Previsão da Qualidade do Leite: Um Estudo de Mineração de Dados e Análise Preditiva

1st Cesar Augusto Gomes dos Santos
BI3003612
IFSP - Campus Birigui
Birigui, São Paulo
cesar.santos@aluno.ifsp.edu.br

2nd Willian Gustavo Rocha Leme
BI3003965
IFSP - Campus Birigui
Birigui, São Paulo
willian.gustavo@aluno.ifsp.edu.br

***Abstract*—The article in question deals with the process of KDD (Knowledge Discovery in Databases) in the database "Milk Quality Prediction", the steps of pre-processing, data transformation, data mining and interpretation of the results were performed. In order to predict an ideal model for a high quality milk.**

***Index Terms*—milk, quality, KDD.**

I. Introdução

A qualidade do leite é uma preocupação fundamental tanto para consumidores quanto para a indústria de laticínios. Afinal, o leite é um alimento essencial em muitas dietas ao redor do mundo, fornecendo nutrientes importantes, como proteínas, vitaminas e minerais.

A qualidade do leite pode ser influenciada por diversos fatores, desde a alimentação e cuidados com o animal até os processos de produção, armazenamento e transporte. Para garantir a qualidade do leite, é necessário monitorar e controlar esses fatores ao longo de toda a cadeia produtiva.

Os principais parâmetros utilizados para avaliar a qualidade do leite incluem a presença de contaminantes, como bactérias patogênicas, resíduos de medicamentos e substâncias químicas indesejadas. Além disso, características físicas, como cor, sabor, odor, acidez e teor de gordura, também são levadas em consideração.

A tecnologia desempenha um papel importante na garantia da qualidade do leite. Métodos avançados de análise, como a espectroscopia e a cromatografia, são utilizados para identificar e quantificar os componentes do leite, proporcionando uma avaliação precisa de sua composição e qualidade.

Além disso, a mineração de dados e a análise preditiva têm sido aplicadas no setor lácteo para prever a qualidade do leite com base em diferentes variáveis, como pH, temperatura, sabor, odor, gordura e turbidez. Essas técnicas permitem a identificação de padrões e correlações entre os parâmetros, auxiliando na tomada de decisões e no controle de qualidade.

É importante ressaltar que garantir a qualidade do leite não se resume apenas a benefícios para a saúde dos consumidores, mas também é essencial para a indústria de laticínios em termos de reputação e competitividade. A adoção de práticas de qualidade rigorosas e a utilização de tecnologias avançadas são fundamentais para assegurar um produto final seguro, saudável e de alto valor nutricional.

II. Objetivos

A. Objetivos Gerais

Utilizar os conhecimentos adquiridos na disciplina de mineração de dados para abordar um problema prático de previsão da qualidade do leite. Isso será alcançado por meio da aplicação de técnicas de seleção, pré-processamento e transformação de dados, visualização, análise descritiva, análise de grupos, classificação e estimação/regressão.

B. Objetivos Especificos

- Seleção e pré-processamento de dados;
- Normalização e redução de dados;
- Análise descritiva de dados - Visualização;
- Análise descritiva de dados - Medidas;
- Análise de grupos;
- Classificação - Arvore de Decisão;
- Classificação - KNN;
- Classificação - SVM;
- Classificação - MLP

III. Materiais e Métodos

Nesta seção, apresentaremos os materiais utilizados e a metodologia adotada para realizar o estudo de previsão da qualidade do leite por meio de técnicas de mineração de dados. Serão descritas as especificações e origem da base de dados, as etapas de pré-processamento, normalização e redução dos dados, bem como as análises descritivas, análise de grupos e classificação utilizando os algoritmos K-NN e SVM. O ambiente de desenvolvimento utilizado foi o Visual Studio Code (VSCode), e a linguagem de programação adotada foi o Python.

A. Base de Dados

A base de dados utilizada neste estudo foi obtida do repositório Kaggle [1], e consiste em observações coletadas manualmente. Esses dados são essenciais para a construção de modelos de aprendizado de máquina que visam prever a qualidade do leite.

O conjunto de dados é composto por sete variáveis independentes: pH, temperatura, sabor, odor, gordura, turbidez e cor. Esses parâmetros desempenham um papel crucial na

análise preditiva da qualidade do leite, uma vez que o grau ou qualidade do leite geralmente depende deles.

Para as variáveis sabor, odor, gordura e turbidez, é adotada uma abordagem binária, onde são atribuídos os valores 1 ou 0. Se esses parâmetros estiverem satisfeitos, é atribuído o valor 1; caso contrário, recebem o valor 0. Já as variáveis temperatura e pH do leite são representadas por seus valores reais no conjunto de dados. O pH varia de 3 a 9,5, e é considerado bom quando está no intervalo de 6,25 a 6,90. A temperatura varia de 34°C a 90°C, sendo considerada boa no intervalo de 34°C a 45,20°C.

A variável alvo neste conjunto de dados é a qualidade do leite, que é classificada em três níveis: Baixo (Ruim), Médio (Moderado) e Alto (Bom). Essa classificação é fundamental para avaliar e prever a qualidade geral do leite com base nos parâmetros mencionados.

B. Pré-processamento de Dados

Para realizar o pré-processamento dos dados, foi utilizada a linguagem de programação Python em conjunto com as bibliotecas pandas, numpy e imblearn.under_sampling. Nessa etapa, foram aplicadas técnicas de limpeza e tratamento dos dados, como identificação e tratamento de valores ausentes, remoção de outliers e normalização de valores discrepantes.

No entanto, é importante ressaltar que a base de dados utilizada já havia passado por essas etapas de pré-processamento, sendo fornecida com critérios de aprendizado. Portanto, no presente estudo, essas etapas foram realizadas apenas com o objetivo de reforçar o conhecimento e aprofundar a compreensão do processo.

Além disso, foi realizada a etapa de balanceamento da base, uma vez que ela apresentava um desequilíbrio entre as classes. Essa etapa é essencial para garantir a qualidade dos dados utilizados nas análises subsequentes, uma vez que o desbalanceamento pode afetar a capacidade de generalização dos modelos e prejudicar os resultados obtidos.

