)*
Lavras-MG, Brasil
danton@ufla.br

Leonardo Goliatt
*Departamento de Mecânica Computacional e Aplicada*
*Universidade Federal de Juiz de Fora (UFJF)*
Juiz de Fora-MG, Brasil
leonardo.goliatt@ufjf.br

Lívia Marques Rodrigues
*Graduanda em Engenharia de Controle e Automação*
*Universidade Federal de Lavras (UFLA)*
Lavras-MG, Brasil
livia.rodrigues3@estudante.ufla.br



***Resumo*—A poluição do ar representa enormes ameaças à saúde pública e à sustentabilidade ambiental em cidades em todo o mundo, impactando diretamente a qualidade de vida da população e o equilíbrio dos ecossistemas. Dessa forma, analisar a qualidade do ar tornou-se vital para a sociedade e seus indivíduos. A análise tradicional da qualidade do ar normalmente depende da simulação numérica de modelos atmosféricos. No entanto, tais métodos baseados em simulação são computacionalmente caros e exigem extenso conhecimento do domínio, dificultando sua aplicação em larga escala e em tempo real. Neste contexto, este trabalho tem como objetivo realizar a classificação da qualidade do ar com dados do Sudeste Asiático, especificamente Paquistão, Índia, Bangladesh e Sri Lanka, empregando três algoritmos de *Machine Learning* (ML): *Naive Bayes* (NB), *Random Forest* (RF) e *Support Vector Machine* (SVM). As técnicas de pré-processamento incluíram normalização e balanceamento de classes, com SMOTE e *Near Miss*. A otimização dos hiperparâmetros dos algoritmos foi realizada via *Grid-Search* e validação cruzada. O RF apresentou os melhores resultados, seguido pelo SVM. Os resultados destacam a importância do ML no monitoramento e na gestão da qualidade do ar, essenciais para a saúde pública e a sustentabilidade ambiental.**



***Palavras-chave*—*Machine Learning*, *Grid-Search*, Qualidade do Ar, Poluição, Desbalanceamento.**



Agradecimentos a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).


## I. Introdução

O ar é uma mistura homogênea de gases que compõe a atmosfera terrestre, essencial para a manutenção da vida no planeta. Em sua composição básica, o ar é formado principalmente por nitrogênio (cerca de 78%) e oxigênio (cerca de 21%), além de outros gases em menores concentrações, como dióxido de carbono (0,04%), argônio (0,93%) e traços de gases nobres, como hélio, néon e criptônio [1].

A qualidade do ar é diretamente influenciada pela concentração de poluentes, originados tanto de fontes naturais (como atividades vulcânicas e emissões biológicas) quanto antropogênicas (como transporte e processos industriais) [2]. Essa composição impacta a saúde humana, o equilíbrio ambiental e o clima, tornando o monitoramento da poluição atmosférica uma ação crítica para a sustentabilidade urbana [3].

A poluição do ar representa enormes ameaças à saúde pública e à sustentabilidade ambiental em cidades em todo

o mundo. Segundo a Organização Mundial da Saúde (OMS), a poluição do ar aumentou o risco de vários problemas de saúde entre os cidadãos e impôs um fardo econômico significativo à sociedade [4]–[6]. Portanto, a análise da qualidade do ar tornou-se vital para a sociedade e seus indivíduos. Por um lado, uma análise correta da poluição do ar permite aos decisores políticos formularem medidas eficazes, regulamentações ambientais e intervenções direcionadas para mitigar as emissões de poluição [7]. Além disso, também pode capacitar os indivíduos a tomarem decisões informadas, como ajustar rotas de viagem ou reduzir atividades ao ar livre, para minimizar a exposição a poluentes nocivos [4].

Todavia, a análise tradicional da qualidade do ar normalmente depende da simulação numérica de modelos atmosféricos. No entanto, tais métodos baseados em simulação são computacionalmente caros e exigem extenso conhecimento do domínio [8]. Além disso, tais abordagens são inevitavelmente insuficientes na capacidade de capturar mecanismos complexos de poluição do ar devido ao conhecimento incompleto sobre a atmosfera e a uma variedade de fatores envolvidos [8]. Como alternativa, modelos de *Machine Learning* (ML) têm proporcionado grandes oportunidades para lidar com tarefas de poluição do ar a partir de uma perspectiva baseada em dados [9], [10], com sua capacidade de processar grandes volumes de dados e identificar padrões complexos.

Nesse sentido, este trabalho tem como objetivo avaliar e comparar o desempenho de três modelos de ML com abordagens diferentes: *Naive Bayes* (NB), *Random Forest* (RF) e *Support Vector Machine* (SVM) na classificação da qualidade do ar. Para isso, utiliza-se o *Grid-Search* para otimizar os parâmetros dos métodos. A análise é realizada com base nos dados da *"Air Quality and Pollution Assessment"* [11]. A base de dados abrange informações ambientais de países do Sudeste Asiático, especificamente Paquistão, Índia, Bangladesh e Sri Lanka.

