



Universidade Federal de Pernambuco
Centro de Informática

Trabalho de Graduação

**Monitoramento Inteligente da Qualidade
do Óleo de Soja Utilizando Nariz
Eletrônico e Modelos de Aprendizado de
Máquina**

Carlos Eduardo Mendonça Clark

Proposta de Trabalho de Graduação
Orientador: Leandro M. Almeida
Área(s): Aprendizado de Máquina

Recife
25 de agosto de 2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Clark, Carlos Eduardo Mendonça.

Monitoramento inteligente da qualidade do óleo de soja utilizando nariz eletrônico e modelos de aprendizado de máquina / Carlos Eduardo Mendonça Clark. - Recife, 2025.

39p. : il., tab.

Orientador(a): Leandro Maciel Almeida

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado, 2025.

Inclui referências.

1. Nariz Eletrônico. 2. Aprendizado de Máquina. 3. Óleo de Soja. 4. Controle de Qualidade. I. Almeida, Leandro Maciel. (Orientação). II. Título.

000 CDD (22.ed.)

Carlos Eduardo Mendonça Clark

**Monitoramento Inteligente da Qualidade do Óleo de Soja Utilizando Nariz
Eletrônico e Modelos de Aprendizado de Máquina**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Ciência da Computação.

Recife: 12 de agosto de 2025

BANCA EXAMINADORA

Prof. Leandro Maciel Almeida (Orientador)

Universidade Federal de Pernambuco(UFPE), Centro de Informática (CIn)

Prof. Paulo Salgado Gomes de Mattos Neto (Examinador Interno)

Universidade Federal de Pernambuco(UFPE), Centro de Informática (CIn)

Resumo

O controle de qualidade do óleo de soja, insumo essencial na fabricação de margarinas, é fundamental para garantir a padronização nos processos industriais. As análises convencionais, entretanto, dependem de painéis sensoriais compostos por especialistas treinados, que classificam as amostras com base no olfato. Esse método, além de oneroso, é suscetível à subjetividade e à fadiga sensorial. Este trabalho propõe um sistema alternativo baseado em um nariz eletrônico (E-nose), capaz de capturar os perfis de compostos orgânicos voláteis (VOCs) presentes nas amostras de óleo. Os dados obtidos foram utilizados para treinar modelos de classificação com o objetivo de realizar uma avaliação automatizada da qualidade. Os resultados demonstram a eficácia da abordagem, evidenciando sua capacidade de discriminar entre diferentes classes de qualidade. Conclui-se que o uso do E-nose, aliado a modelos de classificação, representa uma ferramenta robusta e promissora, oferecendo uma alternativa rápida, objetiva e de menor custo para a análise de insumos na indústria de alimentos.

Palavras-chave: Nariz Eletrônico, Óleo de Soja, Controle de Qualidade, Aprendizado de Máquina

Sumário

Introdução	1
1.1 Objetivos	1
Fundamentação Teórica	3
2.1 Óleo de Soja: Processos e Controle de Qualidade	3
2.2 E-Nose	4
2.3 Modelos para Classificação	5
2.3.1 Modelos de aprendizado de máquina supervisionada	6
2.3.2 Modelos de séries temporais	7
2.4 Métricas	7
2.5 Conjunto de dados	8
2.6 Calibração dos modelos	9
Metodologia	10
3.1 Natureza dos dados	10
3.1.1 Amostras e processo de coleta	10
3.1.2 sensores do E-nose	11
3.1.3 Formato da base de dados	13
3.1.4 Colunas	13
3.2 Pré processamento	14
3.2.1 Escolha de colunas	14
3.2.2 Divisão dos dados em ciclos	14
3.2.3 Normalização	15
3.2.4 Divisão de treinamento e teste	15
3.2.5 Visualização Assinaturas Olfativas	17
3.2.6 Transformação das assinaturas para dados 3D	17
3.2.7 Escolha dos sensores	17
Resultados	19
4.1 Modelagem	19
4.2 Experimentos	19
4.2.1 Conjunto Treinamento: CA - Teste CI	20
4.2.2 Análise dos resultados	22
4.2.2.1 Resultado dos 20 melhores modelos na base de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.	22

4.2.2.2	Resultado dos 20 melhores modelos na base de teste, na base de teste. Tabela ordenada por modelo e N sensores	23
4.2.3	Conjunto Treinamento: CA, CI, CNA - Teste: CA, CI, CNA	23
4.2.4	Análise dos resultados	25
4.2.4.1	Resultado dos 20 melhores modelos na base de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.	26
4.2.4.2	Resultado dos 20 melhores modelos na base de teste, na base de testes. Tabela ordenada por modelo e N sensores.	27
4.2.4.3	Resultado dos 15 melhores modelos na fase de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.	27
4.2.4.4	Resultado dos 15 melhores modelos na fase de teste, na base de teste. Tabela ordenada por modelo temporal e N sensores.	28
4.2.5	Conjunto Treinamento: CI - Teste CI	28
4.2.5.1	Resultados métodos de escolha dos sensores	28
4.2.6	Análise dos resultados	30
4.2.6.1	Resultado dos 20 melhores modelos na base de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.	32
4.2.6.2	Resultado dos 20 melhores modelos na base de teste, na base de testes. Tabela ordenada por modelo e N sensores.	33
4.2.6.3	Resultado dos 15 melhores modelos na fase de teste, na base de treinamento. Tabela ordenada por modelo temporal e N sensores.	34
4.2.6.4	Resultado dos 15 melhores modelos na fase de teste, na base de teste. Tabela ordenada por modelo temporal e N sensores.	35
Conclusão		36
5.1	Trabalhos Futuros	37
Referências Bibliográficas		38

Introdução

A qualidade do óleo de soja é um fator crítico para o consumo humano. Além de suas propriedades nutricionais, a qualidade, também abrange a sustentabilidade do processo de produção, a eficiência da operação e conformidade com normas e regulamentações internacionais. Nesse contexto, a gestão de qualidade se torna essencial para garantir a segurança alimentar, atender às exigências do mercado e manter a competitividade[1]. Para isso, existe um controle rigoroso sobre as propriedades físico-químicas e sensoriais do produto.

A OMS e a FDA estabeleceram parâmetros analíticos para garantir a segurança do consumo de óleos comestíveis. Métodos como valor de acidez, saponificação, índice de iodo, peróxidos, teor de água, estabilidade oxidativa, ácidos graxos livres e detecção de níquel por polarografia são alguns dos métodos utilizados para avaliar a qualidade desses produtos[2]. No entanto, também existe a utilização de métodos sensoriais, como a análise olfativa, especialmente em contextos onde a presença de compostos voláteis é indicativa de degradação. A percepção de odores pode ser um sinal precoce de oxidação ou contaminação, sendo, portanto, uma das ferramentas fundamentais na avaliação da qualidade do óleo[3].

Embora o painel sensorial treinado (indivíduos capacitados para avaliar alimentos com base em características sensoriais) seja amplamente utilizado na classificação da qualidade de óleos, esse método tem grandes limitações. Por depender da percepção humana, está sujeito à individualidade, o que pode comprometer a precisão do diagnóstico. Além disso, envolve altos custos operacionais, exige uma infraestrutura adequada, treinamento contínuo e não é facilmente escalável para análises em larga escala. Diante dessas limitações, torna-se necessária a adoção de abordagens alternativas que ofereçam maior confiabilidade e padronização.

O uso de sensores inteligentes em conjunto de técnicas de inteligência artificial tem se destacado como uma alternativa promissora para classificar e identificar comida [4], possibilitando substituir ou complementar métodos convencionais de análise. Essa configuração representa um avanço nesse campo, permitindo diagnósticos mais rápidos, objetivos e consistentes. Além disso, permitem a integração de sistemas automatizados de controle de qualidade, promovendo maior segurança alimentar.

1.1 Objetivos

Neste contexto, este trabalho vem com a finalidade do desenvolvimento de uma solução escalável, padronizada, de baixo custo e de fácil aplicabilidade. A abordagem será feita a partir de um nariz eletrônico (E-nose), projetado para mimetizar o funcionamento do olfato humano. Esse dispositivo será responsável pela coleta de dados de específicos tipos de compostos voláteis

presentes no óleo de soja, os quais servirão para o treinamento de um modelo de aprendizado de máquina capaz de classificar automaticamente a qualidade do óleo.

A metodologia do projeto será a partir um pré-processamento dos dados coletados pelo E-nose, seguido por uma análise detalhada para identificar a importância de cada sensor na classificação. Em seguida, os dados serão divididos entre dados de treinamento e dados de testes. Uma bateria de modelos de aprendizado de máquina será treinada, com a busca de melhores parâmetros para cada um deles. A principal métrica de avaliação utilizada será a acurácia na tarefa de classificação da qualidade do óleo.

Este trabalho está organizado em capítulos a seguir para tratar os aspectos citados. O Capítulo 2, intitulado "Fundamentação Teórica", expandirá a compreensão sobre o óleo de soja, o nariz eletrônico (E-nose), os modelos de aprendizado de máquina clássicos e de séries temporais para classificação, as métricas de avaliação, o conjunto de dados e a calibração dos modelos. O Capítulo 3, intitulado "Metodologia", descreverá a natureza dos dados, abrangendo as amostras e o procedimento de coleta, os sensores do E-nose, a estrutura da base de dados e suas colunas, além das fases de pré-processamento e da seleção dos sensores. Por fim, o Capítulo 4, intitulado "Resultados", descreverá a modelagem empregada e os experimentos realizados, abrangendo as diversas configurações dos conjuntos de treinamento e teste, bem como as avaliações de desempenho. O Capítulo 5, "Conclusão", encerrará o trabalho, resumindo as descobertas e contribuições da pesquisa.

Fundamentação Teórica

Neste capítulo, abordaremos com mais profundidade o óleo de soja, o nariz eletrônico, alguns modelos de aprendizado de máquina clássicos para classificação, algumas métricas conhecidas, divisão do conjunto de dados e calibração dos modelos. Esses conhecimentos serão importantes para a compreensão da metodologia apresentada posteriormente.

2.1 Óleo de Soja: Processos e Controle de Qualidade

Os óleos vegetais são amplamente utilizados na alimentação humana, tanto em preparações in natura quanto em processos de cocção. Conforme mencionado na introdução, diversos parâmetros analíticos são utilizados internacionalmente para garantir qualidade e segurança dos óleos cosméticos, incluindo o óleo de soja. Nesta seção, serão detalhados os principais métodos, suas finalidades e limitações, de forma a contextualizar, sendo elas então [5]:

- **Valor de Hidróxila:** Mede quantidade de grupos hidróxila.
- **Índice de acidez:** Mede quantidade de ácidos graxos livres. O aumento do índice de acidez e da concentração de ácidos graxos livres está diretamente relacionado à queda na qualidade do óleo.
- **Índice de saponificação:** Um valor elevado indica que os ácidos graxos do óleo possuem menor peso molecular.
- **Índice de iodo:** Serve como um indicador da presença de ácidos graxos insaturados em óleos e gorduras. Quanto maior a quantidade de gorduras insaturadas, mais o iodo reage com ligações duplas, resultando num índice de iodo elevado.
- **Peróxidos:** Informa sobre a idade e qualidade do óleo. Valores mais baixos indicam óleo mais novo e menos oxidado.
- **Teor de água:** Mede quantidade de água no composto, a qual deve estar dentro de limites adequados para garantir estabilidade.
- **Estabilidade oxidativa:** Técnica aplicada para simular o envelhecimento do composto. Durante o procedimento, uma corrente de ar é direcionada ao óleo mantido a uma temperatura elevada constante, provocando a oxidação dos ácidos graxos. Ao fim do teste, alguns produtos secundários voláteis são formados e transportados pelo fluxo de ar para uma solução de análise (água deionizada), provocando um pico de condutividade elétrica

que é registrada continuamente. O tempo de detecção até reação do produto secundário ser formado e detectado é chamado de tempo de indução, caracterizando a estabilidade oxidativa.

- **Detecção de Níquel:** Na produção de margarina, é comum solidificar o óleo através da hidrogenação catalítica dos ácidos graxos e o níquel é um dos catalisador utilizados para esse processo. A detecção da quantidade de níquel no produto final é comum para classificar a qualidade.

Um dos principais fatores que determinam a qualidade desses óleos é a presença de Compostos Orgânicos Voláteis (VOCs, do inglês Volatile Organic Compounds). Esses compostos são moléculas orgânicas de baixa massa molecular e alta pressão de vapor, o que lhes permite evaporar facilmente à temperatura ambiente. No óleo de soja os VOCs são produtos da degradação química, principalmente por processos de oxidação e hidrólise. Os principais produtos dessa oxidação incluem aldeídos, cetonas, ácidos e outros compostos orgânicos, os quais não apenas alteram negativamente o sabor característico e atividade funcional, mas também podem representar riscos à saúde humana[6].

Cada um desses VOCs é responsável por aromas e odores característicos, agradáveis ou indesejáveis, associados à qualidade do óleo. É possível por meio de sensores, capturar níveis desses compostos em cada amostra. Mesmo produtos da mesma marca podem possuir os mesmos VOCs, mas níveis de concentrações diferentes de cada um deles, o que torna cada amostra ser única e possibilita a criação de uma espécie de "assinaturas olfativas" específica para a qualidade de cada amostra. A partir dessa assinatura podemos categorizar e classificar os compostos de acordo com padrões previamente definidos.