As técnicas utilizadas para o pré-processamento dos dados desempenham um papel crucial na preparação adequada dos dados e na eliminação de possíveis vieses ou distorções que possam comprometer as análises e previsões realizadas posteriormente.

C. Normalização e Redução de Dados

Após a etapa de pré-processamento, os dados passaram por um processo de normalização, com o objetivo de equalizar as escalas dos atributos. Para isso, utilizou-se a biblioteca numpy para aplicar técnicas de normalização, como a normalização por escala mínima e máxima (MinMax). Além disso, a técnica de Análise de Componentes Principais (PCA) foi aplicada para reduzir a dimensionalidade dos dados. Essa técnica busca eliminar redundâncias e extrair os principais componentes explicativos dos dados. O PCA permite uma representação mais compacta dos dados, preservando as informações mais relevantes para as análises posteriores.

D. Análise Descritiva de Dados - Visualização

Realizou-se uma análise descritiva dos dados com o objetivo de obter insights sobre as características das amostras de leite e identificar possíveis padrões ou tendências. Para isso, utilizou-se a biblioteca matplotlib para gerar visualizações gráficas, como histogramas, gráficos de dispersão e gráficos de setores.

Essas visualizações permitiram representar as distribuições de frequência dos atributos, bem como a relação entre eles e outras informações relevantes presentes na base de dados. Essa análise exploratória foi fundamental para uma compreensão mais aprofundada dos dados, fornecendo informações valiosas para as etapas subsequentes do estudo.

E. Análise Descritiva de Dados - Medidas

Foram utilizadas medidas estatísticas descritivas, como média, mediana, desvio padrão e variância, além de quartis, percentis e coeficientes de correlação, para resumir as características dos atributos e descrever a distribuição e as relações presentes nos dados. Essas medidas quantitativas complementaram as análises visuais realizadas anteriormente, fornecendo uma compreensão abrangente dos dados utilizados no estudo de previsão da qualidade do leite. Onde, as medidas de têndencia central indicam valores métricos dos atributos como: media, mediana, moda e ponto médio, buscando resumir os dados. Medidas de dispersão indicam, amplitude, desvio padrão, variância e coeficiente de variação dos atributos, buscando visualizar a variação resumida dos dados. As medidas de posição relativa medem a posição do valor em relação à média e ao desvio padrão de um conjunto de dados. As medidas de associação associam duas classes e calcula métricas que associam ambos, na base em questão, foi calculada a covariância e correlação. Covariância é uma métrica indicativa de quando dois atributos desviam juntos de sua respectiva média, onde uma covariaâcia alta indica que dois atributos mudam juntos. Correlação nada mais é do que a covariância divida pela muiltiplicação do desvio padrão dos dois atributos. Para ambas métricas, o valor 1 é o ideal, em especial na correlação, um valor de -1 indica o pior cenário possivel, com uma correlação fraca ou inexistente.

F. Análise de Grupos

A análise de grupos foi conduzida usando o algoritmo K-means após a preparação dos dados por meio do pré-processamento, balanceamento e normalização. Para facilitar a visualização dos resultados, a técnica de PCA foi empregada para reduzir a dimensionalidade dos dados para dois componentes principais. Isso permitiu plotar os resultados dos agrupamentos em um espaço bidimensional.

Além disso, medidas de avaliação foram utilizadas para quantificar a qualidade dos agrupamentos obtidos. Foram considerados coeficientes de forma, homogeneidade e outras métricas relevantes para verificar a consistência e a separação dos grupos formados. Essas medidas de avaliação forneceram insights sobre a eficácia do algoritmo K-means na formação dos grupos e ajudaram a determinar a qualidade e interpretação dos resultados obtidos.

G. Classificação - KNN

Para realizar a tarefa de classificação, utilizou-se o algoritmo K-NN (K-vizinhos mais próximos). Antes da classificação, os dados foram submetidos a uma etapa adicional de pré-processamento e normalização, garantindo a consistência e qualidade dos dados. Em seguida, a base de dados foi dividida em conjuntos de treinamento e teste, utilizando os métodos holdout (70% para treinamento e 30% para teste) e cross-validation com k=10.

Após a divisão dos dados, o algoritmo K-NN foi aplicado para classificar as amostras de leite de acordo com sua qualidade. Para avaliar o desempenho do modelo classificador, foram utilizadas métricas como matriz de confusão, acurácia e F1 Score. Essas métricas fornecem informações sobre a precisão, a taxa de acerto e o equilíbrio entre precisão e recall, permitindo uma avaliação abrangente do modelo. Esses resultados são essenciais para avaliar a eficácia do algoritmo K-NN na classificação da qualidade do leite.

H. Classificação - SVM

Além do algoritmo K-NN, o algoritmo SVM (Support Vector Machine) também foi empregado para a tarefa de classificação. Assim como no caso do K-NN, os dados passaram pelas etapas de pré-processamento e normalização antes da aplicação do algoritmo SVM. Tanto o método holdout quanto o método de cross-validation foram utilizados para dividir a base de dados em conjuntos de treinamento e teste.

O algoritmo SVM foi aplicado para classificar a qualidade das amostras de leite. Métricas como matriz de confusão, acurácia e F1 Score foram utilizadas para avaliar o desempenho do modelo classificador. Essas métricas forneceram informações sobre a capacidade de discriminação e eficácia da classificação, permitindo uma avaliação abrangente do desempenho do algoritmo SVM na classificação da qualidade do leite.

IV. RESULTADOS OBTIDOS

A seção de resultados obtidos apresenta os principais achados e conclusões desta pesquisa sobre a previsão da qualidade do leite utilizando técnicas de mineração de dados. Os resultados foram organizados em sub tópicos, cada um abordando uma tática de análise de dados específica. Essa abordagem permitirá uma análise mais detalhada e uma compreensão aprofundada dos resultados em relação aos objetivos propostos.

A. Pré-processamento de Dados

Antes de prosseguirmos com o processo de limpeza de dados, é importante verificar a dimensão da base e obter informações gerais sobre os dados. Essa etapa inicial nos permite ter uma visão geral da quantidade de registros e variáveis presentes na base, bem como entender a natureza dos dados que estamos lidando. É possivel ter esta análise completa na figura 1.