O trabalho foi estruturado seguindo os tópicos: Na Seção II são apresentados alguns trabalhos relacionados. A Seção III encontra-se a descrição dos materiais e métodos utilizados para o desenvolvimento do trabalho. Já na Seção IV, por sua vez, são apresentados os métodos de classificação empregados neste trabalho. Na Seção V são apresentados e discutidos os resultados obtidos. Por fim, a Seção VI apresenta as conclusões.

## II. Trabalhos Relacionados

A classificação da qualidade e poluição do ar é um tema que vem sendo estudado por diversos pesquisadores. Nesses trabalhos, análises são realizadas e ferramentas propostas no contexto de ML para compreender o problema e buscar soluções para prever os resultados.

Corani, G., & Scanagatta [12] introduziram um classificador de rede bayesiana multirrótulo para prever simultaneamente múltiplas variáveis de poluição atmosférica, capturando suas dependências estocásticas. Resultados experimentais em três estudos de caso demonstram maior precisão em relação a classificadores independentes, particularmente para a previsão de PM2.5 e ozônio.

Saxena & Shekhawat [13] propuseram uma estrutura matemática para calcular um Índice Cumulativo (IC) a partir de quatro poluentes principais e o utilizam como entrada para um classificador de Máquina de Vetores de Suporte (SVM). O classificador categoriza efetivamente a qualidade do ar como boa ou prejudicial, utilizando dados reais de Calcutá, Delhi e Bhopal.

Liu *et al.* [14] aplicaram Regressão de Vetores de Suporte (RVS) e Regressão de Floresta Aleatória (RFR) para construir modelos de regressão para prever o Índice de Qualidade do Ar (IQA) em Pequim. Os resultados experimentais mostraram que o modelo baseado em SVR teve melhor desempenho na previsão do IQA, com Raiz do Erro Quadrático Médio (RMSE) igual a 7,666 e Coeficiente de Determinação ($R^2$) de 0,9776.

Mahanta *et al.* [15] investigou a eficácia de alguns modelos de ML na previsão dos valores do IQA, com alguns dados de entrada, com base na poluição e em informações meteorológicas em Nova Delhi, Índia. Os resultados mostram que os modelos preditivos ajudam na previsão da qualidade do ar.

Castelli *et al.* [16] utilizaram a RVS com um *kernel* de função de base radial (RBF) para prever com precisão os níveis horários de poluentes e o IQA na Califórnia. O modelo obteve 94,1% de precisão na classificação do IQA em seis categorias definidas pela Agência de Proteção Ambiental, superando a seleção de recursos via Análise de Componentes Principais.

Lei *et al.* [17] utilizaram os métodos de ML, como RF, *Gradiente Boosting* (GB), SVR e Regressão Linear Múltipla (RLM), para prever IQA na região de Macau, na China. O RF foi o método que apresentou os melhores resultados.

Maltare & Vahora [18] compararam vários métodos de ML, como *Seasonal Autorregressive Integrated Moving Average* (SARIMA), SVM com kernel RGB, e *Long Short-Term Memory* (LSTM) para a previsão do IQA para a cidade de Ahmedabad, em Gujarat, Índia. Os resultados obtidos são comparativamente melhores em comparação com outros kernels dos modelos da SVM, bem como os modelos SARIMA e LSTM.

Ravindiran *et al.* [19] buscaram prever o IQA com técnicas de ML, para cidade de Visakhapatnam, Andhra Pradesh, Índia, com foco em 12 contaminantes e 10 parâmetros meteorológicos de julho de 2017 a setembro de 2022. Foram utilizados os algoritmos *LightGBM*, *Random Forest* (RF), *Catboost*, *Adaboost* e *XGBoost*. O algoritmo *Catboost* foi o algoritmo que apresentou os melhores resultados, superando os outros algoritmos com um RMSE de 0,76.

Ahmadi *et al.* [20] apresentaram uma nova função discreta de custo/perda para classificadores inteligentes, abordando a incompatibilidade entre funções de perda contínuas e a natureza discreta da classificação. Aplicado o *perceptrons* multicamadas em conjuntos de dados de qualidade do ar, o modelo proposto supera as versões convencionais, alcançando uma taxa média de classificação de 87,68%.

Percebe-se que aplicar métodos de ML é algo promissor e empregar uma abordagem para selecionar os melhores parâmetros dos métodos irá auxiliar e possibilitar realizar a classificação com um melhor desempenho. Ademais, o desbalanceamento nas classes interfere diretamente no aprendizado dos métodos. Então, objetiva-se classificar a qualidade do ar por meio de métodos de ML, avaliar o uso do *Grid-Search* para encontrar os melhores parâmetros para os métodos de classificação e verificar o impacto do desbalanceamento dos dados nessa tarefa.

## III. Materiais e Métodos

Nesta seção são descritas a base de dados utilizada, o pré-processamento realizado, as técnicas de balanceamento aplicadas, a validação cruzada e a técnica de otimização de hiperparâmetros utilizada.

