```
def estudo_de_caso():

    resultados = “Análise da aplicação de \
    Machine Learning em um dataset sobre \
    vinhos”

    return resultados


estudo_de_caso()
```

```
for i in range(5):
  print(integrantes[i])
```

Gustavo Wohlers
Karine Alves
Luiz Fonseca
Maísa Santos
Pablo Brito

Sumário {

- Objetivo do trabalho
- Exploração do dataset
- Aplicação do estimador Base Line
- Resultados Preliminares
- Resultados Finais
- Conclusões
- Referências Bibliográficas

}

{

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

}

Objetivo do trabalho

Quais são as features presentes no dataset?

```
1 df_winewhite.head()
```

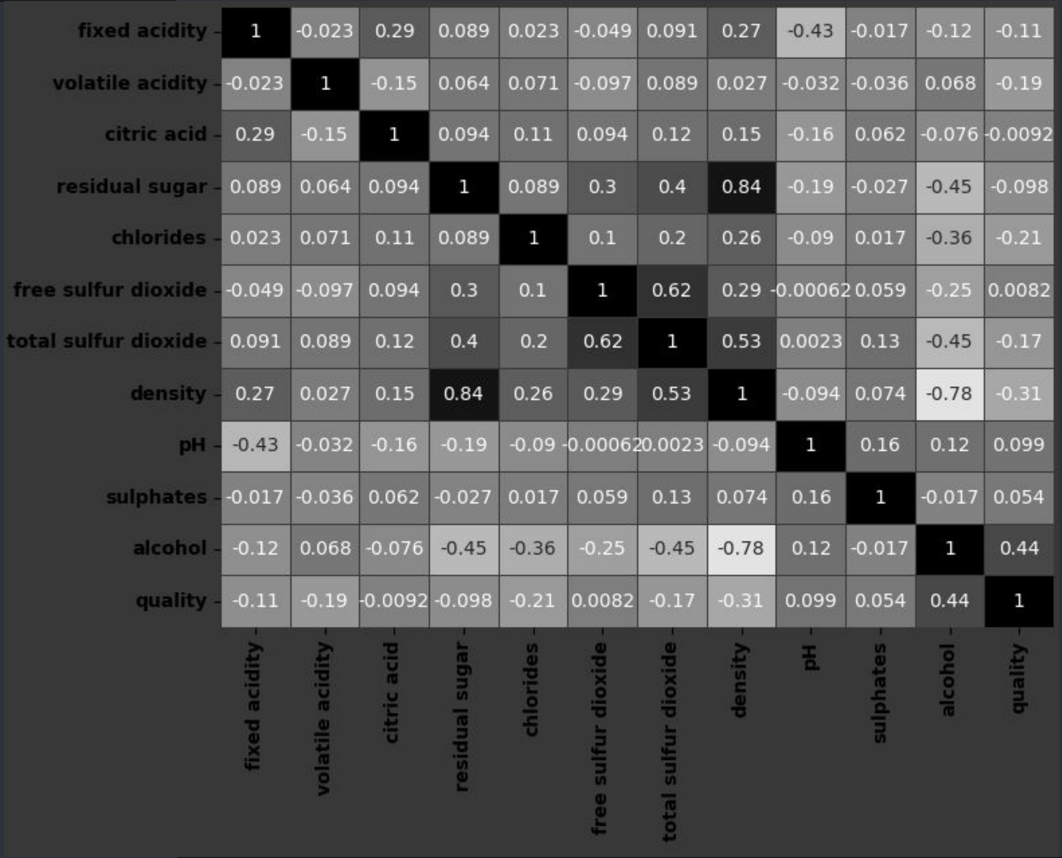
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

Existem valores Null ou NaN no dataset?