Em seguida, é necessário realizar a verificação de dados nulos na base de dados. Isso envolve a identificação de quaisquer

Fig. 1. Informações Gerais dos Dados

```
INFORMAÇÕES GERAIS DOS DADOS

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1057 entries, 0 to 1056
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   pH          1057 non-null   float64
 1   Temprature  1057 non-null   int64
 2   Taste       1057 non-null   int64
 3   Odor        1057 non-null   int64
 4   Fat         1057 non-null   int64
 5   Turbidity   1057 non-null   int64
 6   Colour      1057 non-null   int64
 7   Grade       1057 non-null   object
dtypes: float64(1), int64(6), object(1)
memory usage: 66.2+ KB
None
```

Fonte: Elaborada pelo autor

valores ausentes ou faltantes nas variáveis presentes, a fim de garantir a integridade e a qualidade dos dados utilizados na análise,como pode ser visto na figura 2, a base não apresenta dados faltantes.

Fig. 2. Valores Faltantes

```
VALORES FALTANTES

pH            0
Temprature    0
Taste         0
Odor          0
Fat           0
Turbidity     0
Colour        0
Grade         0
dtype: int64
```

Fonte: Elaborada pelo autor

Após essa etapa, é essencial verificar se os dados estão balanceados em relação à quantidade de atributos alvo de cada categoria (baixa, média e alta). Isso nos permite avaliar se existe uma distribuição equilibrada das amostras em cada classe de qualidade do leite e, como pode ser notado na figura 3, a base se encontra desbalanceada.

Fig. 3. Informações de balanceamento da base

```
Quantidade de dados por target antes do balanceamento:
low       429
medium    374
high      254
Name: Grade, dtype: int64
```

Fonte: Elaborada pelo autor

Devido ao desbalanceamento da base de dados, optou-se por aplicar a técnica de oversampling. Essa técnica foi escolhida porque a base original possuía um baixo número de registros (1059 linhas), e o undersampling reduziria ainda mais essa quantidade. O oversampling consiste em aumentar a quantidade de instâncias da classe minoritária, equilibrando as classes e permitindo que o modelo aprenda de forma

mais eficaz as características dessa classe. Isso evita o viés do modelo em direção à classe majoritária, melhorando a capacidade de previsão para a classe minoritária. Após o balanceamento cada classe alvo ficou com 429 registros, como pode ser visualizado na figura 4.

Fig. 4. Base balanceada

```
Quantidade de dados por target após o balanceamento:
low       429
medium    429
high      429
Name: Grade, dtype: int64
```

Fonte: Elaborada pelo autor

B. Normalização e Redução de Dados

O algoritmo de normalização min-max transformou os dados orignais em um novo intervalo de [0,1]. A figura 5 mostra alguns valores antes do algoritmo Min-max e a figura 6 mostra alguns atributos após a normalização utilizando Min-max. Já a figura 7 mostra a base resumida a dois componentes principais.

Fig. 5. Base antes do min-max

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	8.5	70	1	1	1	1	246	low
1	9.5	34	1	1	0	1	255	low
2	6.6	37	0	0	0	0	255	medium
3	6.6	37	1	1	1	1	255	high
4	5.5	45	1	0	1	1	250	low
5	4.5	60	0	1	1	1	250	low
6	8.1	66	1	0	1	1	255	low
7	6.7	45	1	1	0	0	247	medium
8	6.7	45	1	1	1	0	245	medium
9	5.6	50	0	1	1	1	255	low

Fonte: Elaborada pelo autor

Fig. 6. Base após o min-max

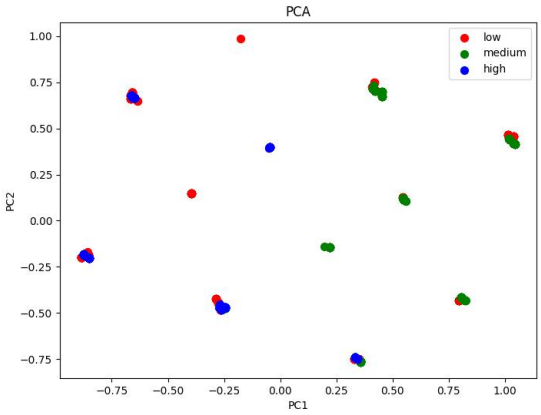
	pH	Temprature
count	1287.000000	1287.000000
mean	0.559907	0.170149
std	0.195489	0.168168
min	0.000000	0.000000
25%	0.538462	0.071429
50%	0.569231	0.107143
8	0.569231	0.196429	1.0
9	0.400000	0.285714	0.0

Fonte: Elaborada pelo autor

.

OBS: Na aplicação do PCA, muitas classes ficaram sobrepostas pelo fato de que muitos atributos possuiam faixa de valores bem próximas.

Fig. 7. PCA aplicado na base

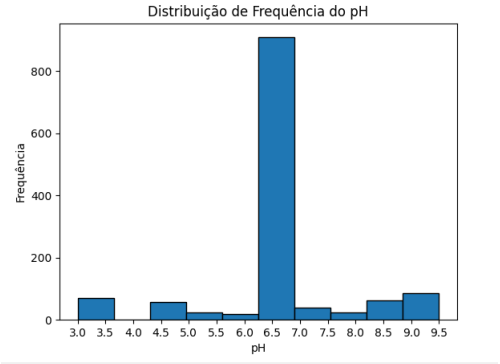


Fonte: Elaborada pelo autor

C. Análise Descritiva de Dados - Visualização

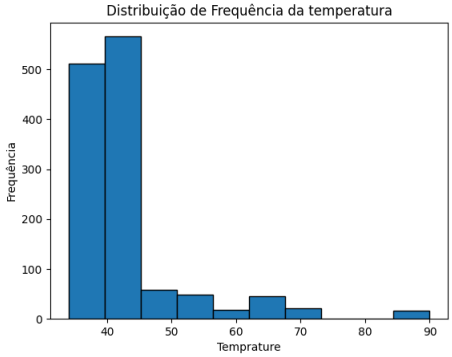
No que tange a descrição da base de dados através da visualização de seus atributos, foram obtidos os resultados a seguir.