2.2 E-Nose

O sistema olfativo permite que os seres vivos estejam atentos ao ambiente, percebam perigos e auxiliem na identificação e classificação de alimentos [7]. Inspirado nesse sistema biológico, o nariz eletrônico (E-nose) foi desenvolvido com o objetivo de simular artificialmente a capacidade de detectar, distinguir e classificar compostos voláteis presentes no ambiente. Esse dispositivo já vem sendo amplamente utilizado em diversas áreas da pesquisa como o da saúde, com aplicações para o diagnóstico médico.

Na área da saúde já existem trabalhos que pesquisam a utilização do nariz eletrônico para diagnóstico rápido de condições ginecológicas e também para candidíase [8] [9]. Além disso, há diversas pesquisas em andamento explorando seu uso em diferentes contextos. O nariz eletrônico também já é amplamente utilizado comercialmente em diversas áreas. Dentre as principais, são destacadas: Segurança pública, monitoramento ambiental, detecção de compostos perigosos e explosivos, controle de qualidade na indústria alimentícia e de bebidas, embalagem, monitoramento em ambientes internos e externos, detecção de frescor de carnes, aves e peixes, monitoramento industrial e doméstico, cuidados pessoais, cosméticos, setor automotivo e monitoramento em ambientes clínicos e hospitalares[4].

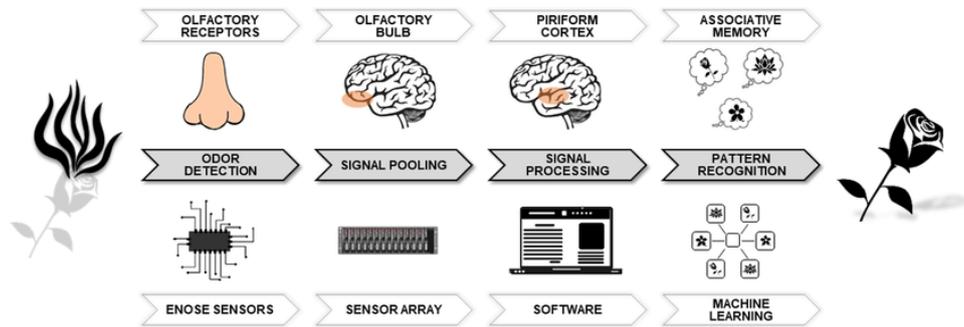


Figura 2.1: Comparação esquemática entre a tecnologia de nariz eletrônico (eNose) e o sistema olfativo biológico. [10].

A mimetização do olfato biológico pelo E-nose ocorre por meio da coleta dos VOCs liberados por substâncias de interesse. Essas substâncias são direcionadas a uma câmara que contém uma série de sensores químicos, normalmente à base de óxidos metálicos, cada um com diferentes graus de seletividade e sensibilidade. Isso possibilita a identificação e quantificação exata dos VOCs presentes no material analisado.

Ao interagirem com os sensores, os compostos produzem uma "assinaturas olfativas"[9]. Então, essa assinatura é tratada por algoritmos de computação, que podem incluir métodos de aprendizado de máquina, redes neurais e classificadores de séries temporais. Com base nesse processamento, o sistema pode comparar a assinatura obtida com padrões já armazenados, possibilitando a categorização da substância examinada e a geração de um diagnóstico ou identificação correspondente.

2.3 Modelos para Classificação

Foi abordado os dados que iremos avaliar e o hardware que será utilizado, e agora é a hora de falarmos sobre o software que será responsável por ajudar o nariz eletrônico a classificar os diferentes aromas de forma eficiente e precisa. Existem muitos métodos definidos para classificação na literatura, nesse trabalho optamos com modelos clássicos de aprendizado de máquina supervisionada para classificação. Serão utilizados: Decision Tree, RandomForest, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Naive Bayes (Gaussiano), Multi Layer Perceptron (MLP) e regressão logística(logistic regression). Esses modelos são comumente utilizados em problemas de classificação devido à sua eficácia, facilidade de implementação, e, para alguns deles, interpretabilidade.

Além dos modelos de aprendizado de máquina supervisionada, também utilizaremos modelos de específicos para séries temporais. A justificativa para essa abordagem surge devido a natureza do nosso conjunto de dados, como trabalharemos com "assinaturas olfativas" que são basicamente a evolução dos VOCs ao longo do tempo de coleta. Essas assinaturas apresentam padrões e variações temporais que não podem ser totalmente capturados por outros tipos de modelos. Modelos de séries temporais são capaz de explorar esses padrões e variações ao longo do tempo o que pode levar a um melhor desempenho que os modelos clássicos utilizados[11]. A escolha dos modelos de série temporais são: SignatureClassifier, KNNTIME-

Series(KNeighborsTimeSeriesClassifier) e RocketClassifier.

2.3.1 Modelos de aprendizado de máquina supervisionada

Decision Tree: O objetivo deste modelo é a tentativa de separação das classes a partir de um conjunto de regras hierárquicas, também conhecida como árvore, e cada divisão desses nós dessa árvore é uma regra aplicada sobre uma feature específica, são bastantes simples de entender e interpretar pois as árvores podem ser visualizadas, são também capazes de lidar com problemas de múltiplas saídas e é considerado um modelo de caixa branca[12].

RandomForest: Modelo de ensemble que emprega diversas árvores de decisão para aprimorar a precisão e robustez. Durante o treinamento dessas árvores, as bases de treinamento são subdivididas em amostras de maneira aleatória e com reposição por meio do método de bootstrap, em seguida são utilizadas para treinar árvores que a compõem. Além disso, cada divisão de nó, apenas um subconjunto de variáveis é considerado, aumentando ainda mais a diversidade entre as árvores o que acaba proporcionando uma redução ao principal problema da árvore de decisão que é o risco de overfitting[13].

Support Vector Machine (SVM): O objetivo desse modelo é fazer a distinção das classes através da separação de classes via um hiperplano de margem máxima que é uma margem de segurança entre os pontos mais próximos ao hiperplano que garante uma generalização mais robusta. Por natureza o SVM é um classificador binário, mas existe algumas técnicas chamadas OvO (One versus One) ou OvR (One versus Rest) que conseguem por si só transforma-lo num classificador de multiplas classes[14].

k-Nearest Neighbors (k-NN): Modelo de aprendizado relativamente simples, cuja premissa é: Dado uma nova entrada, verificar os k vizinhos mais próximos e classificar a nova entrada com base na moda entre os k vizinhos vizinhos mais próximos. A maneira de se calcular os vizinhos mais próximos é bem exaustiva pois terá que medir a distancia entre a nova amostra e todas as da base de treinamento. Utilizam-se métodos de distância ponto a ponto, como, por exemplo, a distância euclidiana[14].

Naive Bayes (Gaussiano): Os métodos classificação Naive Bayes são um conjunto de modelos de classificação que utilizam a suposição de que as variáveis são condicionalmente independentes entre si para inferir a classe. O modelo Gaussiano foi o escolhido e basicamente implica que a probabilidade das características seja Gaussiana[15].

Multi Layer Perceptron (MLP): A MLP é a única dos modelos escolhidos que se classifica como uma rede neural artificial. As MLPs são compostas por camadas de neurônios categorizadas por: Camada de entrada, camada(s) ocultas e uma camada de saída. Cada neurônio que faz parte da camada oculta, realiza uma espécie de combinação ponderada dos valores de camada anterior. A camada de saída recebe os valores da última camada oculta e transforma-os num valor de classificação conhecida. São capazes de aprender funções complexas e encontrar relações não lineares entre as variáveis de entrada[14][16].

LogisticRegression: Modelo linear empregado em problemas de classificação binária e multiclasse que calcula a probabilidade de uma amostra ser classificada em uma determinada categoria utilizando a função logística (sigmoide). A regressão logística procura determinar os coeficientes das variáveis independentes que mais se adequam aos dados, maximizando a

probabilidade da classe[13].

2.3.2 Modelos de séries temporais

SignatureClassifier: O SignatureClassifier basicamente utiliza um método nos dados, conhecido como SignatureTransformer. Esse método consegue extrair representações numéricas importantes da trajetória temporal desses dados, que descrevem como as variáveis evoluem e interagem ao longo do tempo, preservando informações sobre ordem e dependência temporal. Em seguida, após essa transformação, o modelo é passado para um preditor clássico, como por exemplo: RandomForest, SVM, Regressão Logística[17].

KNeighborsTimeSeriesClassifier: Variação do adaptada do KNN para séries temporais. Em vez de calcular distância ponto a ponto que nem o KNN normal ele calcula a distância das curvas[18].

RocketClassifier: Modelo transforma os dados de entrada utilizando o algoritmo de ROCKET(RandOm Convolutional KErnel Transform) que gera uma grande variedade de kernels convolucionais, após isso realiza uma padronização(Standard Scaling) e treina com um modelo de classificação clássico utilizando o dado transformado[19].

2.4 Métricas

Utilizando esses algoritmos, poderemos de maneira abrangente elaborar uma comparação dos modelos tradicionais de aprendizado de máquina. Cada modelo contém particularidades quanto a tempo de treinamento, interpretabilidade, sensibilidade e generalização. Após o treinamento, iremos utilizar métricas conhecidas, como acurácia e F1-score, possibilitando a avaliação do desempenho de cada modelo, portanto, escolhendo o método mais adequado para o nosso contexto do nariz eletrônico.

		POSITIVE	NEGATIVE
		ACTUAL VALUES	
ACTUAL VALUES	POSITIVE	TP	FN
	NEGATIVE	FP	TN

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figura 2.2: Matriz de confusão: Precisão, Sensibilidade, Acurácia e F1-Score [20]

Acurácia: É a porcentagem das respostas corretas do modelo dividido pelo número de total de predições. Alta acurácia não necessariamente significa que a qualidade do modelo está boa, sensível a dados de treinamento não balanceados.

Precisão: Também conhecido como valor preditivo positivo, consegue quantificar a proporção de predições positivas verdadeiras entre todas as predições positivas do modelo. É especialmente importante quando falsos positivos são considerados críticos, como em diagnósticos médicos ou sistemas de segurança.

Recall: Também conhecida como sensibilidade mede a proporção de casos positivos reais que foram corretamente identificados pelo modelo. É uma métrica crucial em cenários onde não detectar um caso positivo pode ser crítico, como em doenças e detecção de fraudes.

F1-score: É a média harmônica entre precisão e recall. É particularmente útil quando existe desequilíbrio entre as classes e busca um compromisso entre evitar falsos positivos e falsos negativos.

Utilizando todas essas ferramentas descritas, esperamos poder atingir uma solução para a classificação automatizada dos aromas.

2.5 Conjunto de dados

Discutiremos um assunto essencial ligado à divisão do conjunto de dados. É fundamental dividir os dados em conjuntos diferentes para assegurar que o modelo aprenda de maneira eficaz e que sua performance seja avaliada de forma precisa. Geralmente, essa divisão é realizada em treinamento, validação e teste, apesar de não haver uma norma estrita quanto às proporções. Normalmente, recomenda-se reservar de 20% a 30% dos dados para testes e validação, enquanto o restante é utilizado para o treinamento.

O conjunto de treinamento é empregado para calibrar os parâmetros do modelo, ou seja, é o momento em que o modelo assimila os padrões contidos nos dados. O conjunto de validação é usado na fase de desenvolvimento para avaliar várias configurações de hiperparâmetros, sem empregar os dados de teste nesse procedimento. Finalmente, o conjunto de teste é formado por dados totalmente inéditos para o modelo, utilizados para avaliar seu desempenho final. Para simular o funcionamento prático do modelo, é fundamental que esse conjunto esteja equilibrado em relação às classes e que represente a distribuição real dos dados.

Há também uma estratégia alternativa, denominada validação cruzada, que pode ser utilizada para evitar a exigência de uma divisão fixa em três conjuntos. Nesse método, os dados são separados apenas em conjuntos de treinamento e teste, e a validação ocorre internamente no conjunto de treinamento. Isso possibilita a otimização do uso dos dados, ampliando a quantidade disponível para treinamento e assegurando uma avaliação mais sólida do modelo.

A validação cruzada opera da seguinte maneira: o conjunto de treinamento é dividido em várias “partições” ou folds de tamanho aproximado. Em cada iteração, um fold, ou dobra, é utilizado como conjunto de validação, ao passo que os outros folds são empregados para treinar o modelo. Esse procedimento continua até que todos os folds tenham sido usados como validação uma vez. Por fim, os resultados são combinados para oferecer uma estimativa precisa do desempenho do modelo e auxiliar na seleção dos hiperparâmetros mais adequados.



Figura 2.3: Representação gráfica da validação cruzada dividido em 5 Folds

Assim, a validação cruzada possibilita um melhor aproveitamento dos dados disponíveis e diminui a probabilidade de sobreajuste, tornando a avaliação mais estável e representativa para questões como a classificação de assinaturas olfativas em nariz eletrônico.

2.6 Calibração dos modelos

Como já foi mencionado, a calibração dos modelos é feita com os dados de validação ou por meio de validação cruzada. Os diferentes modelos possuem características distintas no que diz respeito ao processo de calibração, que tem como objetivo alinhar as probabilidades estimadas com as chances reais de ocorrência das classes.

Por exemplo, árvores de decisão, RandomForest e SVM podem gerar probabilidades que não estão bem calibradas, geralmente sendo muito confiantes ou cautelosas. Calibrar é ajustar esse desvio, para que um modelo de acurácia 70%, por exemplo, resulte em uma acurácia de 85%.