### A. Base de Dados

A base de dados utilizada foi obtida de Mateen [11], intitulada *"Air Quality and Pollution Assessment"*. Essa base contém dados de países do Sudeste Asiático, especificamente Paquistão, Índia, Bangladesh e Sri Lanka, de fatores ambientais e demográficos críticos que influenciam os níveis de poluição. Ao todo, a base possui 5.000 amostras com 10 atributos:

- ***Temperature (°C)***: Temperatura média da região;
- ***Humidity (%)***: Umidade relativa registrada na região;
- ***$PM_{2.5}$ Concentration ($\mu g/m^3$)***: Concentração de partículas finas;
- ***$PM_{10}$ Concentration ($\mu g/m^3$)***: Concentração de partículas grossas;
- ***$NO_2$ Concentration (ppb)***: Concentração de dióxido de nitrogênio;
- ***$SO_2$ Concentration (ppb)***: Concentração de dióxido de enxofre;
- ***CO Concentration (ppm)***: Concentração de monóxido de carbono;
- ***Proximity to Industrial Areas (km)***: Distância até a zona industrial mais próxima;
- ***Population Density (people/km²)***: Densidade populacional (número de pessoas por quilômetro quadrado na região);
- ***Air Quality Levels***: Variável alvo que classifica a qualidade do ar em quatro categorias - *Good* (Bom), *Moderate* (Moderado), *Poor* (Ruim) e *Hazardous* (Perigoso).

### B. Pré-Processamento

No pré-processamento, foi realizada a transformação da variável alvo, utilizando o método *Label Encoder* [21], para transformação de categórica para numérica. Ou seja, as categorias *Good*, *Moderate*, *Poor* e *Hazardous* foram transformadas para classes 0, 1, 2 e 3, respectivamente. O número de amostras para as classes 0, 1, 2 e 3 é 2000, 1500, 1000 e 500, respectivamente. Foi aplicado o *Holdout* [22] para dividir a base em treinamento e teste, sendo 80% e 20%, respectivamente. Além disso, as variáveis utilizadas para prever a variável alvo, no conjunto de treinamento, foram normalizadas com o método *MinMaxScaler* [21], fazendo com que os valores ficassem em um intervalo entre 0 e 1. Esse processo é útil para garantir que todas as variáveis tenham a mesma escala, prevenindo que variáveis com magnitudes significativamente maiores impactem de forma desproporcional a otimização e o treinamento dos modelos de ML [23].

### C. Balanceamento

Um dos problemas mais comuns em bases de dados é o desbalanceamento, que ocorre quando há uma diferença significativa no número de amostras entre as classes. Para lidar com esta situação, duas abordagens geralmente utilizadas são o *Undersampling* e o *Oversampling*. O *Undersampling* consiste em diminuir o número de amostras da classe majoritária, ajustando-a para que a classe majoritária seja igual em termos de amostras à classe minoritária. Em contraste, o *Oversampling* aumenta o número de amostras da classe minoritária, para igualar o número de amostras à classe majoritária. O *Near Miss* e *SMOTE* utilizam as técnicas de *Undersampling* e *Oversampling*, respectivamente.

O método *Near Miss* identifica amostras da classe majoritária para remoção com base em sua proximidade em relação às amostras da classe minoritária. Essa técnica utiliza o conceito de $k$ vizinhos mais próximos, com a distância euclidiana sendo frequentemente aplicada como métrica. As amostras da classe majoritária selecionadas para exclusão são aquelas que apresentam a menor distância média em relação aos três vizinhos mais próximos da classe minoritária [24].

Já o *Synthetic Minority Oversampling Technique* (SMOTE) é um método de sobreamostragem que gera novas amostras sintéticas para a classe minoritária. Esse processo ocorre selecionando cada amostra da classe minoritária e criando exemplos sintéticos ao longo das linhas que conectam a amostra selecionada aos k vizinhos mais próximos dentro da mesma classe [25].

### D. Validação Cruzada e Grid-Search

Validação Cruzada (VC) é uma metodologia estatística utilizada para avaliar a capacidade de generalização de modelos preditivos, com base na divisão do conjunto de dados [26]. Essa técnica é particularmente relevante em cenários nos quais o objetivo principal é realizar predições confiáveis. A VC divide o conjunto de dados em dois subconjuntos: o conjunto de treinamento, utilizado para ajustar os parâmetros do modelo, e o conjunto de teste, empregado para avaliar o desempenho preditivo do modelo em dados não vistos.

Para a avaliação dos classificadores, foi utilizada a estratégia de validação cruzada *K-Fold* (KF) [26]. Nesta abordagem, o conjunto de dados disponível, contendo $N$ amostras, é particionado em K subconjuntos, onde $K > 1$. Após a divisão, $K - 1$ subconjuntos são utilizados para o treinamento do modelo, enquanto o subconjunto restante é reservado para validação. Ao final do processo, calcula-se o erro de validação com base nos resultados obtidos. Esse procedimento é repetido $K$ vezes, alternando o subconjunto de teste a cada iteração,

garantindo que todas as amostras sejam utilizadas tanto para treinamento quanto para validação.

O procedimento de *Grid-Search* consiste em uma busca exaustiva pelos melhores hiperparâmetros de um modelo. Na qual é explorado sistematicamente um intervalo pré-definido de combinações de parâmetros [27]. Quando combinado com a validação cruzada, o *Grid-Search* avalia cada combinação de parâmetros ajustados para o classificador, com o objetivo de identificar aquela que maximiza a acurácia do modelo.