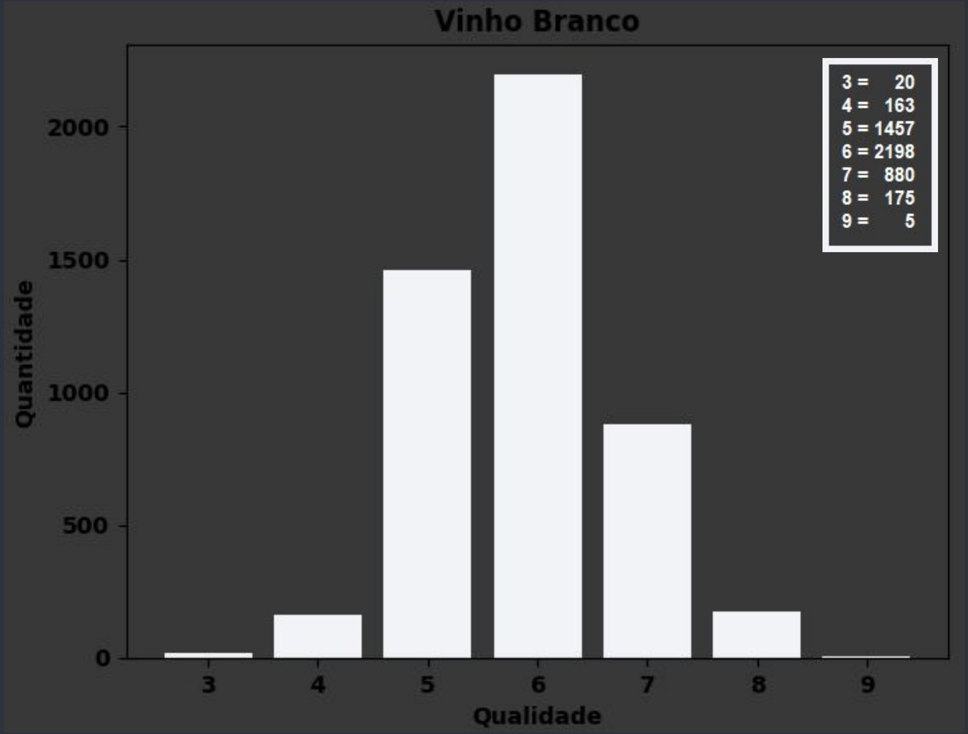
```
1 df_winewhite.info()
```

```
<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
```

Qual a matriz de correlação do dataset?



O quão desbalanceado está o target do dataset?



E em relação aos outliers?

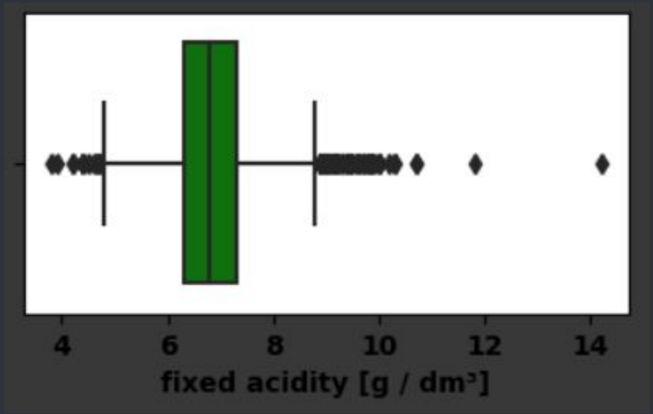
```
1 df_winewhite.describe()
```

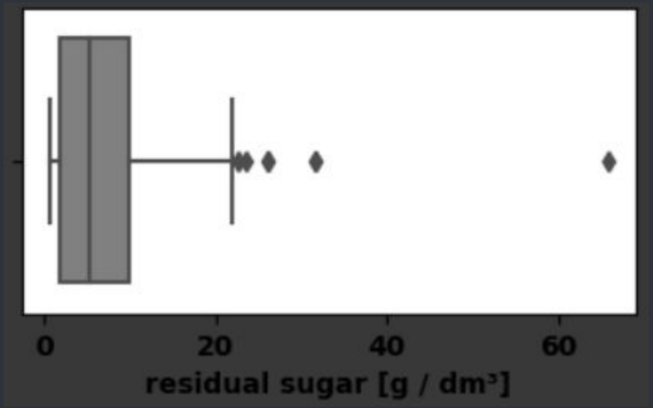
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267	5.877909
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621	0.885639
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000	3.000000
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000	5.000000
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000	6.000000
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000	6.000000
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000	9.000000

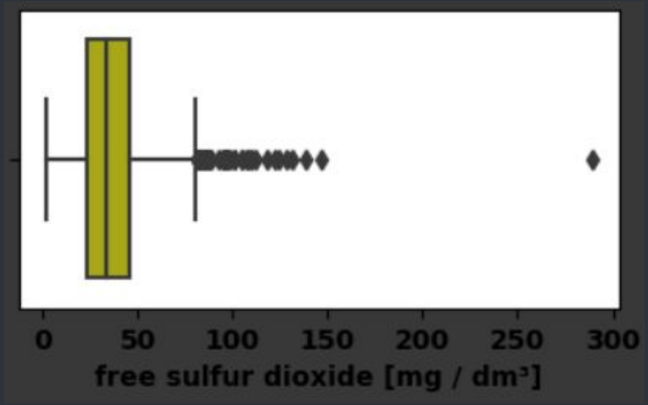
E em relação aos outliers?