Para descobrir a configuração ideal dos hiperparâmetros durante o treinamento e a calibração, geralmente usamos abordagens como Grid Search, que busca de forma exaustiva em um espaço definido de parâmetros a combinação que maximiza o desempenho do modelo.

Dessa forma, a calibração se torna uma etapa essencial para que os modelos ganhem maior confiança e para que as decisões tomadas com base nas probabilidades estimadas estejam mais alinhadas com a realidade observada.

Metodologia

A metodologia seguida pode ser dividida em 3 etapas principais: conforme ilustrado na figura abaixo. A primeira etapa corresponde a natureza dos dados, na qual abordaremos a amostras, coleta. Já na segunda etapa falaremos sobre o pré-processamento e por último a escolha dos sensores.

3.1 Natureza dos dados

3.1.1 Amostras e processo de coleta



Figura 3.1: Amostras analisadas de óleo de soja

As amostras recolhidas foram inicialmente agrupadas em 14 categorias (que, com o aquecimento, se transformam em 27, h_forte não aquecido não teve medição), correspondendo a distintas características sensoriais do óleo e à variação de temperatura. Os nomes das categorias foram codificados para preservar a confidencialidade dos dados, mas abrangem variações como: AC, AF, HE, HI e O, subdivididas em graus de intensidade (leve, moderado e forte) e também avaliadas em temperaturas ambiente (24 °C) e também avaliadas após serem submetidas a aquecimento a 50 °C para potencializar a emissão dos compostos orgânicos voláteis (VOCs). Colocadas de forma individual em placas de Petri descartáveis, garantindo a ausência de contaminação cruzada.

	Leve	Moderado	Forte	Leve 50°C	Moderado 50°C	Forte 50°C
AC						
AF						
HE						
HI						
O						

Figura 3.2: Distribuição das amostras segundo tipo, intensidade e temperatura

Cada amostra contém 100 μL de óleo, sendo submetida a análise num ambiente climatizado e controlado, garantindo padronização. Cada placa era inserida individualmente no sistema de coleta do E-nose que realiza a leitura dos VOCs presentes no ar liberado pelo óleo.

3.1.2 sensores do E-nose

Ao todo, foram empregados 10 sensores na câmara de coleta, dos quais 7 tinham a função de capturar VOCs: TGS826 (Ohm), TGS2611 (Ohm), TGS2603 (Ohm), TGS813 (Ohm), TGS822 (Ohm), TGS2602 (Ohm) e TGS823 (Ohm). O remanescente abrange a captura de temperatura, pressão e umidade. Na tabela abaixo é possível entender um pouco sobre o funcionamento dos sensores de VOCs:

Sensor	Main Function
TGS826	Ammonia detection
TGS2611-E00	Methane detection
TGS2603	Detection of odors and air contaminants (High sensitivity to series of amines and gases with sulfurous odor and high sensitivity to food odors)
TGS813	Detection of combustible gases (High sensitivity to methane, propane, and butane)
TGS822	Detection of Solvent Vapors (High sensitivity to alcohol and organic solvent)
TGS2602	Detection of air contaminants (High sensitivity to gaseous air contaminants)
TGS823	Detection of Vapors from Organic Solvents (High sensitivity to vapors from organic solvents such as ethanol)

Figura 3.3: Sensores utilizados para capturar os VOCs das amostras de Óleo. Tabela de sensores de [9]

Durante o processo de coleta, cada amostra passava por três etapas principais:

1. Sucção do ar contendo os VOCs da amostra para a câmara de sensores, onde as moléculas começavam a interagir com os sensores químicos do dispositivo.
2. Estabilização da massa de ar na câmara de sensores, etapa essencial para garantir a precisão das leituras.
3. Purga do sistema, onde o equipamento era limpo com ar filtrado para eliminar quaisquer resíduos da amostra anterior.

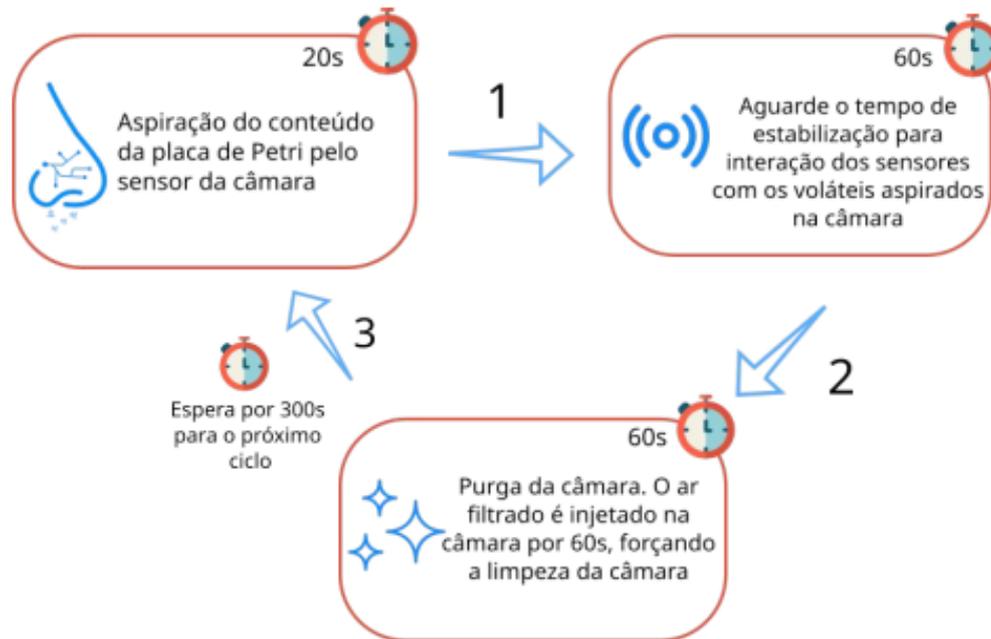


Figura 3.4: Ciclo de coleta

Este ciclo de análise era repetido várias vezes para cada amostra, permitindo a obtenção de múltiplas leituras de ciclos e consequentemente aumentando a quantidade de dados para maior robustez na hora de treinamento.

3.1.3 Formato da base de dados

Ao todo, há 44 arquivos .csv de dados, e o nome de cada arquivo indica a classe a que os dados se referem. O conjunto de dados utilizado neste trabalho foi dividido em três partes: Dois conjuntos, um aquecido e outro sem ser aquecido, foram construídos a partir de amostras analisadas na Universidade Federal de Pernambuco (UFPE), fornecidas pela empresa fabricante de margarina. Essas amostras foram previamente classificadas com base em atributos sensoriais definidos pela própria fabricante, permitindo o desenvolvimento e o ajuste dos modelos de análise e classificação. O terceiro conjunto, por sua vez, é composto por um novo lote de amostras de óleo coletadas diretamente na planta industrial da empresa. Essas amostras foram disponibilizadas após o início do projeto. Dessa forma, o conjunto representa uma base independente e realista, permitindo avaliar a capacidade de generalização dos modelos desenvolvidos em um ambiente de produção real.

3.1.4 Colunas

Os dados brutos contém 13 colunas, as duas primeiras colunas identificam dados da bomba, a terceira coluna indica a hora que o sensor havia coletado aqueles valores, as colunas 4 até 10 são de origem dos sensores de captura de VOCs na ordem apresentada anteriormente, o remanescente em ordem, temperatura, pressão e humidade. Todas as colunas são do tipo flutuante.

```

##### CICLO_TIMESTAMP #####
50000.0 50000.0 1726264557.5387301 551e+05 7.385e+03 6.820e+03 8.14e+04 3.720e+03 6.567e+04 7.231e+03 59.29 69.15 7.28
| 50000.0 1726264558.508223 551e+05 7.385e+03 6.820e+03 8.320e+04 3.770e+03 7.802e+04 7.983e+03 59.25 69.33 7.50
50000.0 50000.0 1726264558.508187 550e+05 7.385e+03 6.820e+03 7.871e+04 3.674e+03 6.249e+04 7.322e+03 59.38 69.42 6.96
50000.0 50000.0 1726264559.5448048 572e+05 7.385e+03 6.820e+03 7.735e+04 3.628e+03 5.976e+04 7.358e+03 59.31 69.15 6.97
50000.0 50000.0 1726264560.5086243 579e+05 7.385e+03 6.820e+03 7.627e+04 3.594e+03 5.786e+04 7.362e+03 59.38 69.42 6.93
50000.0 50000.0 1726264561.5451314 583e+05 7.385e+03 6.820e+03 7.551e+04 3.563e+03 5.622e+04 7.395e+03 59.31 69.42 6.74
50000.0 50000.0 1726264562.4991138 581e+05 7.385e+03 6.820e+03 7.588e+04 3.548e+03 5.511e+04 7.387e+03 59.38 69.38 6.85
.
.
.
##### BOMBA_SESLIGAD_#####
50000.0 50000.0 1726264571.5347762 552e+05 7.385e+03 6.820e+03 7.695e+04 3.412e+03 5.097e+04 7.852e+03 59.26 69.34 6.73
50000.0 50000.0 1726264572.486848 552e+05 7.385e+03 6.820e+03 7.722e+04 3.438e+03 5.079e+04 7.848e+03 59.25 69.46 6.82
50000.0 50000.0 1726264573.5358275 559e+05 7.385e+03 6.820e+03 7.747e+04 3.451e+03 5.066e+04 7.849e+03 59.24 69.42 6.77
50000.0 50000.0 1726264574.4872304 558e+05 7.385e+03 6.820e+03 7.781e+04 3.434e+03 5.051e+04 7.853e+03 59.23 69.36 6.84
50000.0 50000.0 1726264575.5278357 552e+05 7.385e+03 6.814e+03 7.821e+04 3.412e+03 5.046e+04 7.841e+03 59.21 69.38 6.63
##### BOMBA_SESLIGAD_#####
50000.0 50000.0 1726264577.4966064 551e+05 7.385e+03 6.814e+03 7.891e+04 3.401e+03 5.048e+04 7.832e+03 59.43 69.99 7.73
50000.0 50000.0 1726264578.4882397 551e+05 7.385e+03 6.814e+03 8.157e+04 3.387e+03 5.029e+04 7.847e+03 59.48 69.87 8.13
50000.0 50000.0 1726264579.5328554 551e+05 7.385e+03 6.820e+03 8.406e+04 3.345e+03 5.049e+04 7.848e+03 59.37 69.95 8.28
.
.
.
##### BOMBA_SESLIGAD_#####
50000.0 50000.0 1726264601.5454038 557e+05 7.385e+03 6.820e+03 5.488e+04 3.316e+03 7.288e+04 7.957e+03 59.31 69.03 8.42
50000.0 50000.0 1726264604.4465677 523e+05 7.385e+03 6.820e+03 5.488e+04 3.320e+03 7.391e+04 7.959e+03 59.31 69.03 8.47
50000.0 50000.0 1726264605.5454514 523e+05 7.385e+03 6.820e+03 5.488e+04 3.322e+03 7.334e+04 7.968e+03 59.31 69.99 8.48
50000.0 50000.0 1726264606.4558802 523e+05 7.385e+03 6.820e+03 5.488e+04 3.312e+03 7.387e+04 7.964e+03 59.31 69.03 8.48
##### BOMBA_SESLIGAD_#####
50000.0 50000.0 1726264611.5392263 551e+05 7.385e+03 6.820e+03 6.732e+04 3.623e+03 5.116e+04 7.966e+03 58.78 69.56 6.77
50000.0 50000.0 1726264614.429195 503e+05 7.385e+03 6.820e+03 8.316e+04 3.494e+03 5.216e+04 7.957e+03 58.52 69.64 6.52
50000.0 50000.0 1726264615.556899 507e+05 7.385e+03 6.820e+03 8.024e+04 3.377e+03 4.897e+04 7.942e+03 58.48 69.52 6.53
50000.0 50000.0 1726264616.4532597 579e+05 7.385e+03 6.820e+03 7.833e+04 3.174e+03 4.642e+04 7.931e+03 58.25 69.64 6.44
.
.
.
##### BOMBA_SESLIGAD_#####
50000.0 50000.0 1726264618.4936087 551e+05 7.385e+03 6.808e+03 7.588e+04 3.395e+03 4.653e+04 7.939e+03 58.59 69.46 6.51
50000.0 50000.0 1726264619.499594 551e+05 7.385e+03 6.814e+03 7.588e+04 3.484e+03 4.643e+04 7.958e+03 58.96 69.66 6.54
50000.0 50000.0 1726264620.4925347 551e+05 7.385e+03 6.814e+03 7.591e+04 3.473e+03 4.638e+04 7.957e+03 58.93 69.76 6.52
50000.0 50000.0 1726264621.422666 551e+05 7.385e+03 6.814e+03 7.595e+04 3.462e+03 4.629e+04 7.951e+03 58.93 69.76 6.55
##### BOMBA_SESLIGAD_#####
50000.0 50000.0 1726264625.508123 551e+05 7.385e+03 6.814e+03 7.644e+04 3.375e+03 4.636e+04 7.970e+03 58.82 69.89 6.28

```

Figura 3.5: Print de um ciclo antes de ser processado. Os 3 pontos significam que existem dados entre as imagens.

3.2 Pré processamento

Utilizaremos a mesma abordagem do artigo [9] na etapa de pre processamento adicionando, entretanto, algumas etapas a mais. Todo o processamento elaborado feito igual ao artigo são os seguintes subtopicos seguintes: **3.2.1 Escolha de colunas**, **3.2.2 Divisão dos dados em ciclos**.