Fig. 8. Histograma pH



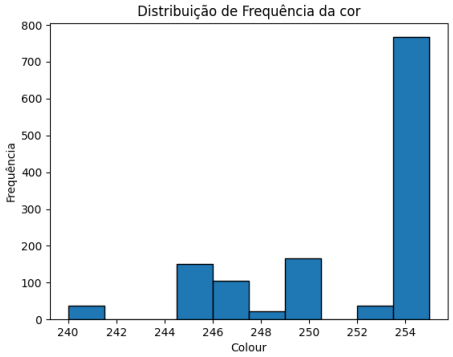
Fonte: Elaborada pelo autor

Fig. 9. Histograma Temperatura



Fonte: Elaborada pelo autor

Fig. 10. Histograma cor



Fonte: Elaborada pelo autor

D. Análise Descritiva de Dados - Medição

Com a base de dados limpa, com o intuito de obter um resumo métrico da base:

1) Medidas de tendencia central: aplicada nos valores numéricos.

Temperatura
- Media: 43.52836052836053
- Mediana: 40.0
- Ponto médio: 62.0
- Moda: 45.0

pH
- Media:6.639393939393939
- Mediana:6.7
- Ponto médio:6.25
- Moda:6.8

Cor
- Media: 251.87412587412587
- Mediana: 255.0
- Ponto médio:247.5
- Moda:255

2) Medidas de dispersão: aplicada nos valores numéricos

Temperatura
- Amplitude: 56
- Desvio padrão: 0.09417428452771771
- Variância: 88.6879586630753
- Coeficiente de variação: 21.635155421569177

pH
- Amplitude:6.5
- Desvio padrão: 0.01270680475317081
- Variância: 1.614628870352043
- Coeficiente de variação: 19.138500997473155

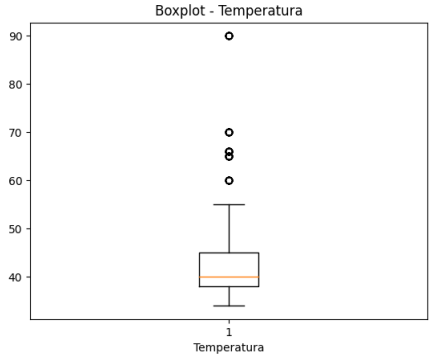
Cor
- Amplitude: 15
- Desvio padrão:0.043051931091782646
- Variância:18.534687707316014
- Coeficiente de variação: 1.7092637420525623

3) Medidas de posição relativa: :

Quartis:
- 0.25 38.0
- 0.50 40.0
- 0.75 45.0

Escore-z:
- 2.810920
- -1.011779
- -0.693221
- -0.693221
- 0.156268
- ...
- -0.268477
- -0.799407
- -0.799407
- 0.156268
- -0.587035

Fig. 11. Diagrama de caixas - Temperatura



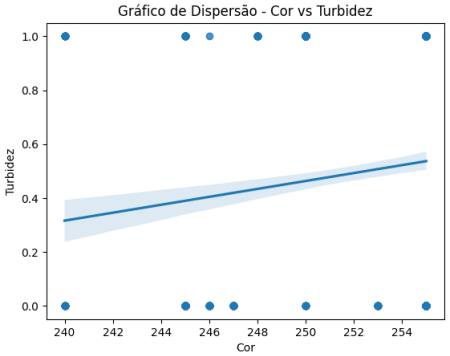
Fonte: Elaborada pelo autor

4) Medidas de associação: :

Covariância entre Cor e Turbidez: 0.2725913278012814

Correlação entre Cor e Turbidez: 0.12660474045084952

Fig. 12. Diagrama de dispersão com reta para visualização da variação da dispersão



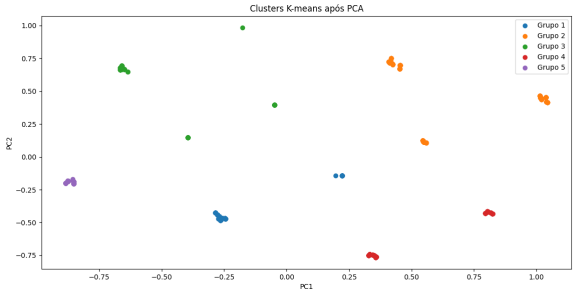
Fonte: Elaborada pelo autor

E. Análise de Grupos

Utilizando a base limpa e o PCA, aplicando os algortimos K-Means e GMM, com o valor de ”K” variando entre 3,5 e 10. No algoritmo K-means também foi utilizado as distancias Euclidiana e Manhattan.

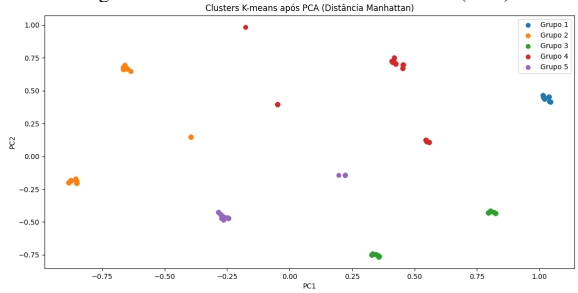
OBS: Na aplicação dos algoritmos de agrupamento, muitos grups ficaram sobrepostos pelo fato de que muitos atributos possuiam faixa de valores bem próximas.

Fig. 13. K-kmeans - Distancia Euclidiana (K=5)



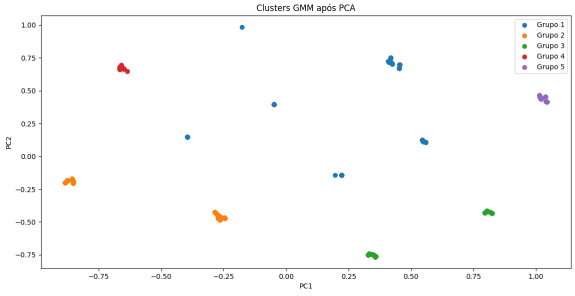
Fonte: Elaborada pelo autor

Fig. 14. K-kmeans - Distancia Manhattan (K=5)



Fonte: Elaborada pelo autor

Fig. 15. GMM (K=5)



Fonte: Elaborada pelo autor

Foi utilizado duas medidas para avaliar a qualidade dos agrupamentos realizados: o Coeficiente de forma e a Homogenidade.