## IV. Métodos de Classificação

Nesta seção, são apresentados os métodos de classificação utilizados neste trabalho para a obtenção dos resultados e as métricas para avaliar o desempenho dos métodos.

### A. Naïve Bayes (NB)

O método *Naïve Bayes* (NB) é baseado na geração de estimativas de probabilidade. Para cada classe, o algoritmo produz uma estimativa da probabilidade de um objeto pertencer a essa classe [28]. O NB calcula essas probabilidades com base nas frequências de associação entre os valores da classe e os valores dos diferentes atributos de entrada observados nos dados de treinamento. Contudo, é fundamental destacar uma premissa essencial para o funcionamento desse classificador: a suposição de independência condicional entre os atributos, dado que a classe é conhecida. Essa hipótese implica que, para cada classe, as variáveis de entrada são consideradas estatisticamente independentes umas das outras. O Naive Bayes utiliza o teorema de Bayes para calcular a probabilidade posterior $P(C_k|X)$ de um exemplo $X$ pertencer a uma classe $C_k$. A fórmula geral é dada por:

$$P(C_k \mid X) = \frac{P(C_k) \cdot P(X \mid C_k)}{P(X)} \tag{1}$$

onde $P(C_k|X)$ indica a probabilidade posterior da classe $C_k$ dado um exemplo $X$, $P(C_k)$ é a probabilidade a priori da classe $C_k$, já $P(X|C_k)$ é a probabilidade de observar $X$ dado que pertence à classe $C_k$ e $P(X)$ é a probabilidade do exemplo $X$ ser observado (constante para todas as classes).

### B. Random Forest (RF)

O *Random Forest* (RF), proposto por Breiman [29], é um método de aprendizagem por ensemble que opera construindo k árvores de decisão a partir do conjunto de treinamento em $k$ iterações. Em cada iteração, primeiro é selecionado aleatoriamente um conjunto de amostras do conjunto de treinamento. Para reproduzir uma árvore de decisão desse subconjunto, o RF escolhe aleatoriamente um subconjunto de atributos candidatos para cada nó. Desta forma, cada árvore de decisão é construída através do *ensemble*, empregando subconjuntos aleatórios independentes de características e amostras. A predição de uma nova classe de amostra é realizada da seguinte forma: cada classificador individual vota e a classe mais votada é eleita.

Um RF consiste em uma combinação de classificadores, onde cada classificador contribui com um único voto para a atribuição da classe mais frequente ao vetor de entrada. O fato de ser uma combinação de muitos classificadores confere ao RF algumas características que o tornam substancialmente diferentes de uma árvore de classificação tradicional e, portanto, deve ser entendido como um novo conceito de classificadores. A Fig. 1 ilustra um exemplo do algoritmo RF, na qual a predição final é da classe que mais foi votada dentre das árvores.

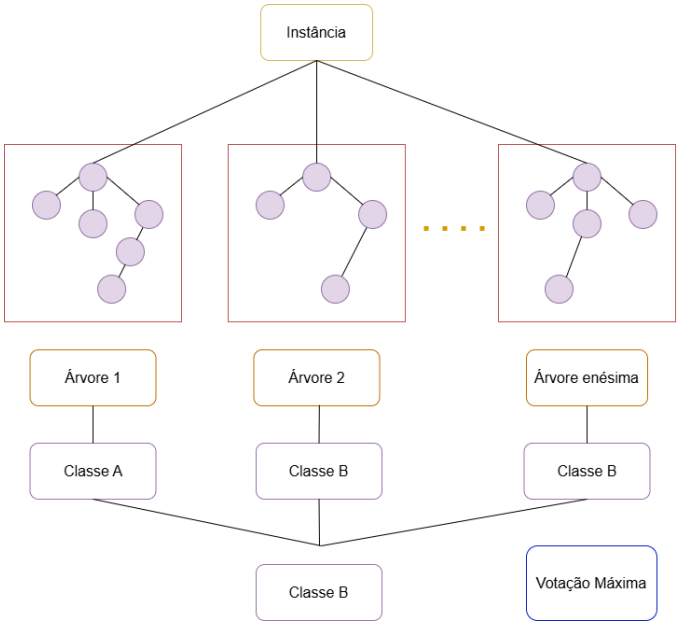


Fig. 1. Exemplo de uma arquitetura do RF. Fonte: Elaborado pelos autores.