3.2.1 Escolha de colunas

As colunas escolhidas primariamente serão apenas de todos os sensores. Ou seja, removeremos as colunas de bomba(0,1), timestamp, temperatura, pressão e humidade. O motivo da remoção dessas colunas é que se mostraram poucos relevantes para o domínio problema.

3.2.2 Divisão dos dados em ciclos

Para obtermos a "assinatura olfativa" anteriormente citada, devemos concatenar todas as leituras de todos os sensores de um ciclo numa única linha do dataset, agora cada amostra está relacionada a um ciclo. Vale ressaltar que nem todo ciclo contém o mesmo tamanho devido a inconsistência da leitura dos dados pelo E-nose.

Para resolver essa questão, é necessário uniformizar o tamanho de todos os ciclos. A abordagem adotada consiste em identificar o ciclo de menor tamanho dentre todas as classes e utilizá-lo como referência. Caso um ciclo apresente um tamanho superior ao estabelecido, ele será "podado", sendo considerado apenas da sua origem até o limite definido.

3.2.3 Normalização

Os dados flutuantes de cada coluna apresentavam uma discrepância significativa de magnitude, o que poderia dificultar a visualização dos dados e o aprendizado em algum dos modelos selecionados. A normalização selecionada foi a Min-Max, uma vez que, embora seja simples, resolveu todas as questões pertinentes.

3.2.4 Divisão de treinamento e teste

O experimento proposto será uma abordagem com diversas distribuições de treinamento e teste. Utilizaremos as seguintes nomenclaturas para os 3 conjuntos de dados, Conjunto aquecido - CA, Conjunto coletado e analisado indústria - CI, Conjunto não aquecido = CNA

Faremos as seguintes divisões:

1. Treinamento: CA - Teste: CI

Classe	Ciclos
ac_leve	6
ac_forte	6
af_leve	6
af_moderado	6
af_forte	6
he_leve	6
he_moderado	6
he_forte	6
hi_leve	6
hi_moderado	6
hi_forte	6
ox_leve	6
ox_moderado	6
ox_forte	6
pa	6

Tabela 3.1: Base de treinamento aquecida

Classe	Ciclos
ac_leve	6
ac_forte	6
af_leve	6
af_moderado	6
af_forte	6
he_leve	6
he_moderado	6
he_forte	6
hi_leve	6
hi_moderado	6
hi_forte	6
ox_leve	6
ox_moderado	6
ox_forte	6
pa	6

Tabela 3.2: Base de teste CI

2. Treinamento: CA, CI, CNA - Teste: CA, CI, CNA. Utilizaremos validação cruzada dividindo o conjunto em 80% para treinamento e 20% para teste. Nesse conjunto houve uma unificação das classes aquecidas e não aquecidas.

Classe	Ciclos
ac_leve	15
ac_forte	14
af_leve	14
af_moderado	14
af_forte	14
he_leve	14
he_moderado	13
he_forte	14
hi_leve	14
hi_moderado	14
hi_forte	10
ox_leve	15
ox_moderado	15
ox_forte	14
pa	14

Tabela 3.3: Base de treinamento 80% da base

Classe	Ciclos
ac_leve	3
ac_forte	3
af_leve	4
af_moderado	4
af_forte	4
he_leve	4
he_moderado	3
he_forte	4
hi_leve	4
hi_moderado	4
hi_forte	2
ox_leve	3
ox_moderado	3
ox_forte	4
pa	4

Tabela 3.4: Base de teste 20%

3. Treinamento: CI - Teste: CI. Utilizaremos validação cruzada dividindo o conjunto em 2/3 para treinamento e 1/3 para teste.

Classe	Ciclos
ac_leve	4
ac_forte	4
af_leve	4
af_moderado	4
af_forte	4
he_leve	4
he_moderado	4
he_forte	4
hi_leve	4
hi_moderado	4
hi_forte	4
ox_leve	4
ox_moderado	4
ox_forte	4
pa	4

Tabela 3.5: Base de treinamento (2/3 da base)

Classe	Ciclos
ac_leve	2
ac_forte	2
af_leve	2
af_moderado	2
af_forte	2
he_leve	2
he_moderado	2
he_forte	2
hi_leve	2
hi_moderado	2
hi_forte	2
ox_leve	2
ox_moderado	2
ox_forte	2
pa	2

Tabela 3.6: Base de teste (1/3 da base)

3.2.5 Visualização Assinaturas Olfativas

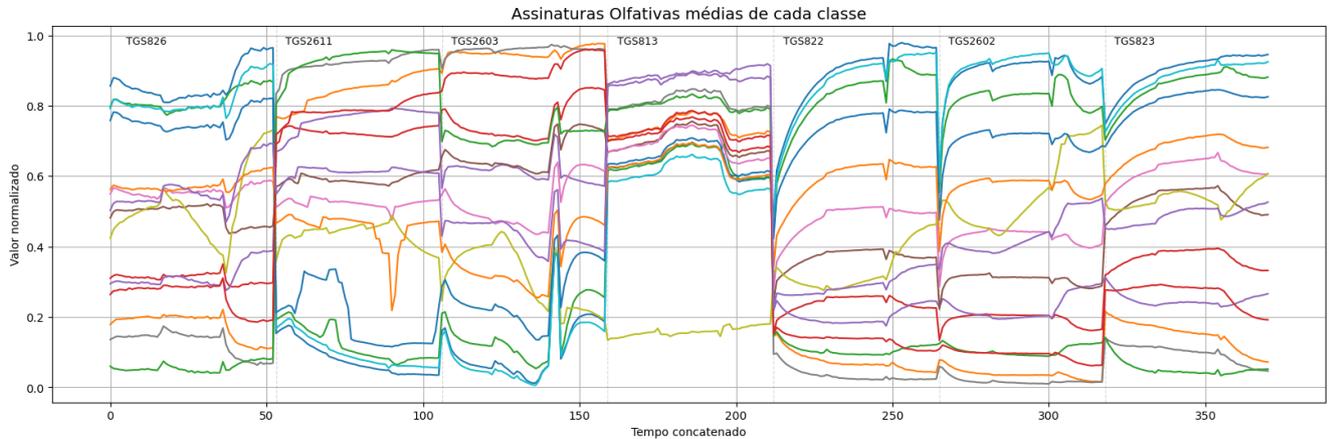


Figura 3.6: Assinaturas olfativas das classes trabalhadas

3.2.6 Transformação das assinaturas para dados 3D

Essa fase é crucial para os modelos temporais que empregaremos na classificação das assinaturas, uma vez que o dataset 3D é exatamente o tipo de entrada necessário para que o modelo aprenda. Isso pode ser descrito como um vetor tridimensional (n_cases , $n_channels$, $n_timepoints$). N_cases , como o nome sugere, refere-se à quantidade de casos, que é o número de linhas no nosso conjunto de dados. $n_channels$ representa o número de canais de captura que foram empregados para obter esse conjunto de dados, enquanto $n_timepoints$ indica a duração temporal de cada ciclo, ou seja, o número de pontos registrados ao longo do tempo em cada amostra.

3.2.7 Escolha dos sensores

É necessário entender quais sensores são importantes e quais não são. Os métodos escolhidos para essa tarefa foi o $r2_score$, o SHAP (SHapley Additive exPlanations) e também a utilização do modelo de RandomForest com seu cálculo de $feature_importance$. Ambos o SHAP e $feature_importance$ da RandomForest foram treinadas utilizando a base de treinamento.

- **R2_score:** O método $r2_score$ recebe duas curvas e calcula a distância entre elas. Quanto maior o valor da distância, mais distintas são as curvas e, em tese, melhor é a separação entre as classes. Com base nisso, nossa abordagem consistiu em calcular a média das curvas de cada sensor dentro de cada classe. Em seguida, comparamos essas médias entre as diferentes classes, sensor por sensor, utilizando o $r2_score$ para identificar quais sensores apresentam maior capacidade de distinção entre as classes.
- **SHAP:** Avalia a contribuição de cada feature levando em conta todas as combinações possíveis (coalizões) de features e seu efeito na saída do modelo. Posteriormente, ele determina a média dessas contribuições marginais em todas as possíveis ordenações das

características. Este processo mede de forma eficaz o efeito individual de cada feature na previsão.

- **Feature_importance RandomForest:** As importâncias das features são fornecidas pelo atributo ajustado `feature_importances_` e são calculadas como a média e o desvio padrão do acúmulo da redução de impureza dentro de cada árvore[21].

Resultados

4.1 Modelagem

A modelagem foi elaborada da seguinte maneira, foi realizada uma busca de hiperparâmetros por meio da técnica GridSearch, validação cruzada com busca em grade. Para cada modelo avaliado, foi previamente definido um conjunto específico de hiperparâmetros, formando uma grade de busca (`param_grid`) necessária para utilização do GridSearch. Essa grade é explorada exaustivamente, avaliando todas as combinações possíveis entre os valores especificados de hiperparâmetros na grade de busca.

A avaliação de cada combinação foi realizada utilizando StratifiedKFold com 3 dobraduras, que visa garantir que a proporção entre as classes seja mantida. Visando reduzir o risco de overfitting e oferecer de maneira mais generalizada dados ao modelo. Vale ressaltar que durante a etapa realizada para busca de hiperparâmetros a base de dados utilizada foi a base de dados de treinamento.

A métrica utilizada nesta etapa para escolha dos melhores hiperparâmetros foi a média da acurácia das cinco dobras. Ao final da busca, o GridSearch retorna o modelo com os melhores hiperparâmetros encontrados, ou seja, aquele que obteve o maior desempenho médio com a base de dados de treinamento. Esse modelo final é utilizado para fazer a predição dos dados de testes com os hiperparâmetros já otimizados e agora é calculado as métricas finais de desempenho (acurácia, precisão, recall e F1-score)

A seleção dos sensores seguiu a ordem de relevância previamente definida: os modelos com apenas um sensor utilizaram o mais relevante da lista, os com dois sensores utilizaram os dois mais relevantes, e assim sucessivamente. Toda essa etapa acima foi realizada separadamente para cada combinação de subconjunto de sensores e modelo de classificação, possibilitando uma comparação justa e abrangente entre diferentes cenários de entrada.

4.2 Experimentos

Na sequência já citada, serão analisados três experimentos: Conjunto de Treinamento: CA - Teste: CI; Conjunto de Treinamento: CA, CI, CNA - Teste: CA, CI, CNA; Conjunto de Treinamento: 2/3 CI - Teste: 1/3 CI.

Ao examinar os gráficos exibidos em cada subseção, poderemos determinar quais sensores são mais relevantes para distinguir as classes. Para isso, usaremos principalmente a ordem dos valores do `r2_score`, organizando os sensores de acordo com sua importância.

Os resultados de cada experimento serão analisados em maior profundidade em cada sub-

seção, com tabelas apresentando os 20 melhores modelos clássicos e os 15 melhores modelos temporais.

4.2.1 Conjunto Treinamento: CA - Teste CI

Nesse conjunto de dados é possível a remoção de 2 sensores inexpressivos, sendo eles "TGS2611" e "TGS2603". Utilizaremos partir de agora, uma ordem de importância entre os sensores a escolhida será. Essa ordem será: "TGS823", "TGS826", "TGS813", "TGS2602", "TGS822".

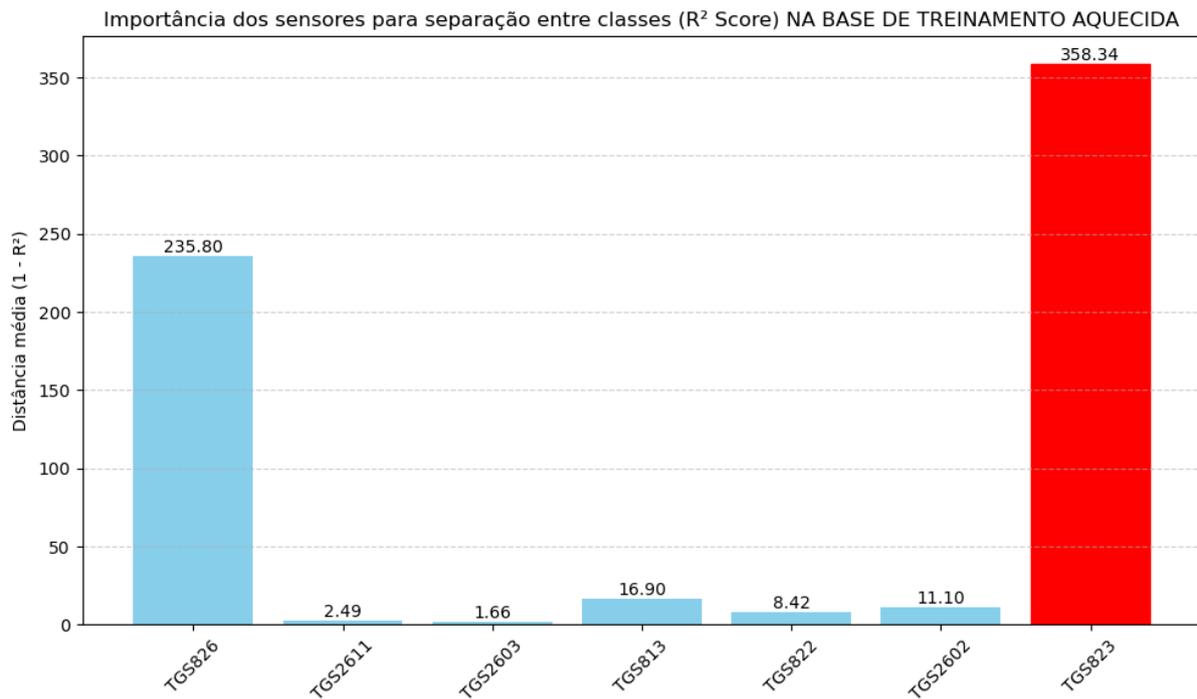


Figura 4.1: Importância dos sensores quantificada via R^2 _score.