1) Coeficiente de forma

: K-Means (Distancia Euclidiana)

K3 = 0.40246724529144123

K5 = 0.20268239032562355

K10 =0.023440377867055884

K-Means (Distancia Manhattan)

K3 = 0.45895422253875

K5 = 0.224443811142217

K10 = 0.1038507034678613

GMM

K3 = 0.4105691464559676

K5 = 0.3373034798290252

K10 = 0.09551553478201857

2) Homogenidade

: K-Means (Distancia Euclidiana)

K3 = 0.2766691913417482

K5 = 0.30356149881449795

K10 = 0.36749783082413234

K-Means (Distancia Manhattan)

K3 = 0.24936000363501326

K5 = 0.24441188763277777

K10 = 0.35451597076560104

GMM

K3 = 0.20860604862337456

K5 = 0.2620263757465329

K10 = 0.36749783082413245

Onde o coeficiente de forma mede o quão compacto sao os agrupamentos feitos pelos algoritmos(quão próximo cada grupo está um do outro). Homogeneidade mede o quão bem um os grupos mantém apenas uma única classe ou atributo, na base em questão, foi utilizada a homogeneidade de rotulos. Para ambas medidas, bons indicativos de valores são os mais próximos de 1.

F. Técnicas de Divisão de Dados Utilizadas na Classificação

A técnica de divisão é um procedimento comum em aprendizado de máquina para separar conjuntos de dados em subconjuntos diferentes, geralmente com o objetivo de treinamento e teste de modelos. Duas técnicas populares de divisão são a validação cruzada (cross-validation) e o holdout.

Essas técnicas de divisão são úteis para avaliar o desempenho do modelo, garantir que ele generalize bem para novos dados e evitar o superajuste (overfitting). A escolha da técnica de divisão adequada depende da natureza dos dados e dos objetivos do projeto de aprendizado de máquina.

1) Cross-validation: É uma técnica utilizada para avaliar o desempenho de um modelo de aprendizado de máquina. Ela envolve dividir o conjunto de dados em partes chamadas ”folds” e realizar repetidas vezes o treinamento e teste do modelo em diferentes combinações desses folds, garantindo que todos os dados sejam utilizados tanto para treinamento quanto para teste. Podemos ver um exemplo de codificação deste método na figura 16.

Fig. 16. Código Cross-validation

```
# Dividir os dados com Cross-Validation k=10

scores = cross_val_score(knn_cv, X, y, cv=10)
```

Fonte: Elaborada pelo autor

2) Holdout: É uma técnica de divisão de dados em que o conjunto de dados é separado em duas partes: um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento é usado para treinar o modelo, enquanto o conjunto de teste é usado para avaliar a performance do modelo em dados não vistos anteriormente. Podemos ver um exemplo de codificação deste método na figura 17.

Fig. 17. Código Holdout

```
# Dividir os dados em treinamento e teste usando Holdout
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Fonte: Elaborada pelo autor

G. Métricas Utilizadas na Classificação

Métricas são medidas utilizadas para avaliar o desempenho de modelos de aprendizado de máquina. Elas fornecem uma maneira objetiva de quantificar o quão bem um modelo está realizando suas previsões. Existem várias métricas comuns usadas em diferentes tipos de problemas, como classificação, regressão ou clustering.

Em problemas de classificação, métricas comuns incluem a acurácia, que mede a proporção de exemplos classificados corretamente, e o F1-score, que combina a precisão e a revocação em uma única medida para avaliar o equilíbrio entre precisão e abrangência do modelo.

1) F1-score: É uma métrica de avaliação comumente utilizada para medir o desempenho de modelos de classificação. Ela combina a precisão (proporção de verdadeiros positivos em relação ao total de predições positivas) e a revocação (proporção de verdadeiros positivos em relação ao total de exemplos positivos reais) em uma única medida, fornecendo uma visão geral do equilíbrio entre precisão e revocação. Podemos ver um exemplo de codificação deste método na figura 18.

Fig. 18. Código F1-score

```
# Avaliar a acuracia
accuracy_holdout = accuracy_score(y_test, y_pred_holdout)
```

Fonte: Elaborada pelo autor

2) Acurácia: É uma métrica utilizada para medir a performance de modelos de classificação, representando a proporção de exemplos classificados corretamente em relação ao total de exemplos. É uma medida simples, mas pode ser enganosa quando há classes desbalanceadas. Podemos ver um exemplo de codificação deste método na figura 19.

Fig. 19. Código Acurácia