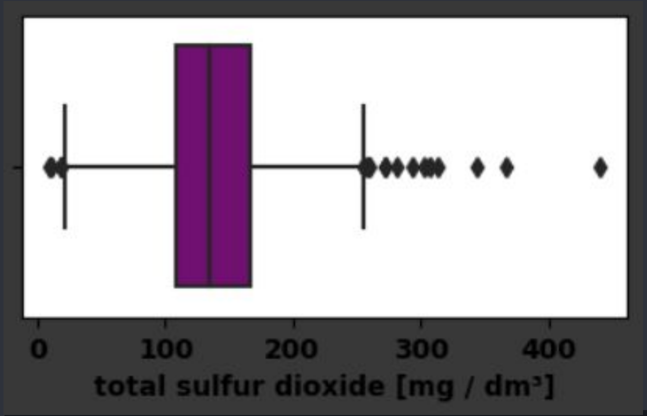
```
df_winewhite.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267	5.877909
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621	0.885639
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000	3.000000
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000	5.000000
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000	6.000000
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000	6.000000
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000	9.000000


4 6 8 10 12 14
fixed acidity [g / dm³]

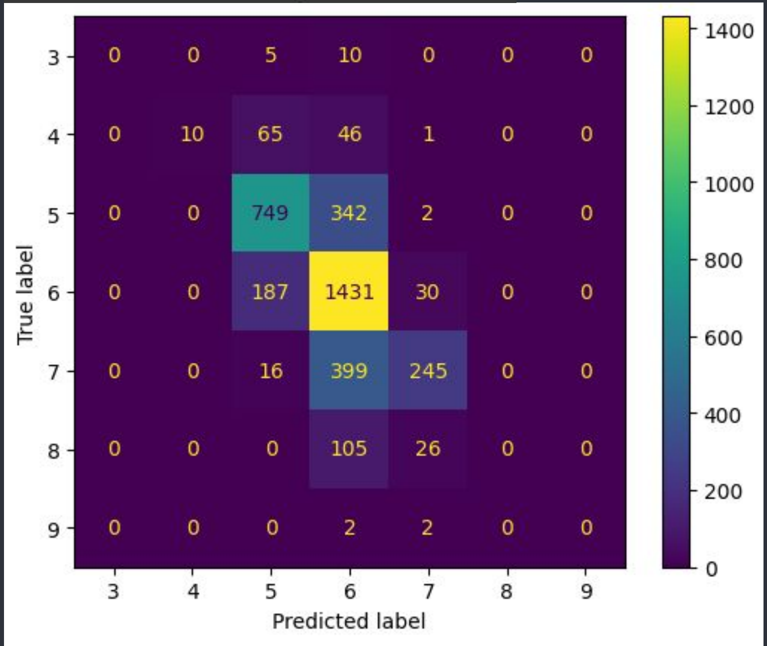

0 20 40 60
residual sugar [g / dm³]


0 50 100 150 200 250 300
free sulfur dioxide [mg / dm³]

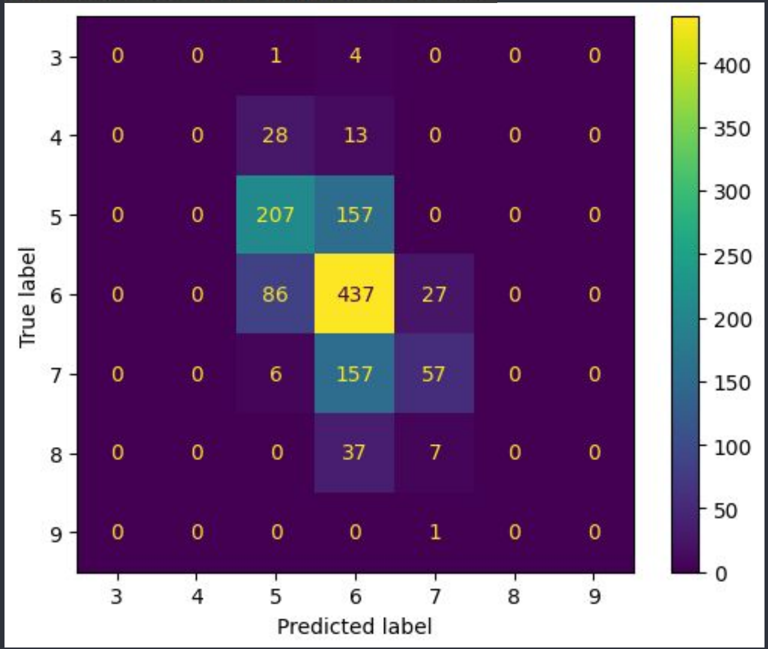

0 100 200 300 400
total sulfur dioxide [mg / dm³]

Random Forest - Base Line

Métricas de TREINO



Métricas de TESTE



Aplicação do estimador Base Line

Random Forest - Base Line