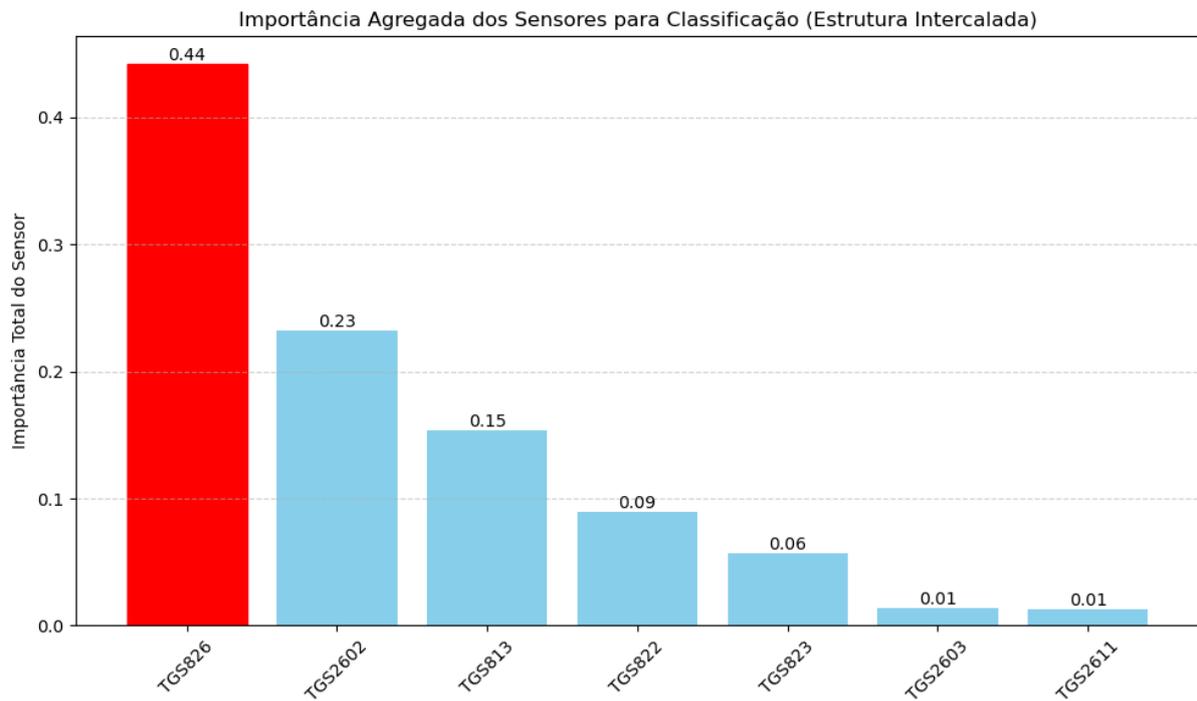


Figura 4.2: Importância dos sensores quantificada via `feature_importance` da RandomForest treinada na base de treinamento.

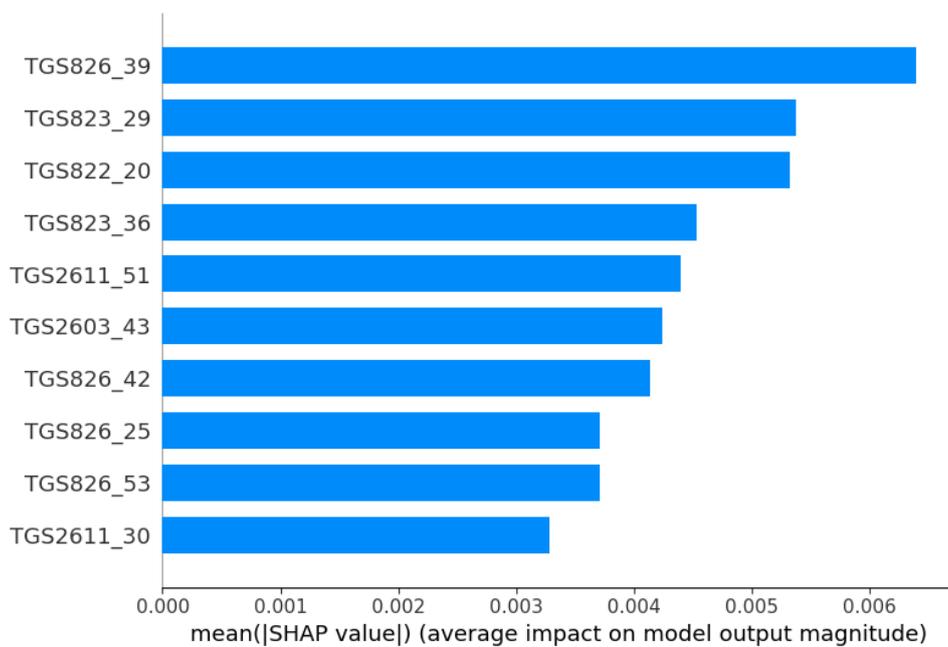


Figura 4.3: Importância dos sensores quantificada via SHAP da RandomForest treinada, na base de treinamento, para classe `af_leve`. Labels no eixo Y estão no formato `Sensor_tempo`, onde cada linha representa um sensor em um instante de tempo específico.

4.2.2 Análise dos resultados

É evidente que os conjuntos de dados são totalmente diferentes, principalmente por causa da degradação que o conjunto CA apresenta. Essa degradação acontece porque as amostras do conjunto CA não foram coletadas e analisadas no mesmo dia, o que faz com que as características dos dados mudem ao longo do tempo. Essa disparidade temporal afeta a qualidade e a uniformidade das amostras.

Além disso, essa hipótese é reforçada pela discrepância entre a alta acurácia no conjunto de treinamento e o baixo desempenho nos testes. Alguns modelos apresentam acurácias de 100% aos dados de treinamento, indicando que conseguiu representar adequadamente as características presentes nesse conjunto. No entanto, ao serem avaliados na base de teste, a acurácia é reduzida, mostrando a disparidade entre os conjuntos. Comprometendo sua capacidade de generalização.

4.2.2.1 Resultado dos 20 melhores modelos na base de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.

Modelo	Nº Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
DecisionTree	2	0.99	0.99	0.99	0.99	{"max_depth": 10}
DecisionTree	3	1.00	1.00	1.00	1.00	{"max_depth": 10}
DecisionTree	5	1.00	1.00	1.00	1.00	{"max_depth": 10}
KNN	1	0.46	0.48	0.46	0.45	{"n_neighbors": 7}
KNN	3	0.70	0.74	0.70	0.69	{"n_neighbors": 3}
LogisticRegression	1	0.54	0.57	0.54	0.50	{"C": 10, "solver": "liblinear"}
LogisticRegression	2	0.84	0.86	0.84	0.83	{"C": 10, "solver": "liblinear"}
LogisticRegression	3	0.88	0.89	0.88	0.87	{"C": 10, "solver": "liblinear"}
LogisticRegression	5	0.92	0.93	0.92	0.92	{"C": 10, "solver": "liblinear"}
MLP	1	0.84	0.86	0.84	0.85	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	2	0.99	0.99	0.99	0.99	{"alpha": 0.01, "hidden_layer_sizes": [50, 50]}
MLP	5	1.00	1.00	1.00	1.00	{"alpha": 0.001, "hidden_layer_sizes": [100]}
NaiveBayes	1	0.34	0.26	0.34	0.29	{"var_smoothing": 1e-09}
NaiveBayes	3	0.79	0.83	0.79	0.77	{"var_smoothing": 1e-09}
RandomForest	1	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 100}
RandomForest	2	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 100}
RandomForest	4	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 200}
RandomForest	5	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 200}
SVM	3	0.83	0.83	0.83	0.82	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	5	0.92	0.93	0.92	0.92	{"C": 10, "gamma": "scale", "kernel": "rbf"}

Tabela 4.1: Tabela com os 20 melhores modelos, ordenada por nome e número de sensores

4.2.2.2 Resultado dos 20 melhores modelos na base de teste, na base de teste. Tabela ordenada por modelo e N sensores

Modelo	Nº Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
DecisionTree	2	0.02	0.01	0.02	0.02	{"max_depth": 10}
DecisionTree	3	0.10	0.04	0.10	0.06	{"max_depth": 10}
DecisionTree	5	0.07	0.03	0.07	0.04	{"max_depth": 10}
KNN	1	0.11	0.03	0.11	0.05	{"n_neighbors": 7}
KNN	3	0.06	0.01	0.06	0.01	{"n_neighbors": 3}
LogisticRegression	1	0.07	0.01	0.07	0.02	{"C": 10, "solver": "liblinear"}
LogisticRegression	2	0.06	0.03	0.06	0.04	{"C": 10, "solver": "liblinear"}
LogisticRegression	3	0.07	0.03	0.07	0.03	{"C": 10, "solver": "liblinear"}
LogisticRegression	5	0.02	0.02	0.02	0.02	{"C": 10, "solver": "liblinear"}
MLP	1	0.11	0.02	0.11	0.03	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	2	0.01	0.02	0.01	0.02	{"alpha": 0.01, "hidden_layer_sizes": [50, 50]}
MLP	5	0.04	0.02	0.04	0.03	{"alpha": 0.001, "hidden_layer_sizes": [100]}
NaiveBayes	1	0.05	0.02	0.05	0.03	{"var_smoothing": 1e-09}
NaiveBayes	3	0.01	0.04	0.01	0.02	{"var_smoothing": 1e-09}
RandomForest	1	0.07	0.03	0.07	0.05	{"max_depth": 10, "n_estimators": 100}
RandomForest	2	0.01	0.01	0.01	0.01	{"max_depth": 10, "n_estimators": 100}
RandomForest	4	0.02	0.03	0.02	0.03	{"max_depth": 10, "n_estimators": 200}
RandomForest	5	0.02	0.02	0.02	0.02	{"max_depth": 10, "n_estimators": 200}
SVM	3	0.01	0.07	0.01	0.02	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	5	0.06	0.02	0.06	0.03	{"C": 10, "gamma": "scale", "kernel": "rbf"}

Tabela 4.2: Tabela com os 20 melhores modelos, ordenada por nome e número de sensores

4.2.3 Conjunto Treinamento: CA, CI, CNA - Teste: CA, CI, CNA

Nesse conjunto de dados é possível a remoção de 2 sensores inexpressivos, sendo eles "TGS822" e "TGS2603". Utilizaremos partir de agora, uma ordem de importância entre os sensores a escolhida será. Essa ordem será: "TGS813", "TGS823", "TGS2611", "TGS2602", "TGS826".

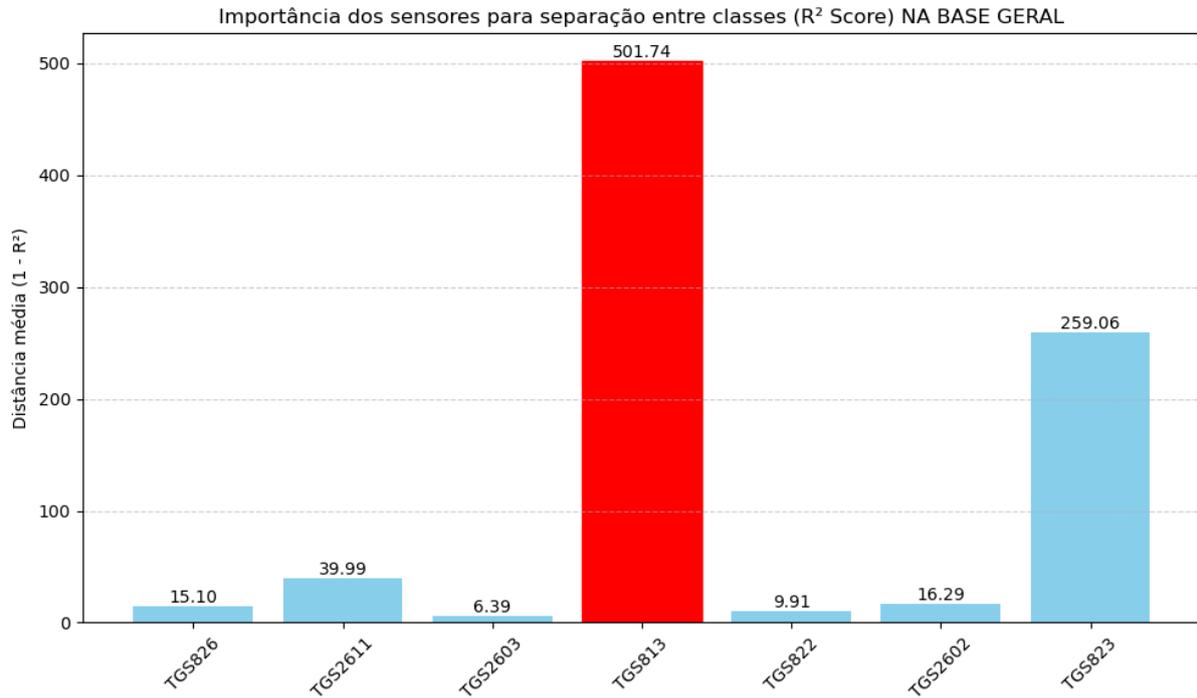


Figura 4.4: Importância dos sensores quantificada via $R2_score$.

