

{

Análise da aplicação de  
Machine Learning em um  
dataset sobre vinhos

}

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

01

Proposta

- Objetivo principal

02

Análise Inicial

- Exploração do banco de dados
- Aplicação de estimador *Base Line*

03

Resultados Preliminares

- Resultados iniciais de diferentes estimadores
- Comparação de resultados

04

Resultados Finais

- Novas abordagens
- Otimização de modelos
- Resultados finais

05

Conclusões

- Conclusão sobre modelo final proposto
- Sugestões de uso do modelo

{

O objetivo do trabalho é fazer uso de algoritmos de aprendizado de máquinas para analisar as características e prever a qualidade de vinhos com base em suas propriedades químicas, como teor alcoólico, acidez, pH e outros fatores.

}

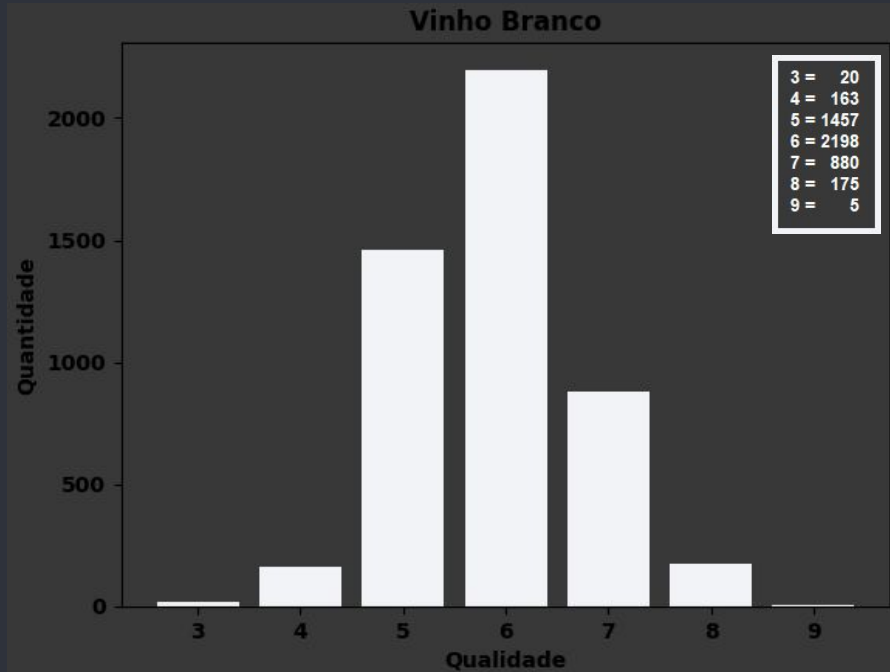
Propriedades químicas do vinho presentes no banco de dados:

- Acidez fixa;
- Acidez volátil;
- Ácido cítrico;
- Açúcar residual;
- Cloretos;
- Dióxido de Enxofre livre;
- Dióxido de Enxofre total;
- Densidade;
- pH;
- Sulfatos;
- Álcool.

Os vinhos são classificados por nota (qualidade), entre 0 e 10, onde:

- Nota 0 = péssimo
- Nota 5 = mediano
- Nota 10 = excelente

A quantidade de vinhos, por nota (qualidade) no banco de dados é:



Podemos perceber que existe uma discrepância significativa entre as qualidades dos vinhos.

Assim, dizemos que este banco de dados está desbalanceado.

# Verificação da existência de dados Null ou NaN no banco de dados:

Dados Null são espaços vazios dentro do banco de dados, ou seja, linhas que não foram preenchidas com valores.

Dados NaN (ou *Not a Number*, do inglês) são valores que não são considerados números válidos para operações matemáticas.