### C. Support Vector Machine (SVM)

O *Support Vector Machine* (SVM) é um método de aprendizagem supervisionada, com o objetivo de criar um limite de decisão entre duas ou mais classes, permitindo a predição de rótulos de um ou mais recursos de vetores [30]. O limite de decisão é chamado de hiperplano, é orientado de forma que fique o mais longe possível dos pontos de dados com maior proximidade de cada uma das classes. São chamados de vetores de suporte estes pontos que estão mais próximos. Seja um conjunto de dados de treinamento rotulado expresso pela Eq. 2:

$$(x_1, y_1), \ldots, (x_n, y_n), x_i \in \mathbb{R}^d \quad \text{e} \quad y_i \in (-1, +1) \tag{2}$$

onde $x_i$ é a representação do vetor de recursos, $y_i$ a classe, podendo ser negativa ou positiva de um item de treinamento $i$. A otimização do hiperplano é definida pela Eq. 3:

$$w^T x + b = 0 \tag{3}$$

onde $w$ é o vetor de peso, $x$ é o vetor de recurso de entrada, e $b$ é o viés. $w$ e $b$ devem satisfazer as seguintes desigualdades para todo conjunto de treinamento (Eq. 4).

$$w^T x + b \geq +1 \quad \text{se} \quad y_i = 1$$
$$w^T x + b \leq -1 \quad \text{se} \quad y_i = -1 \tag{4}$$

O modelo SVM é treinado para encontrar um $w$ e $b$ que permita realizar a separação dos dados no hiperplano e maximize a margem. Muitas vezes, os dados reais não são linearmente separáveis no espaço original das características. Para contornar essa limitação, o SVM utiliza a técnica de funções *kernel*, que consiste em mapear os dados para um espaço de dimensão superior, onde a separação linear torna-se possível. O modelo SVM, por se tratar de um problema de classificação, é denominado como *Support Vector Classification* (SVC).

### D. Métricas de Avaliação

Para avaliar o desempenho, foram utilizadas as seguintes métricas: Acurácia (Ac), Precisão (Pr), *Recall* (Re) e Coeficiente de *Kappa*. Na equação das métricas, *True Positives* (TP) representa as amostras corretamente classificadas como pertencentes a uma determinada classe, *False Positives* (FP) é o número de casos incorretamente classificados como pertencente à classe, *True Negative* (TN) representa o número de casos corretamente classificados como não pertencente a uma classe e *False Negative* (FN) indica o número de casos incorretamente classificados como não pertencente à classe avaliada.

A acurácia (Eq. 5) calcula a proporção das previsões corretas em relação ao total de instâncias.

$$\text{Acurácia} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

Precisão (Eq. 6) é uma métrica que mede o número de *True Positives* entre as amostras classificadas como pertencentes a determinada classe [31].

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

*Recall* (Eq. 7), também denominado de Sensibilidade ou *True Positives Rate* (TPR) é a medida de *True Positives*, isto é, o número de amostras classificadas como pertencentes a determinada classe, entre todas as amostras originalmente pertencentes a essa classe [31].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

O Coeficiente *Kappa* (Eq. 8) mede o nível de concordância entre dois especialistas, corrigindo a concordância esperada pelo acaso. Um ($k = 1$) indica uma concordância perfeita, para o ($k = 0$) representa concordância similar ao acaso e ($k < 1$) indica uma discordância [31].

$$kappa = \frac{P_o - P_e}{1 - P_e} \tag{8}$$

Onde, $P_o$ é a proporção de concordâncias observadas e $P_e$ é a proporção de concordâncias esperadas ao acaso. A interpretação da Estatística Kappa, utilizada para avaliar o nível de concordância entre os avaliadores, é segundo [32].

## V. Resultados e Discussões

Os experimentos computacionais foram realizados utilizando a linguagem de programação *Python* e implementações baseadas nas bibliotecas de estrutura *Scikit-Learn* [21], Pandas [33]. Todos os experimentos foram executados em um computador com as seguintes especificações: $13^a$ geração *Intel CoreTM i7-13700* (16-cores, 24 threads, cache de 30 MB, 2.10 GHz a 5.20 GHz Turbo, 65 W) e sistema operacional *Ubuntu* 22.04. Foram realizadas 30 iterações independentes para avaliar a metodologia, como também foi utilizado um KF com valor de $K$ igual a 5.

A Tabela I apresenta a descrição dos métodos utilizados no *Grid-Search* com a validação cruzada, indicando o método, os parâmetros para maximizar o desempenho do modelo e qual a configuração utilizada neles, indicando os valores usados durante as execuções.

TABELA I
DESCRIÇÃO DOS MÉTODOS.

| Método | Parâmetros | Variação Parâmetros |
|---|---|---|
| NB | *var_smoothing* | $[1 \times 10^{-9}, 1 \times 10^{-8}, 1 \times 10^{-7}$, $1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}$ $1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$ |
| RF | criterion | ['*gini*', '*entropy*'] |
| | *n_estimators* | [50, 100, 200, 300] |
| | *max_depth* | [None, 5, 10, 20, 30] |
| | *min_samples_split* | [2, 5, 10] |
| | *min_samples_leaf* | [1, 2, 4] |
| SVM | C | $[1 \times 10^{-2}, 1 \times 10^{-1}, 1 \times 10^{0}]$ $[1 \times 10^{1}, 1 \times 10^{2}]$ |
| | *gamma* | $[1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$ |
| | *kernel* | ['*linear*', '*rbf*', '*poly*', '*sigmoid*'] |

Na Tabela II pode-se visualizar os resultados obtidos, com a média e o desvio padrão das iterações para cada métrica. O melhor resultado obtido foi com o algoritmo RF, com uma acurácia de 0.9540, precisão de 0.9543, *recall* de 0.9540 e um kappa de 0.9340, mostrando um nível de concordância perfeita do resultado.