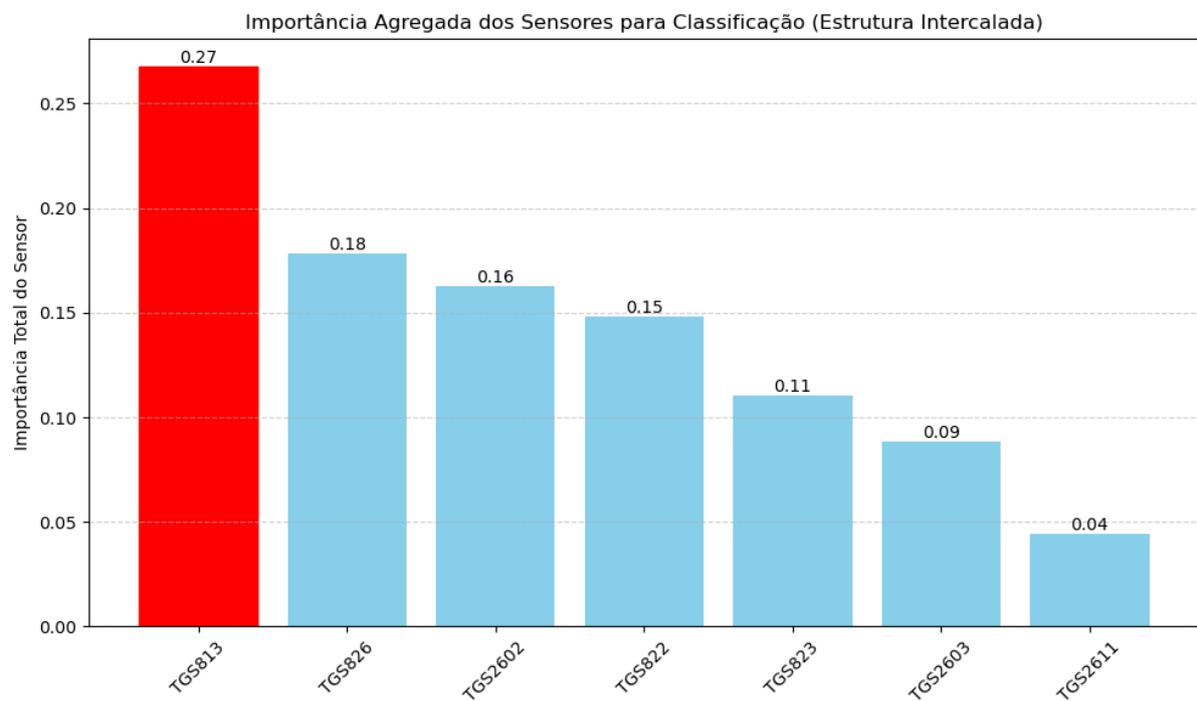


Figura 4.5: Importância dos sensores quantificada via `feature_importance` da RandomForest treinada na base de treinamento.

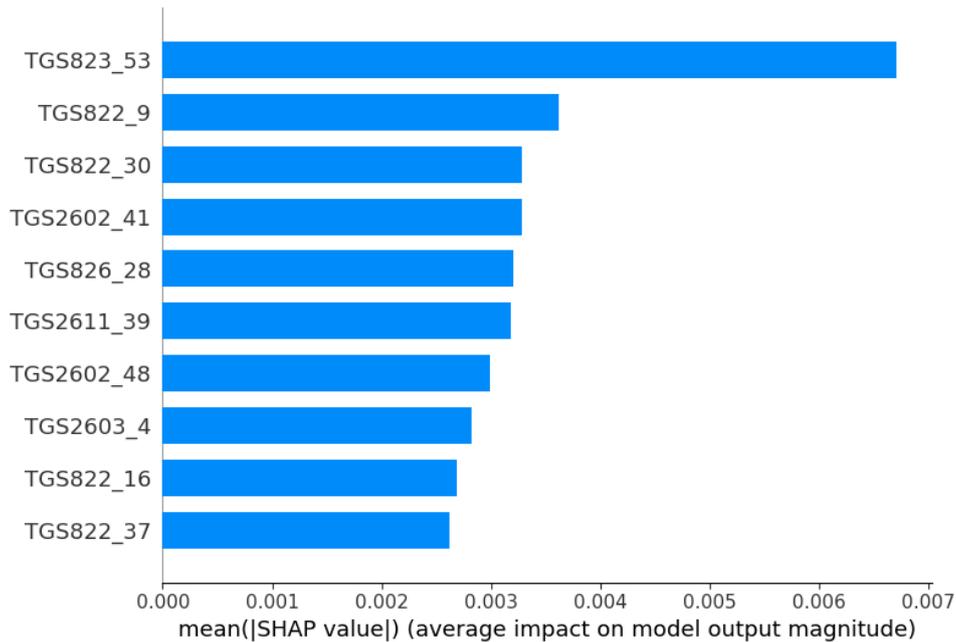


Figura 4.6: Importância dos sensores quantificada via SHAP da RandomForest treinada, na base de treinamento, para classe *af_leve*. Labels no eixo Y estão no formato *Sensor_tempo*, onde cada linha representa um sensor em um instante de tempo específico.

4.2.4 Análise dos resultados

Os resultados mostraram um desempenho superior quando todas as bases foram utilizadas em conjunto para treinar e testar os modelos. Porém, mesmo com essa unificação dos dados, os resultados obtidos ainda não conseguem atingir uma acurácia superior a 80%.

Essa restrição provavelmente se deve à degradação das amostras dos conjuntos CA e CI, que foram coletadas em momentos distintos e, portanto, passaram por mudanças físico químicas ao longo do tempo. A inconsistência dos dados temporais torna difícil para os modelos aprenderem de forma robusta e prejudica sua capacidade de generalização para novos dados.

Apesar de todos esses desafios, os modelos temporais apresentaram uma ligeira superioridade na acurácia em relação aos modelos clássicos de classificação, o que demonstra uma evolução no desempenho. Isso indica que os métodos que tiram proveito das dinâmicas temporais das assinaturas olfativas estão capturando informações relevantes que os métodos convencionais, que se baseiam em características mais fixas, estão perdendo.

4.2.4.1 Resultado dos 20 melhores modelos na base de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.

Modelo	Nº Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
DecisionTree	1	1.00	1.00	1.00	1.00	{"max_depth": 20}
DecisionTree	2	1.00	1.00	1.00	1.00	{"max_depth": 20}
DecisionTree	3	1.00	1.00	1.00	1.00	{"max_depth": 20}
DecisionTree	4	0.94	0.95	0.94	0.94	{"max_depth": 10}
DecisionTree	5	1.00	1.00	1.00	1.00	{"max_depth": 20}
KNN	1	0.53	0.52	0.53	0.51	{"n_neighbors": 3}
KNN	2	0.74	0.75	0.74	0.73	{"n_neighbors": 3}
KNN	3	0.73	0.76	0.73	0.73	{"n_neighbors": 3}
KNN	4	0.81	0.83	0.81	0.81	{"n_neighbors": 3}
KNN	5	0.80	0.81	0.80	0.80	{"n_neighbors": 3}
RandomForest	1	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 100}
RandomForest	2	1.00	1.00	1.00	1.00	{"max_depth": 20, "n_estimators": 200}
RandomForest	3	1.00	1.00	1.00	1.00	{"max_depth": 20, "n_estimators": 200}
RandomForest	4	1.00	1.00	1.00	1.00	{"max_depth": 20, "n_estimators": 100}
RandomForest	5	1.00	1.00	1.00	1.00	{"max_depth": 20, "n_estimators": 100}
SVM	1	0.58	0.64	0.58	0.57	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	2	0.72	0.76	0.72	0.72	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	3	0.78	0.83	0.78	0.78	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	4	0.82	0.85	0.82	0.82	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	5	0.83	0.85	0.83	0.82	{"C": 10, "gamma": "scale", "kernel": "rbf"}

Tabela 4.3: Tabela com os 20 melhores modelos, ordenada por nome e número de sensores

4.2.4.2 Resultado dos 20 melhores modelos na base de teste, na base de testes. Tabela ordenada por modelo e N sensores.

Modelo	Nº Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
DecisionTree	1	0.42	0.40	0.42	0.39	{"max_depth": 20}
DecisionTree	2	0.51	0.54	0.51	0.50	{"max_depth": 20}
DecisionTree	3	0.49	0.50	0.49	0.46	{"max_depth": 20}
DecisionTree	4	0.49	0.58	0.49	0.50	{"max_depth": 10}
DecisionTree	5	0.51	0.50	0.51	0.47	{"max_depth": 20}
KNN	1	0.47	0.51	0.47	0.47	{"n_neighbors": 5}
KNN	2	0.64	0.64	0.64	0.62	{"n_neighbors": 3}
KNN	3	0.55	0.55	0.55	0.53	{"n_neighbors": 3}
KNN	4	0.66	0.74	0.66	0.66	{"n_neighbors": 3}
KNN	5	0.68	0.74	0.68	0.69	{"n_neighbors": 3}
RandomForest	1	0.45	0.47	0.45	0.44	{"max_depth": 10, "n_estimators": 100}
RandomForest	2	0.58	0.60	0.58	0.57	{"max_depth": 20, "n_estimators": 200}
RandomForest	3	0.58	0.65	0.58	0.58	{"max_depth": 20, "n_estimators": 200}
RandomForest	4	0.66	0.71	0.66	0.66	{"max_depth": 20, "n_estimators": 100}
RandomForest	5	0.68	0.71	0.68	0.68	{"max_depth": 20, "n_estimators": 100}
SVM	1	0.42	0.50	0.42	0.41	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	2	0.49	0.57	0.49	0.47	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	3	0.43	0.49	0.43	0.43	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	4	0.66	0.72	0.66	0.65	{"C": 10, "gamma": "scale", "kernel": "rbf"}
SVM	5	0.64	0.72	0.64	0.62	{"C": 10, "gamma": "scale", "kernel": "rbf"}

Tabela 4.4: Tabela com os 20 melhores modelos, ordenada por nome e número de sensores

4.2.4.3 Resultado dos 15 melhores modelos na fase de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.

Modelo	Nº Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
KNNTTimeSeries	1	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	2	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	3	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	4	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	5	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
Rocket	1	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	2	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	3	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	4	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	5	0.97	0.97	0.97	0.97	{"n_kernels": 1000}
Signature	1	1.00	1.00	1.00	1.00	{"depth": 3, "window_name": "global"}
Signature	2	1.00	1.00	1.00	1.00	{"depth": 3, "window_name": "global"}
Signature	3	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}
Signature	4	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}
Signature	5	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}

Tabela 4.5: Tabela com os 15 melhores modelos, ordenada por nome e número de sensores

4.2.4.4 Resultado dos 15 melhores modelos na fase de teste, na base de teste. Tabela ordenada por modelo temporal e N sensores.

Modelo	Nº Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
KNNTTimeSeries	1	0.53	0.55	0.53	0.53	{"n_neighbors": 1}
KNNTTimeSeries	2	0.64	0.68	0.64	0.63	{"n_neighbors": 1}
KNNTTimeSeries	3	0.60	0.69	0.60	0.59	{"n_neighbors": 1}
KNNTTimeSeries	4	0.66	0.67	0.66	0.66	{"n_neighbors": 1}
KNNTTimeSeries	5	0.70	0.75	0.70	0.70	{"n_neighbors": 1}
Rocket	1	0.40	0.43	0.40	0.38	{"n_kernels": 2000}
Rocket	2	0.38	0.41	0.38	0.37	{"n_kernels": 2000}
Rocket	3	0.32	0.35	0.32	0.30	{"n_kernels": 2000}
Rocket	4	0.36	0.35	0.36	0.34	{"n_kernels": 2000}
Rocket	5	0.38	0.39	0.38	0.37	{"n_kernels": 1000}
Signature	1	0.43	0.45	0.43	0.42	{"depth": 3, "window_name": "global"}
Signature	2	0.64	0.65	0.64	0.63	{"depth": 3, "window_name": "global"}
Signature	3	0.68	0.76	0.68	0.67	{"depth": 2, "window_name": "global"}
Signature	4	0.68	0.74	0.68	0.67	{"depth": 2, "window_name": "global"}
Signature	5	0.72	0.79	0.72	0.72	{"depth": 2, "window_name": "global"}

Tabela 4.6: Tabela com os 15 melhores modelos, ordenada por nome e número de sensores

4.2.5 Conjunto Treinamento: CI - Teste CI

4.2.5.1 Resultados métodos de escolha dos sensores

Nesse conjunto de dados não foi possível a remoção clara de um dos sensores. Utilizaremos partir de agora, uma ordem de importância entre os sensores a escolhida será. Essa ordem será: "TGS2602", "TGS2603", "TGS2611", "TGS813", "TGS826", "TGS823", "TGS822".