```
#Avaliar o F1-score
f1_holdout = f1_score(y_test, y_pred_holdout, average='macro')
```

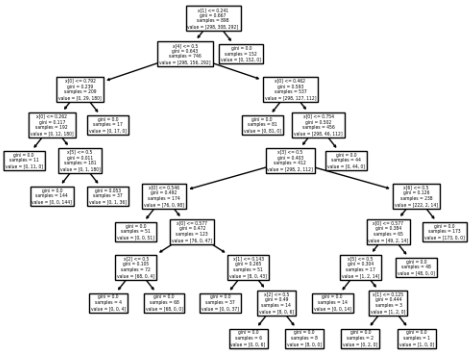
Fonte: Elaborada pelo autor

H. Classificação - Árvore de Decisão

A árvore de decisão é um algoritmo de aprendizado de máquina que utiliza uma estrutura em forma de árvore para tomar decisões com base em condições lógicas. Ela é construída a partir de um conjunto de dados rotulados, onde cada nó interno representa um teste em uma determinada característica do dado, cada ramo representa uma possível resposta a esse teste, e cada folha representa uma decisão ou um valor previsto.

Foi implementado o algoritmo de Árvore de Decisão utilizando a base já pré-processada e balanceada e a classe DecisionTreeClassifier do módulo sklearn.tree do scikit-learn. A árvore resultante possui 18 nós, sendo esse número determinado após testes para maximizar a acurácia. Como pode ser visto na figura 20

Fig. 20. Árvore de Decisão



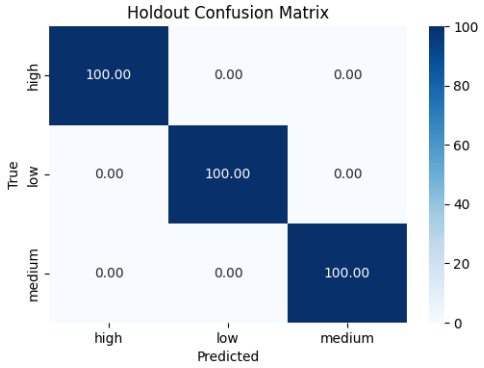
Fonte: Elaborada pelo autor

Foram utilizados dois métodos de divisão de dados, a primeira foi a Divisão Hougout, que apresentou 100% de acurácia e F1-score igual a 1. O que significa, que com este modelo de divisão e com o algoritmo de arvore de decisão, todos os dados foram classificados corretamente, como pode ser visto na figura 21, que apresenta a matriz de confusão.

Em seguida foi utilizado o método de divisão Cross-Validation, que apresentou 99,69% de acurácia e F1-score igual a 0.99688, errando apenas 4 classificações, onde o pior caso se encontra no dado previsto como qualidade alta, onde na verdade era baixa, como pode ser visto na figura 22, o que leva-se a considerar que um leite impróprio para consumo poderia ser consumido.

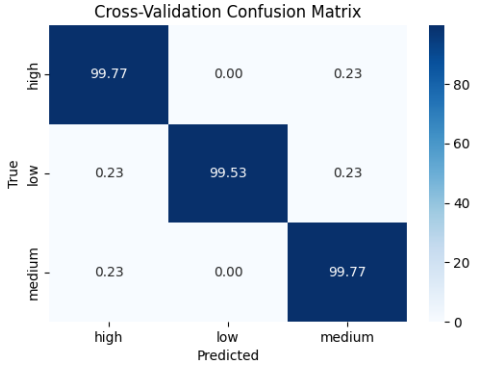
No geral, o algoritmo demonstrou um desempenho excelente, apresentando um índice de erro muito baixo.

Fig. 21. Matriz de Confusão Árvore de Decisão Hougout



Fonte: Elaborada pelo autor

Fig. 22. Matriz de Confusão Árvore de Decisão Cross-Validation



Fonte: Elaborada pelo autor

I. Classificação - K-Mearest Neighbors(KNN)

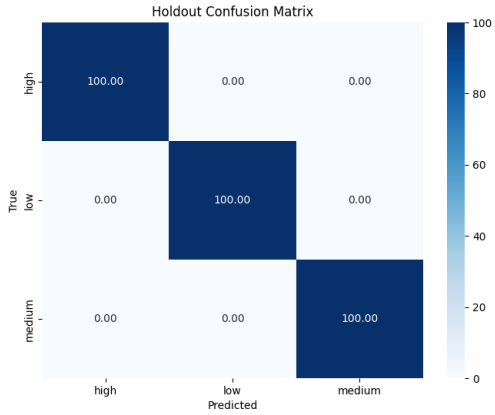
O KNN (K-Nearest Neighbors) é um algoritmo de aprendizado de máquina usado para classificação e regressão. Ele se baseia na proximidade entre objetos semelhantes. No KNN, os objetos são classificados com base na classe majoritária dos K vizinhos mais próximos. O valor de K determina a quantidade de vizinhos considerados.

Após realizar o pré-processamento e balanceamento da base de dados, foi implementado o algoritmo KNN utilizando a classe KNeighborsClassifier do módulo sklearn.neighbors do scikit-learn. Foi escolhido um valor de K igual a 5, pois obteve o melhor resultado após testes. Verificou-se que ao aumentar esse parâmetro, as métricas não se alteravam.

Assim como na Árvore de Decisão, foram utilizados dois métodos de divisão da base de dados. O primeiro método foi o Holdout, que resultou em uma acurácia de 100% e um F1-Score de 1, indicando que o algoritmo não cometeu erros nas classificações. Isso pode ser confirmado na Matriz de Confusão exibida na figura 23.

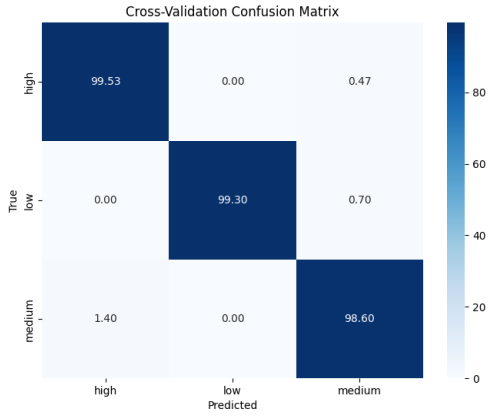
O segundo método utilizado foi o Cross-Validation, que resultou em uma acurácia de 99.1437% e um F1-Score de 0.9915. Isso significa que o algoritmo teve um desempenho excepcional, cometendo apenas 10 erros. Esses erros ocorreram ao classificar incorretamente 3 casos de leite de baixa qualidade como média qualidade e 6 casos de leite de média qualidade como alta qualidade e levar 2 leites de alta qualidade a serem classificados como de média qualidade. A Figura 24 apresenta a matriz de confusão com essas informações.

Fig. 23. Matriz de Confusão KNN Cross-Validation



Fonte: Elaborada pelo autor

Fig. 24. Matriz de Confusão KNN Cross-Validation



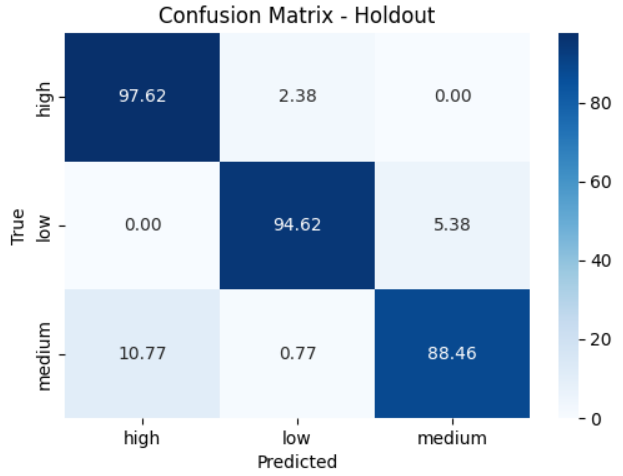
Fonte: Elaborada pelo autor

J. Classificação - Support Vectors Machine(SVM)

Support Vector Machine (SVM) é um algoritmo de aprendizado de máquina que é usado tanto para problemas de classificação quanto para problemas de regressão. Ele funciona encontrando um hiperplano que melhor separa os pontos de diferentes classes em um espaço multidimensional. Os pontos mais próximos do hiperplano são chamados de vetores de suporte e são essenciais para a construção do modelo. O SVM busca maximizar a margem entre os vetores de suporte e o hiperplano, tornando-o robusto a novos dados. Além disso, o SVM pode usar uma função de kernel para lidar com dados não linearmente separáveis, mapeando-os para um espaço de maior dimensão onde a separação linear é possível. O