```
Random best parameters: {'rf__n_estimators': 180, 'rf__max_features': 2, 'rf__max_depth': 7}
Random best Score: 0.5818112486672196
```

Métricas de TREINO

	precision	recall	f1-score	support
3	0.00	0.00	0.00	15
4	1.00	0.08	0.15	122
5	0.73	0.69	0.71	1093
6	0.61	0.87	0.72	1648
7	0.80	0.37	0.51	660
8	0.00	0.00	0.00	131
9	0.00	0.00	0.00	4
accuracy			0.66	3673
macro avg	0.45	0.29	0.30	3673
weighted avg	0.67	0.66	0.63	3673

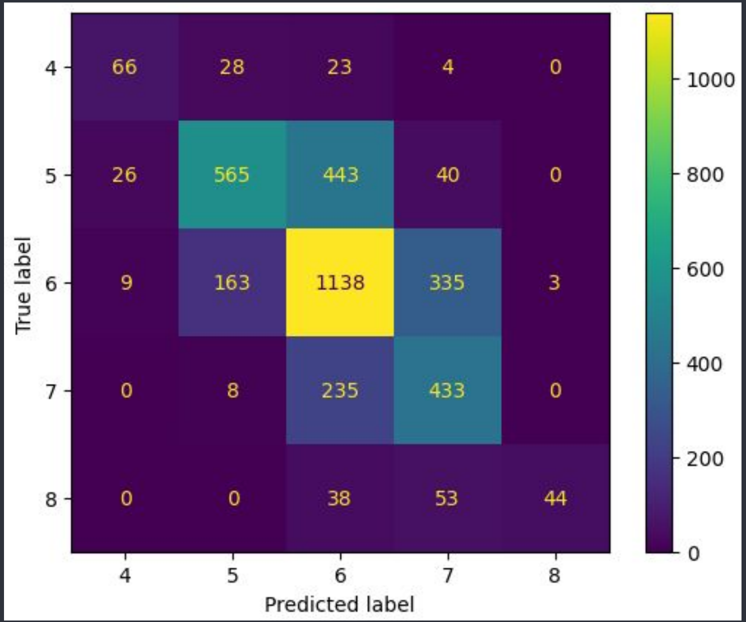
Métricas de TESTE

	precision	recall	f1-score	support
3	0.00	0.00	0.00	5
4	0.00	0.00	0.00	41
5	0.63	0.57	0.60	364
6	0.54	0.79	0.65	550
7	0.62	0.26	0.37	220
8	0.00	0.00	0.00	44
9	0.00	0.00	0.00	1
accuracy			0.57	1225
macro avg	0.26	0.23	0.23	1225
weighted avg	0.54	0.57	0.53	1225

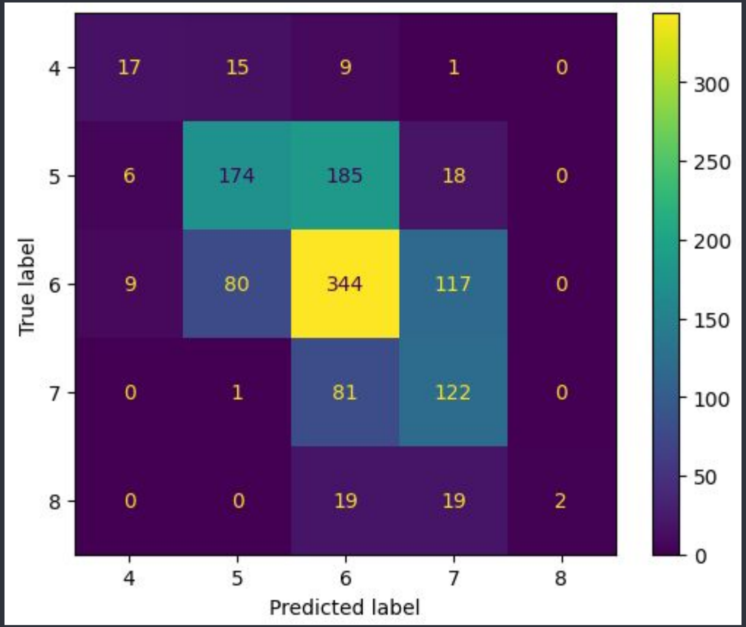
Aplicação do estimador Base Line

Random Forest - SMOTE/STRATIFIED KFOLD

Métricas de TREINO



Métricas de TESTE



Random Forest - SMOTE/STRATIFIED KFOLD