Na coluna em destaque, tem-se 4.898 dados *non-null* (de um total de 4.898). Logo, não existem valores Null ou NaN no banco de dados.

```
1 df_winewhite.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 4898 entries, 0 to 4897
```

```
Data columns (total 12 columns):
```

#	Column	Non-Null Count	Dtype
0	fixed acidity	4898 non-null	float64
1	volatile acidity	4898 non-null	float64
2	citric acid	4898 non-null	float64
3	residual sugar	4898 non-null	float64
4	chlorides	4898 non-null	float64
5	free sulfur dioxide	4898 non-null	float64
6	total sulfur dioxide	4898 non-null	float64
7	density	4898 non-null	float64
8	pH	4898 non-null	float64
9	sulphates	4898 non-null	float64
10	alcohol	4898 non-null	float64
11	quality	4898 non-null	int64

```
dtypes: float64(11), int64(1)
```

```
memory usage: 459.3 KB
```

# Como nossos dados estão correlacionados?

Dados correlacionados representam a influência de alguns componentes na qualidade do vinho.

As principais correlações são, respectivamente:

- Álcool;
- Densidade;
- Cloretos.

alcohol	1	-0.78	-0.36	0.44
density	-0.78	1	0.26	-0.31
chlorides	-0.36	0.26	1	-0.21
quality	0.44	-0.31	-0.21	1
	alcohol	density	chlorides	quality

## E em relação aos outliers?

Outliers, mais conhecidos como pontos fora da curva, são pontos que podem prejudicar a performance do nosso modelo de aprendizado de máquina.

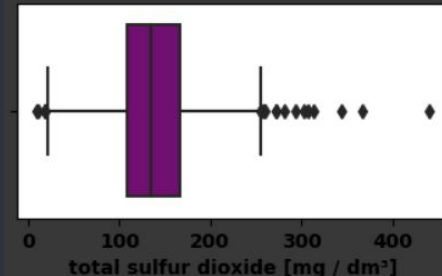
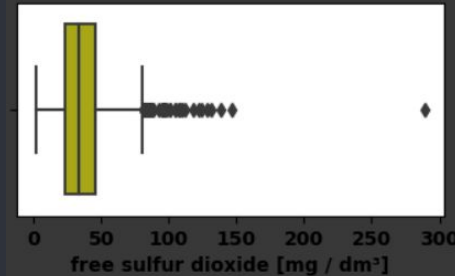
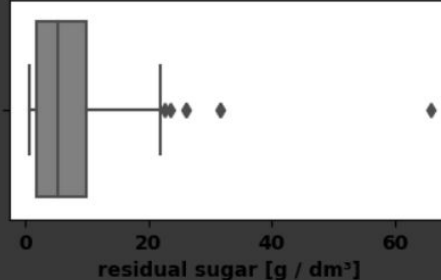
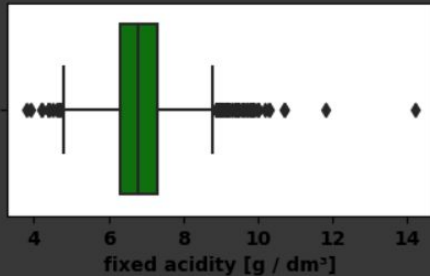
As principais colunas que apresentam outliers entre seus dados são as mostradas ao lado.

	fixed acidity	residual sugar	free sulfur dioxide	total sulfur dioxide
count	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	6.391415	35.308085	138.360657
std	0.843868	5.072058	17.007137	42.498065
min	3.800000	0.600000	2.000000	9.000000
25%	6.300000	1.700000	23.000000	108.000000
50%	6.800000	5.200000	34.000000	134.000000
75%	7.300000	9.900000	46.000000	167.000000
max	14.200000	65.800000	289.000000	440.000000



Para visualização dos outliers, é comum o uso de um gráfico chamado *boxplot*.

Abaixo, vemos os *boxplots* das quatro colunas citadas anteriormente, sendo possível perceber os outliers (pontos além dos *T's* dos *boxs* centrais).



{

Para finalizar a análise inicial, foi realizada a aplicação do estimador *Base Line*. Esta nomenclatura é dada para o modelo que usaremos como base para decidir se os resultados encontrados com uso de outros estimadores são melhores ou piores que nossa base.

Ou seja, ele é usado para que possamos ter uma base a partir da qual podemos aperfeiçoar nosso modelo final, buscando sempre melhores resultados.