TABELA II
RESULTADO DAS MÉTRICAS DE DESEMPENHO DOS MODELOS PARA OS DADOS DESBALANCEADOS. O * INDICA QUE HÁ DIFERENÇA ESTATISTICAMENTE SIGNIFICANTE, DE ACORDO COM O TESTE DE *Tukey*, COMPARADO AO MELHOR MODELO (EM NEGRITO).

| Método | Ac | Pr | Re | *Kappa* |
|---|---|---|---|---|
| NB | 0.9267 ± 0.0109 * | 0.9266 ± 0.0109 * | 0.9267 ± 0.0109 * | 0.8948 ± 0.0153 * |
| **RF** | **0.9540 ± 0.0062** | **0.9543 ± 0.0061** | **0.9540 ± 0.0062** | **0.9340 ± 0.0088** |
| SVM | 0.9421 ± 0.0071 * | 0.9420 ± 0.0074 * | 0.9421 ± 0.0074 * | 0.9168 ± 0.0105 * |

A Tabela III apresenta a configuração dos parâmetros do melhor modelo do RF no conjunto de teste. O melhor método é composto pela métrica usada para medir a qualidade das divisões (splits) nos nós das árvores de decisão dentro do RF, que foi o Índice de Gini, a profundidade máxima de cada árvore igual a 30, o número mínimo de amostras que um nó folha (nó terminal) deve ter igual a 1, o número mínimo de amostras exigidas para que um nó possa ser dividido igual

a 10 e o número de árvores que compõem é igual a 300. Tal configuração resultou, no conjunto de teste, acurácia de 0.9590, precisão de 0.9590, *recall* de 0.9590 e *kappa* de 0.9413.

TABELA III
DESEMPENHO DO MELHOR MODELO NO CONJUNTO DE TESTE COM DADOS DESBALANCEADOS.

| Parâmetros | Ac | Pr | Re | *Kappa* |
|---|---|---|---|---|
| *criterion* : *gini*, *max_depth*: 30, *min_samples_leaf*: 1, *min_samples_split*: 10 *n_estimators*: 300 | 0.9590 | 0.9590 | 0.9590 | 0.9413 |

Na Tabela IV exibe os resultados obtidos para os dados balanceados com o método *Near Miss*, com a média e o desvio padrão das iterações para cada métrica. O melhor resultado obtido foi com o algoritmo RF, com uma acurácia de 0.9247, precisão de 0.9261, *recall* de 0.9247 e um kappa de 0.8720, mostrando um nível de concordância perfeita do resultado. Para o método *Near Miss*, que consiste em diminuir o número de amostras da classe majoritária, ajustando-a para que a classe majoritária seja igual em termos de amostras à classe minoritária, o método RF mostrou-se mais adequado. A diferença entre o RF e SVM não se mostrou estatisticamente significante de acordo com o teste de *Tukey*.

TABELA IV
RESULTADO DAS MÉTRICAS DE DESEMPENHO DOS MODELOS PARA OS DADOS BALANCEADOS COM O MÉTODO *Near Miss*. O * INDICA QUE HÁ DIFERENÇA ESTATISTICAMENTE SIGNIFICANTE, DE ACORDO COM O TESTE DE *Tukey*, COMPARADO AO MELHOR MODELO (EM NEGRITO).

| Método | Ac | Pr | Re | *Kappa* |
|---|---|---|---|---|
| NB | 0.9042 ± 0.0141 * | 0.9076± 0.0129 * | 0.9042 ± 0.0141 * | 0.8954 ± 0.0115 * |
| **RF** | **0.9247 ± 0.0150** | **0.9261 ± 0.0146** | **0.9247 ± 0.0150** | **0.8720 ± 0.0188** |
| SVM | 0.9217 ± 0.0113 | 0.9231± 0.0119 | 0.9217 ± 0.0113 | 0.8954 ± 0.0115 |

A Tabela V mostra a configuração dos parâmetros do melhor modelo do RF no conjunto de teste. O melhor método é composto pela métrica usada para medir a qualidade das divisões (splits) nos nós das árvores de decisão dentro do RF, que foi o Índice de Gini, a profundidade máxima de cada árvore igual a 10, o número mínimo de amostras que um nó folha (nó terminal) deve ter igual a 2, o número mínimo de amostras exigidas para que um nó possa ser dividido igual a 5 e o número de árvores que compõem é igual a 200. Tal configuração resultou, no conjunto de teste, acurácia de 0.9600, precisão de 0.9600, *recall* de 0.9600 e *kappa* de 0.9427.

TABELA V
DESEMPENHO DO MELHOR MODELO NO CONJUNTO DE TESTE COM DADOS BALANCEADOS COM O MÉTODO *Near Miss*.

| Parâmetros | Ac | Pr | Re | *Kappa* |
|---|---|---|---|---|
| *criterion* : *gini*, *max_depth*: 10, *min_samples_leaf*: 2, *min_samples_split*: 5 *n_estimators*: 200 | 0.9600 | 0.9600 | 0.9600 | 0.9427 |

A Tabela VI apresenta os resultados obtidos para os dados balanceados com o SMOTE, com a média e o desvio padrão das iterações para cada métrica. O melhor resultado obtido foi com o algoritmo RF, com uma acurácia de 0.9640, precisão de 0.9640, *recall* de 0.9640 e um kappa de 0.9520, mostrando um nível de concordância perfeita do resultado. Para o método SMOTE, que consiste em aumentar o número de amostras da classe minoritária para igualar o número de amostras à classe majoritária, o método RF se mostrou mais adequado.