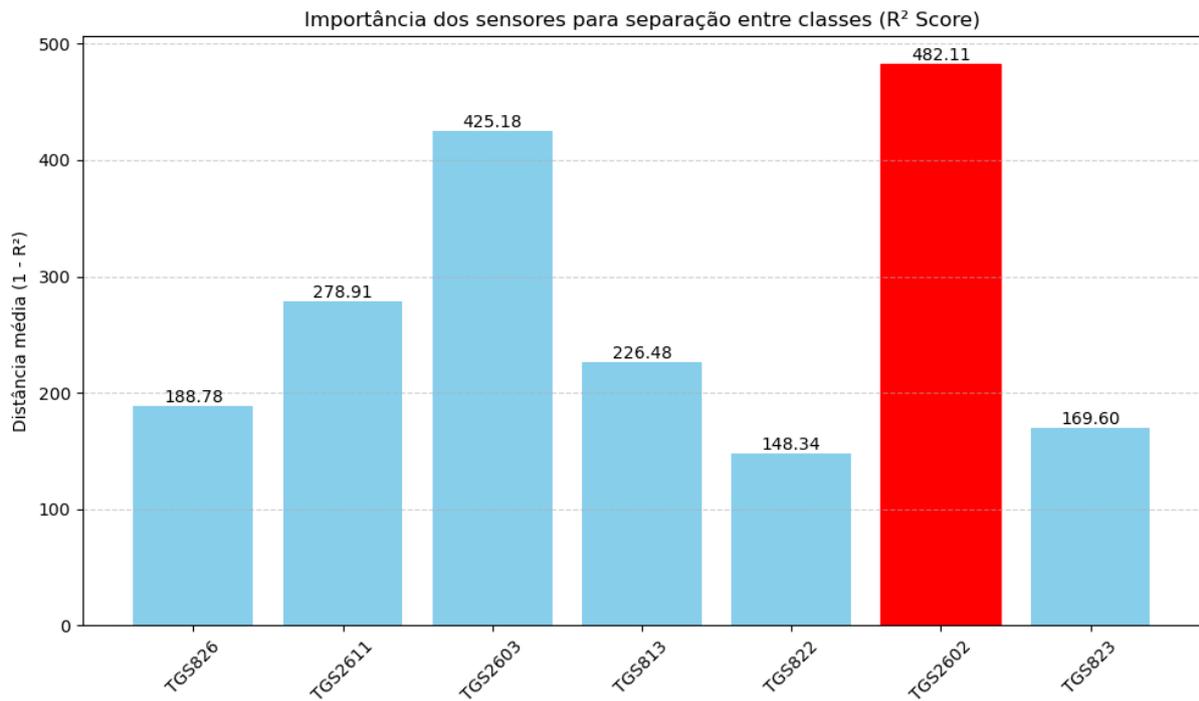


Figura 4.7: Importância dos sensores quantificada via R^2 _score.

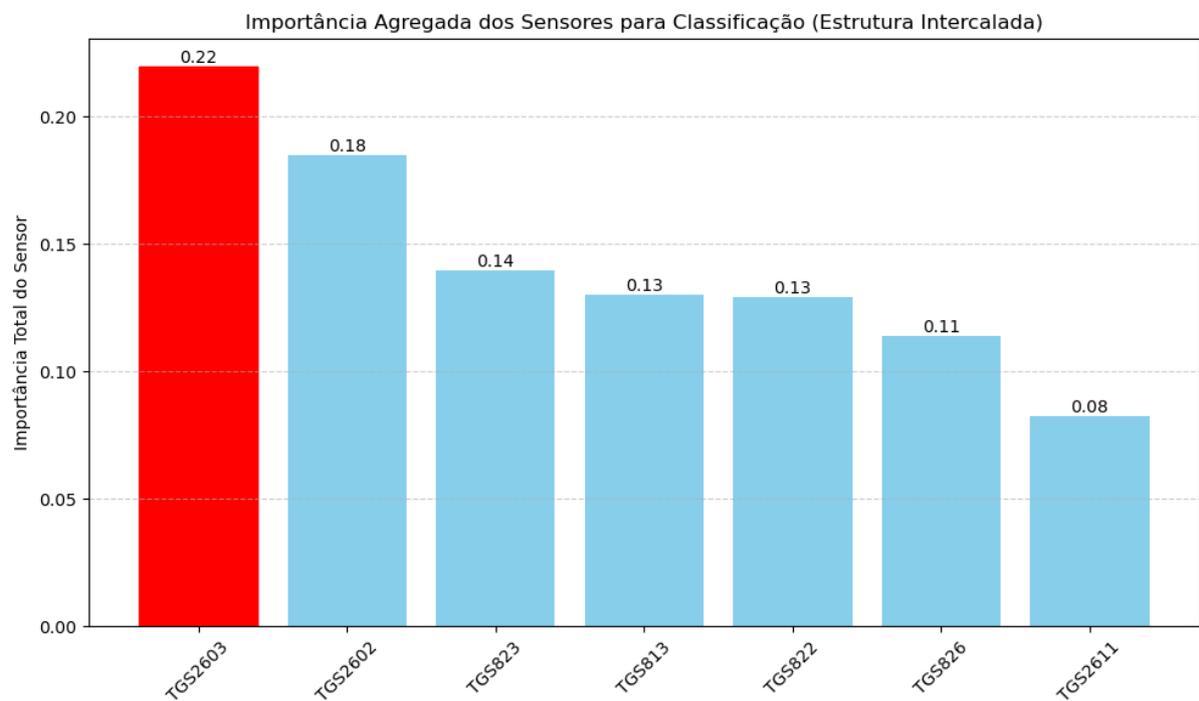


Figura 4.8: Importância dos sensores quantificada via `feature_importance` da RandomForest treinada na base de treinamento.

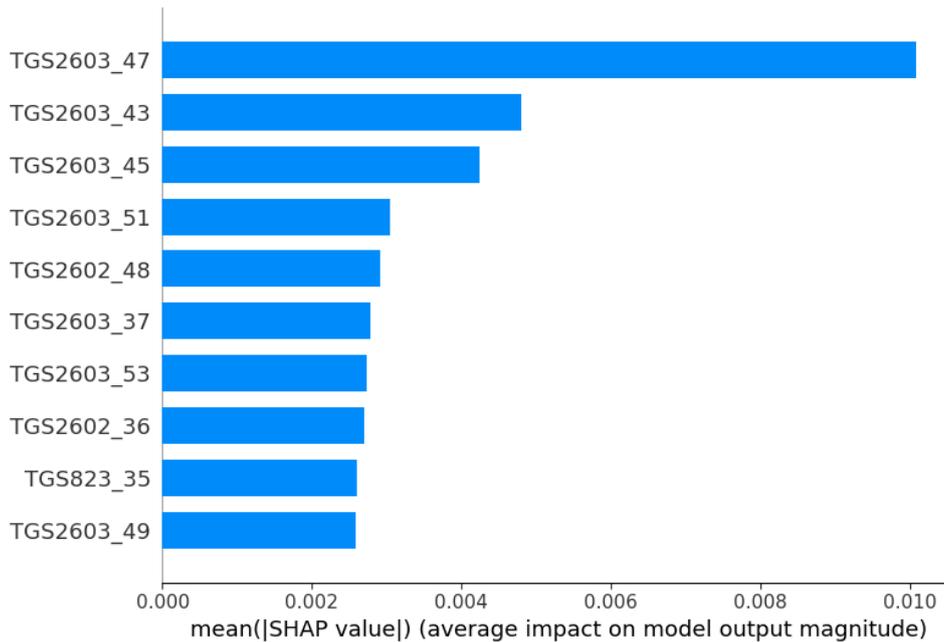


Figura 4.9: Importância dos sensores quantificada via SHAP da RandomForest treinada, na base de treinamento, para classe *af_leve*. Labels no eixo Y estão no formato *Sensor_tempo*, onde cada linha representa um sensor em um instante de tempo específico.

4.2.6 Análise dos resultados

Esse conjunto é o conjunto mais importante para verificar se é possível a classificação utilizando modelos de classificação para a análise de amostras, pois, estão alinhados para melhor caracterização das amostras visto que essa é uma separação pura das amostras que não sofreram degradação e foram coletadas e analisadas no mesmo dia. Entretanto, a quantidade limitada de dados para treinamento pode ter levado a um overfitting na fase de ajuste dos modelos, mesmo com a técnica de Stratified K-Fold sendo utilizada. Este fenômeno levou a uma classificação perfeita (100% de acurácia) nas amostras do conjunto de treinamento na maioria dos modelos testados, evidenciada na tabela 4.7. Esse desempenho perfeito indica que os modelos decoraram os dados de treinamento, o que pode comprometer a habilidade de generalizar para novos dados.

Portanto, a análise dos resultados foi restringida ao conjunto de teste, que é composto por dados que os modelos nunca viram antes durante o treinamento. A avaliação desse conjunto é mais confiável para determinar a verdadeira capacidade preditiva dos algoritmos.

Quando se analisam os resultados do conjunto de teste, é evidente que os modelos clássicos como: MLP e Random Forest se sobressaem, alcançando um desempenho consideravelmente superior ao dos outros modelos, evidenciada na tabela 4.8. Isso sugere que essas arquiteturas conseguiram captar melhor os padrões significativos nos dados.

O MLP foi o único modelo utilizado que é classificado como rede neural, apresentando um maior custo computacional em relação aos outros modelos utilizados. Porém, fica evidenciado o ganho obtido com essa maior complexidade, demonstrando que o MLP foi capaz de capturar

padrões mais complexos nos dados. Além disso, a flexibilidade das redes neurais em modelar relações não lineares pode justificar o seu desempenho.

O desempenho do Random Forest deve-se à sua capacidade de lidar com dados de alta dimensionalidade e de capturar interações complexas entre características. Ademais, sua interpretabilidade relativa, por meio de métricas como a importância das features, contribui para a compreensão dos fatores que influenciam a classificação, o que representa uma vantagem prática para aplicações que demandam explicabilidade.

Entretanto, o modelo Signature se destacou, alcançando uma impressionante acurácia de 100% no conjunto de teste. Isso se deve ao fato de que a estratégia de signatures é especialmente poderosa para identificar padrões temporais complexos e dependências de alta ordem em séries temporais, o que explica esse desempenho. Este método, ao mapear as trajetórias dos sinais em um espaço de características denso e resistente a algumas transformações, consegue manter informações cruciais sobre a forma e a dinâmica das curvas, que poderiam ser comprometidas em métodos convencionais de extração de atributos.

Outro aspecto interessante observado foi que vários dos dez melhores modelos utilizaram apenas um subconjunto dos sensores disponíveis. Esse resultado indica que não é necessário contar com todos os sensores para se obter uma classificação eficaz. Isso implica que uma seleção eficiente de atributos é viável, o que não apenas diminui o custo computacional do treinamento e da inferência, mas também torna o processo de coleta de dados mais simples em situações do mundo real.

Esses resultados destacam a relevância da seleção cuidadosa do modelo e da análise rigorosa das variáveis empregadas, levando a soluções mais eficientes e viáveis em contextos semelhantes.

4.2.6.1 Resultado dos 20 melhores modelos na base de teste, na base de treinamento. Tabela ordenada por modelo e N sensores.

Modelo	N sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
DecisionTree	1	1.00	1.00	1.00	1.00	{"max_depth": 10}
DecisionTree	6	1.00	1.00	1.00	1.00	{"max_depth": 10}
KNN	1	0.88	0.90	0.88	0.88	{"n_neighbors": 3}
KNN	2	0.83	0.86	0.83	0.83	{"n_neighbors": 3}
KNN	3	0.86	0.88	0.86	0.87	{"n_neighbors": 3}
KNN	4	0.91	0.93	0.91	0.91	{"n_neighbors": 3}
KNN	5	0.88	0.91	0.88	0.88	{"n_neighbors": 3}
KNN	7	0.83	0.86	0.83	0.83	{"n_neighbors": 3}
MLP	2	1.00	1.00	1.00	1.00	{"alpha": 0.001, "hidden_layer_sizes": [100]}
MLP	3	1.00	1.00	1.00	1.00	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	4	1.00	1.00	1.00	1.00	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	5	1.00	1.00	1.00	1.00	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	6	1.00	1.00	1.00	1.00	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	7	1.00	1.00	1.00	1.00	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
RandomForest	1	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 100}
RandomForest	2	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 150}
RandomForest	4	1.00	1.00	1.00	1.00	{"max_depth": 20, "n_estimators": 200}
RandomForest	5	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 150}
RandomForest	6	1.00	1.00	1.00	1.00	{"max_depth": 10, "n_estimators": 100}
RandomForest	7	1.00	1.00	1.00	1.00	{"max_depth": 20, "n_estimators": 150}

Tabela 4.7: Tabela com os 20 melhores modelos da base de treino, ordenada por nome e número de sensores

4.2.6.2 Resultado dos 20 melhores modelos na base de teste, na base de testes. Tabela ordenada por modelo e N sensores.

Modelo	N sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
DecisionTree	1	0.80	0.82	0.80	0.78	{"max_depth": 10}
DecisionTree	6	0.83	0.82	0.83	0.80	{"max_depth": 10}
KNN	1	0.87	0.91	0.87	0.86	{"n_neighbors": 3}
KNN	2	0.87	0.86	0.87	0.84	{"n_neighbors": 3}
KNN	3	0.87	0.91	0.87	0.86	{"n_neighbors": 3}
KNN	4	0.87	0.92	0.87	0.86	{"n_neighbors": 3}
KNN	5	0.83	0.89	0.83	0.83	{"n_neighbors": 3}
KNN	7	0.83	0.89	0.83	0.83	{"n_neighbors": 3}
MLP	2	0.90	0.93	0.90	0.89	{"alpha": 0.001, "hidden_layer_sizes": [100]}
MLP	3	0.83	0.86	0.83	0.83	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	4	0.93	0.96	0.93	0.93	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	5	0.87	0.92	0.87	0.86	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	6	0.87	0.92	0.87	0.86	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
MLP	7	0.90	0.93	0.90	0.89	{"alpha": 0.001, "hidden_layer_sizes": [50, 50]}
RandomForest	1	0.87	0.92	0.87	0.86	{"max_depth": 10, "n_estimators": 100}
RandomForest	2	0.90	0.93	0.90	0.89	{"max_depth": 10, "n_estimators": 150}
RandomForest	4	0.90	0.93	0.90	0.89	{"max_depth": 20, "n_estimators": 200}
RandomForest	5	0.83	0.86	0.83	0.82	{"max_depth": 10, "n_estimators": 150}
RandomForest	6	0.87	0.86	0.87	0.84	{"max_depth": 10, "n_estimators": 100}
RandomForest	7	0.87	0.91	0.87	0.86	{"max_depth": 20, "n_estimators": 150}

Tabela 4.8: Tabela com os 20 melhores modelos, ordenada por nome e número de sensores

4.2.6.3 Resultado dos 15 melhores modelos na fase de teste, na base de treinamento. Tabela ordenada por modelo temporal e N sensores.