}

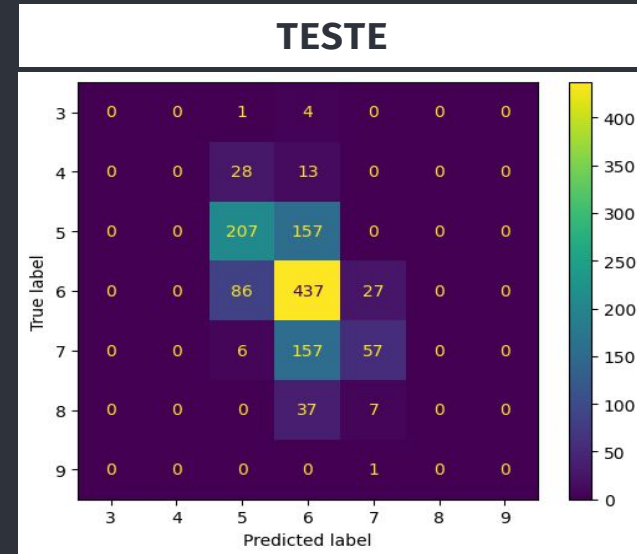
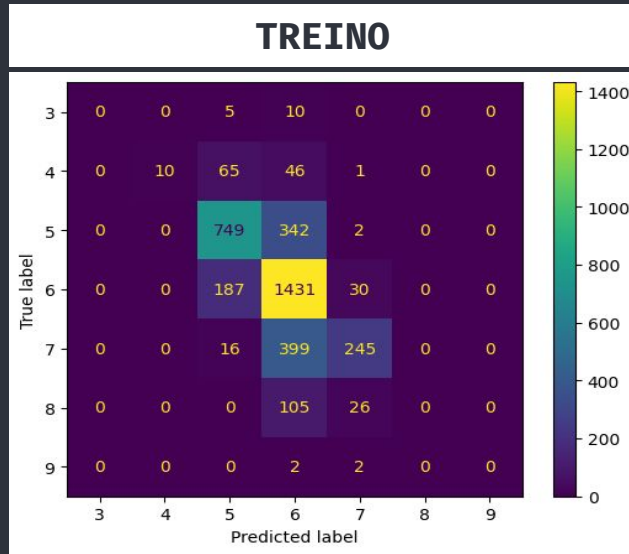
O *Base Line* utilizado foi um **Random Forest**, o qual tivemos como melhores parâmetros:

- Números de estimadores: 180
- Máximo de *features*: 2
- Máximo de profundidade: 7

As métricas alcançadas, para o treinamento do modelo e para o teste, foram:

MODELO	TREINO	TESTE
• Score: 0,5809	• Acurácia: 66%	• Acurácia: 57%

A matriz de confusão do modelo *Base Line* é apresentada abaixo, tanto para o treino como para o teste. É possível perceber que o modelo teve uma precisão maior para qualidades entre 5 e 7, uma vez que essas são as qualidades com maior quantidade de dados no banco de dados.



A fim de **melhorar** os resultados, algumas modificações foram realizadas nos **parâmetros** e nos **estimadores**, como dito anteriormente. Os resultados obtidos, com as métricas do treino e teste, são apresentados na tabela abaixo.

Modelos e técnicas analisadas	Score	Acurácia	
		Treino	Teste
Random Forest com SMOTE	0,5328	61%	54%
Random Forest com <i>Feature Selection</i>	0,5556	61%	55%
Random Forest com SMOTE sem OUTLIERS	0,5303	69%	57%
Support Vector Classifier com SMOTE	0,5688	77%	58%
Support Vector Classifier sem SMOTE	0,6371	66%	57%
Support Vector Classifier com SMOTE sem OUTLIERS	0,5900	95%	63%
AdaBoost sem SMOTE	0,4868	46%	45%
AdaBoost com SMOTE	0,4386	48%	47%
Random Forest (modelo <i>Base Line</i> )	0,5809	66%	57%

{

Por mais que o modelo de Support Vector Classifier com SMOTE e sem os outliers tenha apresentado uma acurácia no treino de 63%, é possível perceber que este modelo possa estar enviesado, uma vez que sua acurácia no treino foi de 95%.

Logo, os resultados preliminares demonstram que é necessário o tratamento dos dados, principalmente em relação ao desbalanceamento do banco de dados.

}

## < Nova abordagem >

Como possível **solução** para a questão da acurácia baixa, o problema foi dividido em duas etapas:

1º → Classificar o vinho como **bom** ou **ruim**;

2º → Diferenciar a classificação do vinho **bom**.