TABELA VI
RESULTADO DAS MÉTRICAS DE DESEMPENHO DOS MODELOS PARA OS DADOS BALANCEADOS COM O MÉTODO SMOTE. O * INDICA QUE HÁ DIFERENÇA ESTATISTICAMENTE SIGNIFICANTE, DE ACORDO COM O TESTE DE *Tukey*, COMPARADO AO MELHOR MODELO (EM NEGRITO).

| Método | Ac | Pr | Re | *Kappa* |
|---|---|---|---|---|
| NB | 0.9144 ± 0.0087* | 0.9143 ± 0.0086 * | 0.9144 ± 0.0087 * | 0.8858 ± 0.0117 * |
| **RF** | **0.9640 ± 0.0056** | **0.9640 ± 0.0056** | **0.9640 ± 0.0056** | **0.9520 ± 0.0075** |
| SVM | 0.9342 ± 0.0063 * | 0.9340 ± 0.0064 * | 0.9342 ± 0.0063 * | 0.9123 ± 0.0085 * |

A Tabela VII apresenta a configuração dos parâmetros do melhor modelo do RF no conjunto de teste. O melhor método é composto por: a métrica usada para medir a qualidade das divisões (*splits*) nos nós das árvores de decisão dentro do RF foi a entropia, a profundidade máxima de cada árvore igual a 20, o número mínimo de amostras que um nó folha (nó terminal) deve ter igual a 1, o número mínimo de amostras exigidas para que um nó possa ser dividido igual a 2 e o número de árvores que compõem é igual a 50. Tal configuração resultou, no conjunto de teste, acurácia de 0.9550, precisão de 0.9550, *recall* de 0.9550 e *kappa* de 0.9356.

TABELA VII
DESEMPENHO DO MELHOR MODELO NO CONJUNTO DE TESTE COM DADOS BALANCEADOS COM O MÉTODO SMOTE.

| Parâmetros | Ac | Pr | Re | *Kappa* |
|---|---|---|---|---|
| *criterion* : *entropy*, *max_depth*: 20, *min_samples_leaf*: 1, *min_samples_split*: 2 *n_estimators*: 50 | 0.9550 | 0.9550 | 0.9550 | 0.9356 |

## VI. CONCLUSÕES

Neste trabalho, foi analisado o desempenho de três métodos de classificação (NB, RF, SVM) para classificação da qualidade do ar no Sudeste Asiático, especificamente Paquistão, Índia, Bangladesh e Sri Lanka. Com o intuito de maximizar o desempenho dos métodos, foi utilizado o *Grid-Search* para otimização dos hiperparâmetros dos métodos. O método RF foi o que apresentou os melhores resultados para os dados desbalanceados e balanceados com o SMOTE e *Near Miss* como abordagem. O RF atingiu uma acurácia de 0.9540 e um *kappa* de 0.9340 com dados desbalanceados. Já com a abordagem de balanceamento *Near Miss*, obteve uma acurácia de 0.9247 e um *kappa* de 0.8720. Com a abordagem de balanceamento SMOTE, a acurácia foi 0.9640 e o *kappa* 0.9520.

Além disso, o método SVM também apresentou bons resultados, com o balanceamento dos dados com a abordagem *Near Miss* e SMOTE, com uma acurácia de 0.9217 e 0.9342, respectivamente. Dessa forma, a metodologia proposta se mostrou eficiente, podendo auxiliar especialistas na tarefa de classificação da qualidade do ar. Como trabalhos futuros, pretende-se aplicar algumas meta-heurísticas, como Otimização de Lobos Cinzentos e Otimização por Enxames de Partículas, para encontrar os parâmetros ótimos, realizando uma busca contínua. Como também, realizar formulações de políticas públicas voltadas à mitigação da poluição atmosférica e na implementação de sistemas inteligentes de monitoramento da qualidade do ar em tempo real. Tais desdobramentos podem contribuir significativamente para a tomada de decisão em contextos urbanos e para a ampliação do acesso a informações ambientais em regiões carentes de infraestrutura de medição convencional.

## AGRADECIMENTOS

Os autores agradecem o apoio financeiro da Universidade Federal de Lavras (UFLA), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

## REFERÊNCIAS

[1] J. H. Seinfeld, S. N. Pandis *et al.*, "From air pollution to climate change," *Atmospheric chemistry and physics*, vol. 1326, 1998.

[2] N. S. Holmes and L. Morawska, "A review of dispersion modelling and its application to the dispersion of particles: An overview of different dispersion models available," *Atmospheric environment*, vol. 40, no. 30, pp. 5902–5928, 2006.