Modelo	Nº Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
KNNTTimeSeries	1	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	2	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	3	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	4	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	5	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	6	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
KNNTTimeSeries	7	1.00	1.00	1.00	1.00	{"n_neighbors": 1}
Rocket	1	0.97	0.98	0.97	0.97	{"n_kernels": 2000}
Rocket	2	0.97	0.98	0.97	0.97	{"n_kernels": 500}
Rocket	3	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	4	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	5	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	6	1.00	1.00	1.00	1.00	{"n_kernels": 2000}
Rocket	7	1.00	1.00	1.00	1.00	{"n_kernels": 500}
Signature	1	1.00	1.00	1.00	1.00	{"depth": 4, "window_name": "global"}
Signature	2	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}
Signature	3	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}
Signature	4	1.00	1.00	1.00	1.00	{"depth": 3, "window_name": "global"}
Signature	5	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}
Signature	6	1.00	1.00	1.00	1.00	{"depth": 3, "window_name": "global"}
Signature	7	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}

Tabela 4.9: Tabela com os 15 melhores modelos, ordenada por nome e número de sensores

4.2.6.4 Resultado dos 15 melhores modelos na fase de teste, na base de teste. Tabela ordenada por modelo temporal e N sensores.

Modelo	N Sensores	Accuracy	Precision	Recall	F1-score	Parâmetros
KNNTTimeSeries	1	0.86	0.90	0.86	0.86	{"n_neighbors": 1}
KNNTTimeSeries	2	0.86	0.87	0.86	0.86	{"n_neighbors": 1}
KNNTTimeSeries	3	0.90	0.92	0.90	0.89	{"n_neighbors": 1}
KNNTTimeSeries	4	0.93	0.95	0.93	0.92	{"n_neighbors": 1}
KNNTTimeSeries	5	0.93	0.94	0.93	0.93	{"n_neighbors": 1}
KNNTTimeSeries	6	0.90	0.92	0.90	0.89	{"n_neighbors": 1}
KNNTTimeSeries	7	0.90	0.94	0.90	0.89	{"n_neighbors": 1}
Rocket	2	0.56	0.50	0.56	0.51	{"n_kernels": 1000}
Rocket	3	0.76	0.77	0.76	0.73	{"n_kernels": 2000}
Rocket	4	0.60	0.60	0.60	0.56	{"n_kernels": 2000}
Rocket	5	0.66	0.61	0.66	0.61	{"n_kernels": 500}
Rocket	6	0.66	0.57	0.66	0.59	{"n_kernels": 500}
Rocket	7	0.63	0.58	0.63	0.57	{"n_kernels": 500}
Signature	1	0.86	0.90	0.86	0.86	{"depth": 2, "window_name": "global"}
Signature	2	0.93	0.93	0.93	0.93	{"depth": 2, "window_name": "global"}
Signature	3	0.86	0.88	0.86	0.86	{"depth": 2, "window_name": "global"}
Signature	4	0.93	0.93	0.93	0.93	{"depth": 2, "window_name": "global"}
Signature	5	0.93	0.95	0.93	0.92	{"depth": 2, "window_name": "global"}
Signature	6	0.90	0.91	0.90	0.89	{"depth": 2, "window_name": "global"}
Signature	7	1.00	1.00	1.00	1.00	{"depth": 2, "window_name": "global"}

Tabela 4.10: Tabela com os 15 melhores modelos, ordenada por nome e número de sensores

Conclusão

O presente trabalho propôs e demonstrou a possibilidade de utilizar um sistema para o monitoramento inteligente da qualidade do óleo de soja, que integra um nariz eletrônico (E-nose) com modelos de aprendizado de máquina. Essa abordagem visa superar as limitações dos métodos convencionais de controle de qualidade, os quais dependem de painéis sensoriais de especialistas, caracterizados por serem onerosos, subjetivos e suscetíveis à fadiga sensorial.

A solução desenvolvida se concentra na capacidade do E-nose de mimetizar o olfato humano, capturando os perfis de Compostos Orgânicos Voláteis (VOCs) presentes nas amostras de óleo. Esses VOCs, que são subprodutos da degradação química do óleo, como oxidação e hidrólise, permitem a formação de "assinaturas olfativas" únicas para cada amostra, possibilitando a categorização e classificação da qualidade. Para isso, foram empregados 10 sensores no dispositivo E-nose, dos quais 7 são dedicados à captura de VOCs (TGS826, TGS2611, TGS2603, TGS813, TGS822, TGS2602 e TGS823), além de sensores para temperatura, pressão e umidade. As amostras de óleo foram organizadas em 14 categorias distintas, considerando tanto suas características sensoriais quanto variações de temperatura, incluindo amostras aquecidas a 50°C e em temperatura ambiente.

A metodologia do projeto abrangeu um robusto pré-processamento dos dados, que incluiu a seleção de colunas relevantes, a uniformização do tamanho dos ciclos de coleta para criar as assinaturas olfativas, a normalização dos dados usando Min-Max para padronizar as magnitudes e a transformação dos dados para um formato 3D, essencial para a aplicação de modelos temporais.

A importância dos sensores foi criteriosamente avaliada através de métricas como r^2_score , SHAP e a funcionalidade `feature_importance` do modelo `RandomForest`, o que permitiu identificar e, em alguns casos, remover sensores menos expressivos para a diferenciação de classes. No estágio de modelagem, utilizou-se uma vasta gama de algoritmos de aprendizado de máquina, incluindo modelos de aprendizado supervisionado clássicos (`Decision Tree`, `RandomForest`, `Support Vector Machine (SVM)`, `k-Nearest Neighbors (k-NN)`, `Naive Bayes (Gaussiano)`, regressão logística, `Multi Layer Perceptron (MLP)`) e modelos de classificação temporais, como `RocketClassifier`, `KNNTIMEseries` e o `SignatureClassifier`, este último se destacando nos resultados finais.

A otimização dos modelos foi realizada por meio de busca de hiperparâmetros com `GridSearch` e validação cruzada `StratifiedKFold`, utilizando a acurácia média como a principal métrica para selecionar os melhores modelos na fase de treinamento. O desempenho final foi avaliado na base de teste utilizando métricas como acurácia, precisão, recall e F1-score.

Nas tentativas de treinamento com dados aquecidos (CA) e teste com dados industriais (CI), observou-se uma baixa capacidade de generalização, atribuída à degradação temporal

das amostras do conjunto CA, o que levou a um overfitting nos dados de treinamento. A unificação de todas as bases de dados (CA, CI, CNA) para treinamento e teste, não mostrou resultado suficiente para aplicação prática devido à persistência de inconsistências temporais nas amostras.

Contudo, nesse cenário, os modelos temporais demonstraram uma ligeira superioridade em capturar as dinâmicas relevantes dos dados. O cenário mais relevante e promissor foi o conjunto de dados "Treinamento: CI - Teste CI", que compreendia amostras consistentes, coletadas e analisadas no mesmo dia.

Nesse contexto, os modelos MLP e RandomForest se destacaram entre os modelos clássicos, evidenciando sua capacidade de aprender padrões complexos e interações entre características. Notavelmente, o SignatureClassifier alcançou uma acurácia de 100% no conjunto de teste, comprovando sua robustez na identificação de padrões temporais complexos em séries de dados.

5.1 Trabalhos Futuros

Com base nos resultados e no escopo do projeto, algumas direções para trabalhos futuros podem ser exploradas:

- **Expansão e Diversificação da Base de Dados:** Aumentar a quantidade das amostras de óleo de soja, incluindo diferentes fornecedores, condições de armazenamento e tipos de degradação. Isso poderia aprimorar a capacidade de generalização e robustez dos modelos.
- **Otimização dos Sensores:** Desenvolver o E-nose, possivelmente, visando otimizar a configuração atual dos sensores, diminuindo o custo do produto final.
- **Implementação em Tempo Real e Portabilidade:** Desenvolver um protótipo do sistema para validação em um ambiente industrial real, focando na velocidade de análise, portabilidade do dispositivo e integração com os processos de controle de qualidade existentes na indústria.
- **Avaliação de Longo Prazo e Monitoramento Contínuo:** Estudar a capacidade do sistema de monitorar a qualidade do óleo ao longo do tempo, identificando tendências de degradação e fornecendo alertas proativos para a indústria.
- **Expansão do caso de uso:** Verificar a solução proposta em outros alimentos de base para o processo fabril da indústria alimentícia.

Referências Bibliográficas

- [1] R. Amaral, “Gestão de qualidade para indústrias de Óleos vegetais.”
- [2] L. Dudi, N. V. Jillellamudi, C. Chanda, and G. Kanuri, “Assessment of Quality Parameters in Edible Vegetable Oils,” *International Journal of Pharmaceutical Investigation*, vol. 11, pp. 296–299, oct 19 2021.
- [3] R. Ravi, M. Prakash, and K. Bhat, “Sensory odour profiling and physical characteristics of edible oil blends during frying,” *Food Research International*, vol. 38, no. 1, pp. 59–68, 2005.
- [4] D. Karakaya, O. Ulucan, and M. Turkan, “Electronic nose and its applications: A survey,” *International Journal of Automation and Computing*, vol. 17, pp. 179–209, Apr 2020.
- [5] Metrohm USA Inc., “Seven important parameters for analysis of edible oils and fats.” <https://www.news-medical.net/whitepaper/20190821/Seven-Important-Parameters-for-Analysis-of-Edible-Oils-and-Fats.aspx>, May 2020. Accessed: 2025-07-26.
- [6] X. Sun, Y. Wang, H. Li, J. Zhou, J. Han, and C. Wei, “Changes in the volatile profile, fatty acid composition and oxidative stability of flaxseed oil during heating at different temperatures,” *LWT*, vol. 151, p. 112137, 2021.
- [7] T. M. Dymerski, T. M. Chmiel, and W. Wardencki, “Invited review article: An odor-sensing system—powerful technique for foodstuff studies,” *Review of Scientific Instruments*, vol. 82, p. 111101, 11 2011.
- [8] T. Kuchmenko, V. Krylov, A. Smetankina, E. Dorovskaya, D. Menzhulina, V. Bityukova, and R. Umarkhanov, “Portable electronic nose system for fast gynecological-conditions diagnosis in consulting room: A case study,” *Sensors and Actuators B: Chemical*, vol. 358, p. 131538, 2022.
- [9] M. L. Bastos, C. A. Benevides, C. Zanchettin, F. D. Menezes, C. P. Inácio, R. G. de Lima Neto, J. G. A. T. Filho, R. P. Neves, and L. M. Almeida, “Breaking barriers in candida spp. detection with electronic noses and artificial intelligence,” *Scientific Reports*, vol. 14, p. 956, Jan 2024.
- [10] “The smell of lung disease: a review of the current status of electronic nose technology,” 2021. Scientific Figure on ResearchGate.

- Available from: https://www.researchgate.net/figure/Schematic-comparison-of-eNose-technology-and-the-olfactory-system-12_fig1_354672849.
- [11] R. Shumway and D. Stoffer, *Time Series Analysis and Its Applications With R Examples*, vol. 9. 01 2011.
- [12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*. 02 2009.
- [13] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.
- [14] C. Bishop, *Pattern Recognition and Machine Learning*, vol. 16, pp. 140–155. 01 2006.
- [15] S. Russell and P. Norvig, *Inteligência Artificial: Uma abordagem moderna*. Pearson, 4 ed., 2021.
- [16] J. Heaton, “Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618,” *Genetic Programming and Evolvable Machines*, vol. 19, 10 2017.
- [17] J. Morrill, A. Fermanian, P. Kidger, and T. J. Lyons, “A generalised signature method for time series,” *CoRR*, vol. abs/2006.00873, 2020.
- [18] T. aeon developers (BSD-3 License), “Kneighborstimeseriesclassifier.”
- [19] A. Dempster, F. Petitjean, and G. I. Webb, “Rocket: exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining and Knowledge Discovery*, vol. 34, p. 1454–1495, July 2020.
- [20] “Alleviating class-imbalance data of semiconductor equipment anomaly detection study - confusion matrix: Precision, recall, accuracy, and f1 score.” https://www.researchgate.net/figure/Confusion-matrix-Precision-Recall-Accuracy-and-F1-score_fig4_367393140, 2025. Accessed: 2025-07-28.
- [21] T. scikit-learn developers, “Feature importances with a forest of trees.”