## < Novos algoritmos >

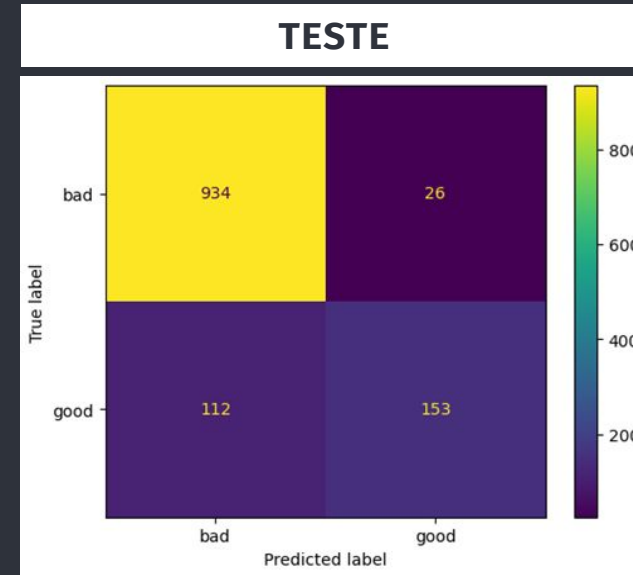
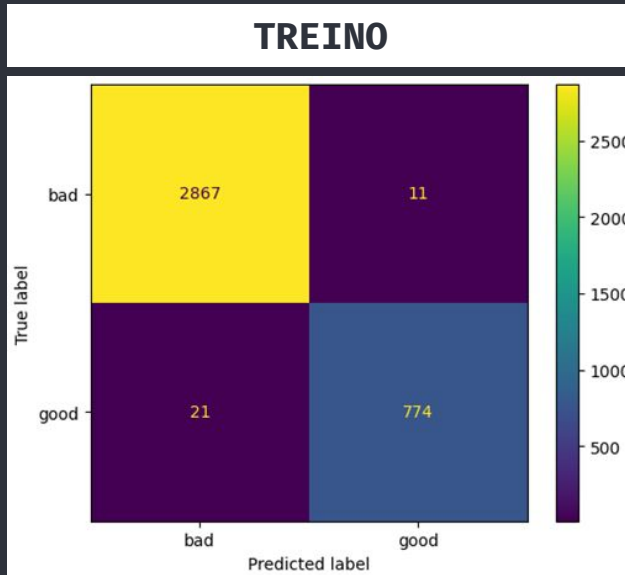
O primeiro algoritmo classificou cada vinho entre vinho bom e vinho ruim, onde:

- Vinho ruim: aquele cuja qualidade é  $\leq 6$ ;
- Vinho bom: aquele cuja qualidade é  $\geq 7$ .

O segundo algoritmo irá diferenciar a qualidade dos vinhos bons, ou seja, atribuir novamente a qualidade 7, 8 ou 9 SOMENTE para os vinhos que foram classificados como bons.



A matriz de confusão do primeiro algoritmo do modelo final é apresentada abaixo, para o treino e teste. É possível perceber a alta acurácia nos dois casos.