SVM é eficiente em problemas com alta dimensionalidade e é amplamente utilizado em diferentes domínios. O código implementado realiza uma Support Vector Machine (SVM) para classificação usando a classe SVC do módulo sklearn.svm do scikit-learn. Ele utiliza um kernel polinomial, pois, foi o que apresentou o melhor resultado dentre ele, o linear e o rbf, e trabalha com um banco de dados pré-processado e balanceado. O desempenho do modelo é avaliado usando o método Holdout e a validação cruzada, e as métricas de avaliação, juntamente com as matrizes de confusão, são exibidas. Inicialmente foi executada usando método de divisão Holdout, que resultou em uma acurácia de 93.5233% e um F1-Score de 0.9351. isso pode ser confirmado na Matriz de Confusão exibida na figura 25.

Fig. 25. Matriz de Confusão SVM Cross-Validation



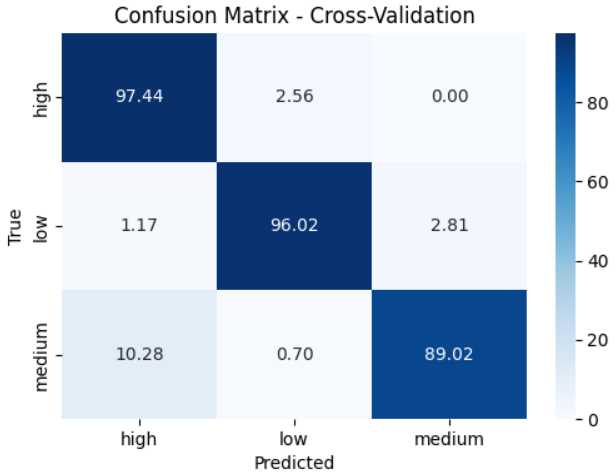
Fonte: Elaborada pelo autor

Em seguida foi utilizado o método de divisão Cross-Validation, que apresentou 94.1589% de acurácia e F1-score igual a 0.9416, errando poucas classificações, onde o pior caso se encontra no dado previsto como qualidade alta, onde na verdade era baixa, representando 0,1 e alta como baixa, que representa 0,03, como pode ser visto na figura 26, o que leva-se a considerar, sendo o pior caso, que um leite impróprio para consumo poderia ser consumido ou que um leite de ótima qualidade fosse descartado.

K. Classificação - Rede Neural Multilayer Perceptron(MLP)

A MLP (Multilayer Perceptron) é um tipo de rede neural artificial que consiste em múltiplas camadas de neurônios interconectados. Cada neurônio em uma camada recebe inputs das unidades da camada anterior, realiza uma combinação linear desses inputs e passa o resultado por uma função de ativação não linear. As camadas intermediárias da MLP são conhecidas como camadas ocultas, enquanto a última camada é a camada de saída. A MLP é capaz de aprender relações complexas entre as features de entrada e as classes de saída através do processo de treinamento, onde os pesos das conexões entre os neurônios são ajustados iterativamente.

Fig. 26. Matriz de Confusão SVM Cross-Validation



Fonte: Elaborada pelo autor

É um algoritmo amplamente utilizado em problemas de classificação e regressão, e pode ser aplicado a uma variedade de domínios, como visão computacional, processamento de linguagem natural e reconhecimento de voz. Neste caso, o modelo possui 2 camadas ocultas, onde cada camada tem 20 neurônios. O número de neurônios em cada camada oculta é especificado pela tupla (20, 20). Você pode ajustar esses valores para alterar o número de camadas ocultas e o tamanho de cada camada.

O código realizado, é uma implementação de um classificador de rede neural MLP (Multi-Layer Perceptron) usando a classe MLPClassifier do módulo sklearn.neural_network do scikit-learn. Neste caso, o modelo possui 2 camadas ocultas, onde cada camada tem 20 neurônios. O número de neurônios em cada camada oculta é especificado pela tupla (20, 20). Você pode ajustar esses valores para alterar o número de camadas ocultas e o tamanho de cada camada.

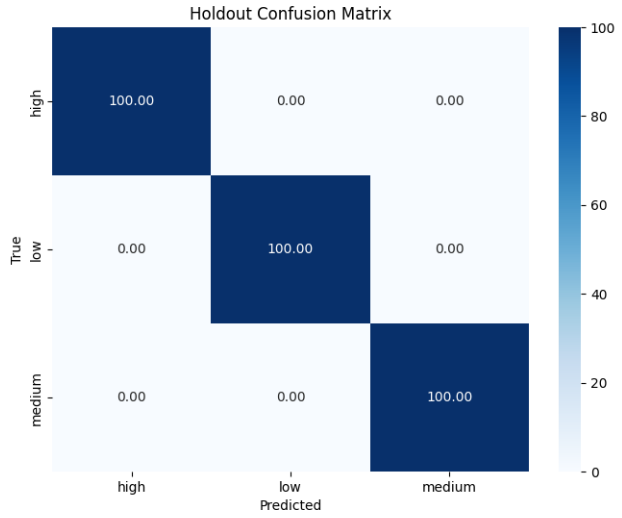
Além disso, o número máximo de iterações é definido como 2000. Isso indica o número máximo de vezes que o otimizador será executado durante o treinamento do modelo. Se o número máximo de iterações for alcançado antes de a convergência ser atingida, um aviso será exibido.

Estes valores foram especificados com base em testes de execução, utilizando-se de dois métodos de divisão, sendo eles o Holdout e Cross-Validation.

Com Holdout foi possivel alcançar uma acurácia de 100% e um F1-Score igual a 1, não tendo classificado nenhum caso incorretamente, como pode ser visto na matriz de correlação disponível na figura 27

Com o uso do método de Cross-Validation, embora não tenha sido possível encontrar um classificador perfeito, obteve-se uma taxa de acerto bastante alta. O classificador cometeu apenas 4 erros de classificação, resultando em uma acurácia de 99,69% e um F1-Score de 0,99. É importante ressaltar que, felizmente, esses erros não incluem os piores casos, sendo apenas 1 caso em que um leite de baixa qualidade foi classificado

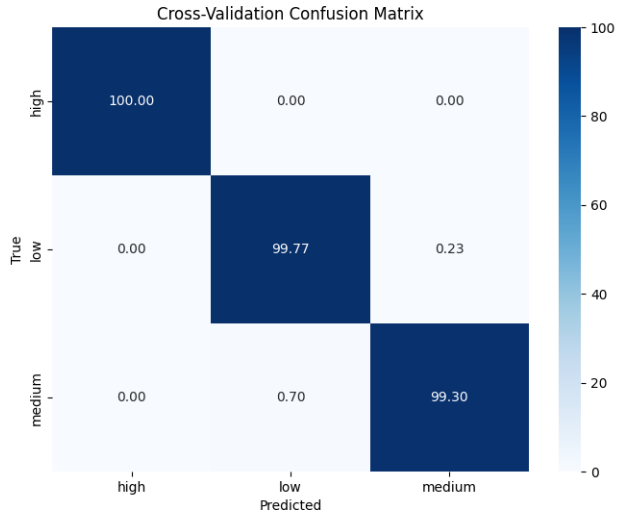
Fig. 27. Matriz de Confusão MLP Cross-Validation



Fonte: Elaborada pelo autor