[3] J. Zaheer, J. Jeon, S.-B. Lee, and J. S. Kim, "Effect of particulate matter on human health, prevention, and imaging using pet or spect," *Progress in Medical Physics*, vol. 29, no. 3, pp. 81–91, 2018.

[4] J. Han, W. Zhang, H. Liu, and H. Xiong, "Machine learning for urban air quality analytics: A survey," *arXiv preprint arXiv:2310.09620*, 2023.

[5] K. Ravindra, "Emission of black carbon from rural households kitchens and assessment of lifetime excess cancer risk in villages of north india," *Environment international*, vol. 122, pp. 201–212, 2019.

[6] W. H. Organization *et al.*, "World health statistics 2021: Monitoring health for the sdgs, sustainable development goals. who, geneva, switzerland, 2021."

[7] Y. Rybarczyk and R. Zalakeviciute, "Assessing the covid-19 impact on air quality: A machine learning approach," *Geophysical Research Letters*, vol. 48, no. 4, p. e2020GL091202, 2021.

[8] S. Vardoulakis, B. E. Fisher, K. Pericleous, and N. Gonzalez-Flesca, "Modelling air quality in street canyons: a review," *Atmospheric environment*, vol. 37, no. 2, pp. 155–182, 2003.

[9] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting fine-grained air quality based on big data," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 2267–2276.

[10] K. Fan, R. Dhammapala, K. Harrington, R. Lamastro, B. Lamb, and Y. Lee, "Development of a machine learning approach for local-scale ozone forecasting: application to kennewick, wa," *Frontiers in big Data*, vol. 5, p. 781309, 2022.

[11] M. Mateen, "Air quality and pollution assessment," 2024. [Online]. Available: https://www.kaggle.com/ds/6197184

[12] G. Corani and M. Scanagatta, "Air pollution prediction via multi-label classification," *Environmental modelling & software*, vol. 80, pp. 259–264, 2016.

[13] A. Saxena and S. Shekhawat, "Ambient air quality classification by grey wolf optimizer based support vector machine," *Journal of environmental and public health*, vol. 2017, no. 1, p. 3131083, 2017.

[14] H. Liu, Q. Li, D. Yu, and Y. Gu, "Air quality index and air pollutant concentration prediction based on machine learning algorithms," *Applied sciences*, vol. 9, no. 19, p. 4069, 2019.

[15] S. Mahanta, T. Ramakrishnudu, R. R. Jha, and N. Tailor, "Urban air quality prediction using regression analysis," in *Tencon 2019-2019 ieee region 10 conference (tencon)*. IEEE, 2019, pp. 1118–1123.

[16] M. Castelli, F. M. Clemente, A. Popovič, S. Silva, and L. Vanneschi, "A machine learning approach to predict air quality in california," *Complexity*, vol. 2020, no. 1, p. 8049504, 2020.

[17] T. M. Lei, S. W. Siu, J. Monjardino, L. Mendes, and F. Ferreira, "Using machine learning methods to forecast air quality: A case study in macao," *Atmosphere*, vol. 13, no. 9, p. 1412, 2022.

[18] N. N. Maltare and S. Vahora, "Air quality index prediction using machine learning for ahmedabad city," *Digital Chemical Engineering*, vol. 7, p. 100093, 2023.

[19] G. Ravindiran, G. Hayder, K. Kanagarathinam, A. Alagumalai, and C. Sonne, "Air quality prediction by machine learning models: A predictive study on the indian coastal city of visakhapatnam," *Chemosphere*, vol. 338, p. 139518, 2023.

[20] M. Ahmadi, M. Khashei, and N. Bakhtiarvand, "Enhancing air quality classification using a novel discrete learning-based multilayer perceptron model (dmlp)," *International Journal of Environmental Science and Technology*, vol. 22, no. 5, pp. 3051–3062, 2025.

[21] P. Fabian, "Scikit-learn: Machine learning in python," *Journal of machine learning research 12*, p. 2825, 2011.

[22] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, 1995, pp. 1137–1143.

[23] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, "The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis," in *International conference on soft computing models in industrial and environmental applications*. Springer, 2023, pp. 344–353.

[24] J. Brownlee, "Undersampling algorithms for imbalanced classification," *Machine Learning Mastery*, vol. 27, 2020.

[25] V. C. Nitesh, "Smote: synthetic minority over-sampling technique," *J Artif Intell Res*, vol. 16, no. 1, p. 321, 2002.

[26] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 14, no. 2, Aug. 1995, pp. 1137–1145.

[27] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[28] A. Jamain and D. J. Hand, "The naive bayes mystery: A classification detective story," *Pattern Recognition Letters*, vol. 26, no. 11, pp. 1752–1760, 2005.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[30] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, "Applications of support vector machine (svm) learning in cancer genomics," *Cancer Genomics & Proteomics*, vol. 15, no. 1, pp. 41–51, 2018.

[31] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

[32] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[33] J. Reback, W. McKinney, J. Van Den Bossche, T. Augspurger, P. Cloud, A. Klein, S. Hawkins, M. Roeschke, J. Tratner, C. She *et al.*, "pandas-dev/pandas: Pandas 1.0. 5," *Zenodo*, 2020.