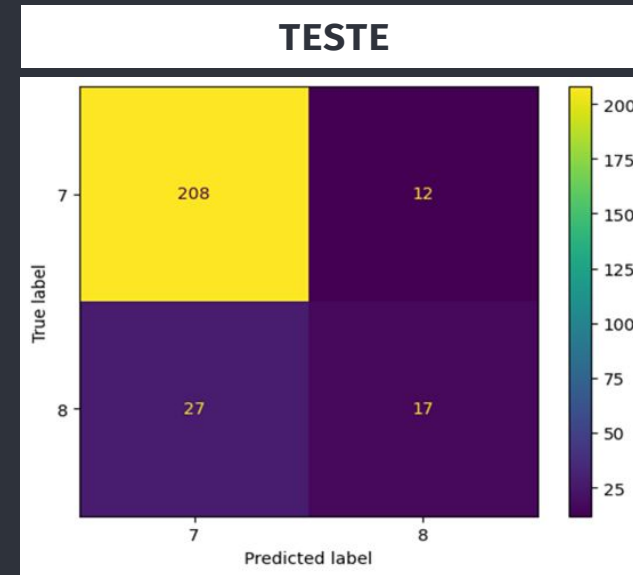
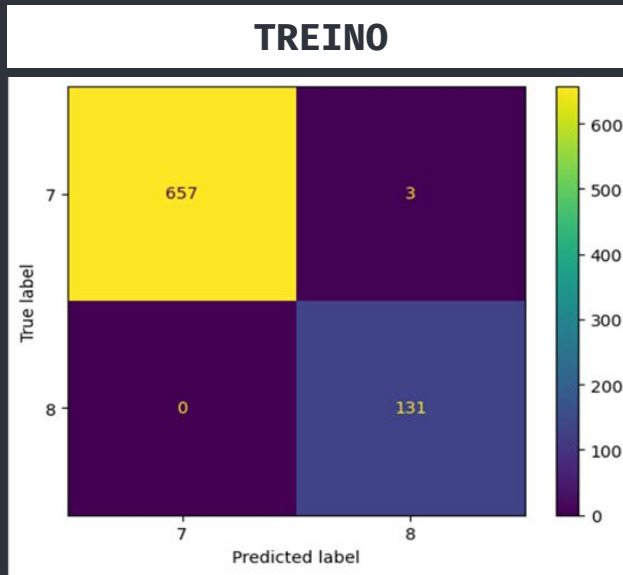


Como resultado, o modelo alcançou as seguintes métricas para o treino e teste:

MODELO	TREINO	TESTE
• Score: 0,8560	• Acurácia: 99%	• Acurácia: 89%

Desta vez, no entanto, podemos dizer que o modelo não está enviesado, quando considerada sua alta acurácia no treino, uma vez que o banco de dados está menos desbalanceado e os possíveis resultados do treino são apenas dois: ou bom ou ruim, e não mais 7 diferentes qualidades, o que fazia diminuir a acurácia.

A matriz de confusão do segundo algoritmo do modelo final é apresentada abaixo, para o treino e teste. É possível perceber, mais uma vez, a alta acurácia nos dois casos.



Como resultado, o modelo alcançou as seguintes métricas para o treino e teste:

MODELO	TREINO	TESTE
• Score: 0,8282	• Acurácia: 100%	• Acurácia: 85%

Novamente, podemos dizer que o modelo não está enviesado, uma vez que os possíveis resultados do treino são apenas dois: ou qualidade 7 ou qualidade 8, o que melhora na acurácia.








< OBS: a qualidade 9 não foi utilizada neste segundo algoritmo, uma vez que representa uma quantidade extremamente pequena do banco de dados, o que poderia influenciar em piores resultados na acurácia do modelo final >

Para concluir, os resultados adquiridos no modelo desenvolvido neste projeto poderiam ser usados, por exemplo, por produtores de vinho para melhorar a qualidade de seus produtos, bem como por consumidores para tomar decisões mais informadas sobre quais vinhos escolher.

Além disso, o modelo poderia ajudar a expandir o conhecimento sobre a produção de vinho, auxiliar no aprendizado de novos enólogos e auxiliar nas avaliações feitas pelos sommeliers que, por se basearem em experiências, estão propensos a fatores subjetivos.

Dessa forma, seria extremamente plausível a utilização de modelos de Machine Learning para auxílio na produção e classificação de vinhos, como ficou demonstrado neste projeto.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

Gustavo Wohlers		
Karine Alves		
Luiz Fonseca		
Maísa Santos		
Pablo Brito		

CORTEZ, P.; CERDEIRA, A.; ALMEIDA, F.; MATOS, T.; REIS, J. **Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, v. 47 (4), p. 547-553, 2009. DOI: 10.1016/j.dss.2009.05.016.**

Kaggle. **Wine Quality Dataset.** Disponível em: <https://www.kaggle.com/datasets/yasserh/wine-quality-dataset> Acesso em: 30 março de 2023.

Slidego. **Oficina de linguagens de programação para iniciantes.** Disponível em: <https://slidesgo.com/pt/tema/oficina-de-linguagens-de-programacao-para-iniciantes#position-8&related-1&rs=detail-related>. Acesso em: 30 de março de 2023.