como média e 3 casos em que um leite de qualidade média foi classificado como de baixa qualidade. Esses resultados podem ser visualizados na matriz de correlação da figura 28.

Fig. 28. Matriz de Confusão MLP Cross-Validation



Fonte: Elaborada pelo autor

V. Analise e Discussão

Os resultados obtidos foram bastante promissores. No que tange a etapa de pré-processamento, houve grande simplicidade, posto que a base de dados em questão não possuia dados faltantes ou anomalias exorbitantes, resumindo essa etapa no processo de balanceamento.

A etapa de transformação de dados foi de fácil aplicação, pois muitos atributos eram categóricos com valores binários, além disso as faixas de valores não tinham grande variação, transformando o algoritmo min-max em uma ótima aplicação. Tais fatores citados também corroboraram na facilidade de visualização de resumos criados para a base de dados.

No que se refere ao agrupamento, houve sobreposição de muitos grupos, e a qualidade em todos algoritmos variou de baixa a moderada, existindo a possibilidade de otimização nesses algortimos.

Em relação à acurácia, todos os algoritmos alcançaram resultados excelentes tanto na validação holdout quanto na validação cruzada.

No caso do algoritmo KNN, a acurácia obtida foi de aproximadamente 100% e um f1-Score de 1 na validação holdout e 99.1437% e um f1-Socre de 0,9915 na validação cruzada. Isso indica que o KNN foi capaz de fazer previsões precisas na classificação da qualidade do leite.

A árvore de decisão também apresentou um desempenho notável. Com o uso da validação holdout, o algoritmo alcançou uma acurácia de cerca de 100% e um f1-Score de 1, e na validação cruzada obteve uma acurácia média de 99,69% e um f1-Score de 0,99688. Esses resultados destacam a capacidade do algoritmo de tomar decisões precisas com base nos atributos fornecidos.

O algoritmo SVM mostrou resultados igualmente impressionantes. Com a validação holdout, obteve uma acurácia de aproximadamente 93.5233% e um f1-Score de 0,9351, e na validação cruzada alcançou uma acurácia média de 94.1589% e um f1-Score de 0,9416. Esses resultados confirmam a eficiência do SVM na classificação da qualidade do leite.

Além disso, o algoritmo MLP também obteve um desempenho notável. Na validação holdout, o MLP alcançou uma acurácia de cerca de 100% um f1-Score de 1, enquanto na validação cruzada obteve uma acurácia média de 99,69% um f1-Score de 0,997. Isso ressalta a capacidade do MLP de aprender e generalizar padrões complexos nos dados.

Todos os algoritmos obtiveram resultados excelentes, demonstrando a capacidade de lidar com a classificação multiclasse de forma precisa e equilibrada. Podemos ver na tabela da figura 29 os resultados gerais.

Fig. 29. Tabela de Resultados de classificação

Métodos	Holdout		Cross-Validation	
Métricas	Acurácia	F1-Score	Acurácia	F1-Score
Árvore de Decisão	100%	1	99,69%	0.99688
K-Nearest Neighbors(KNN)	100%	1	99.1437%	0.9915
Support Vectors Machine(SVM)	93.5233%	0.9351	94.1589%	0.9416
Rede Neural Multilayer Perceptron (MLP)	100%	1	99.6890%	0.9969

Fonte: Elaborada pelo autor

VI. Conclusão

Com base nos resultados obtidos, pode-se concluir que o trabalho foi realizado com sucesso. Os algoritmos de aprendizado de máquina, incluindo KNN, árvore de decisão, SVM e MLP, demonstraram excelentes desempenhos em termos de acurácia e F1-score na classificação da qualidade do leite.

Com os trabalhos feitos na base, ficou possivel concluir que, para a qualidade de um bom leite depende de um pH entre 6.25 a 6.90, uma temperatura entre 34 ºC a 45.20 ºC, com odor e gosto agradável ao consumidor, baixo valor de gordura, sem turbidez, e cor preferencialmente o mais próximo possível do branco natural.

Esses resultados são promissores e indicam que esses algoritmos podem ser aplicados em cenários reais para auxiliar na tomada de decisões relacionadas à qualidade do leite. No entanto, é importante ressaltar que cada algoritmo tem suas próprias características e pode ser mais adequado para diferentes tipos de problemas. Portanto, é recomendado realizar estudos adicionais e considerar outros fatores, como tempo de treinamento e interpretabilidade do modelo, ao escolher o algoritmo mais adequado para uma determinada tarefa de classificação.

References

[1] RAJENDRAN, Shrijayan. Milk Quality Prediction. Kaggle.com. Disponível em: ¡https://www.kaggle.com/datasets/cpluzshrijayan/milkquality?resource=download¿. Acesso em: 23 abr. 2023.