```
Grid best parameters: {'rf__n_estimators': 165, 'rf__max_features': 2, 'rf__max_depth': 7}
Grid best Score: 0.5327800829875518
```

Métricas de TREINO

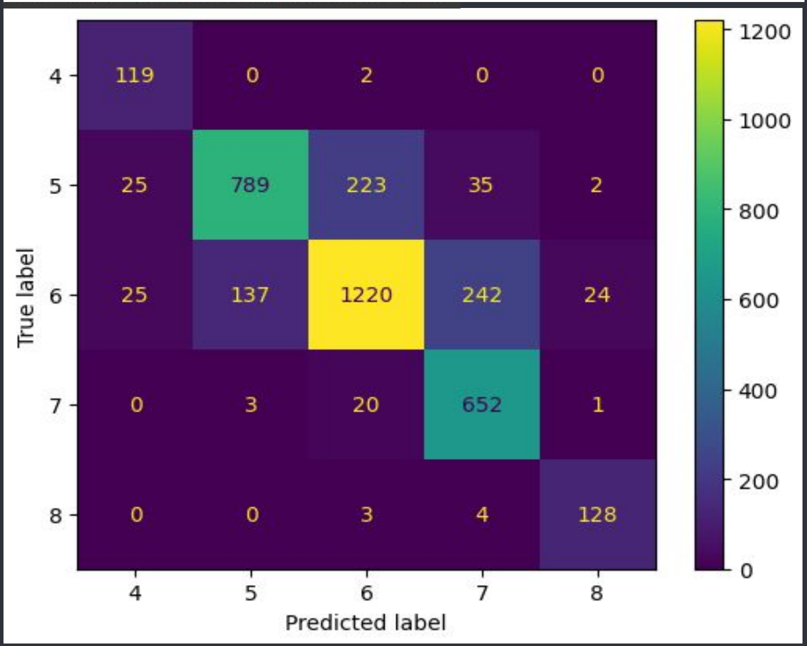
	precision	recall	f1-score	support
4	0.65	0.55	0.59	121
5	0.74	0.53	0.61	1074
6	0.61	0.69	0.65	1648
7	0.50	0.64	0.56	676
8	0.94	0.33	0.48	135
accuracy			$\boxed{0.61}$	3654
macro avg	0.69	0.55	0.58	3654
weighted avg	0.64	0.61	0.61	3654

Métricas de TESTE

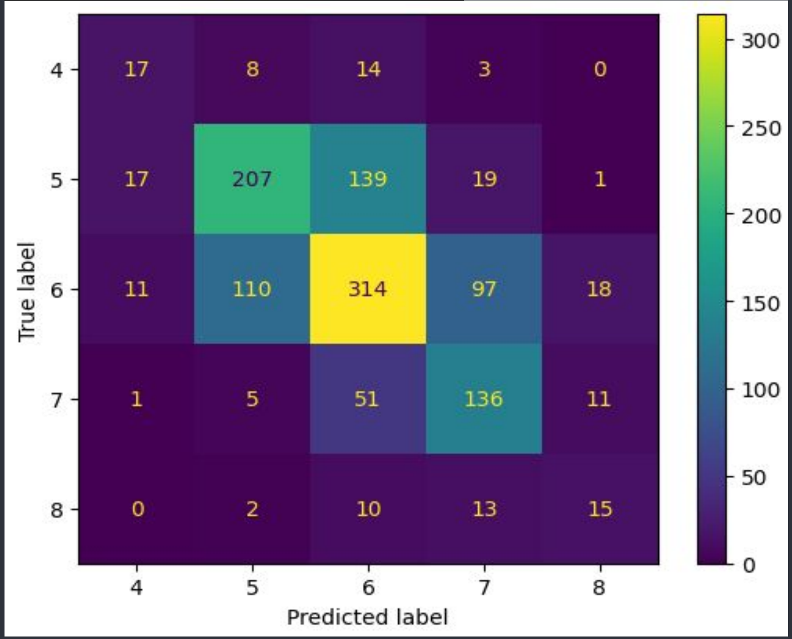
	precision	recall	f1-score	support
4	0.53	0.40	0.46	42
5	0.64	0.45	0.53	383
6	0.54	0.63	0.58	550
7	0.44	0.60	0.51	204
8	1.00	0.05	0.10	40
accuracy			$\boxed{0.54}$	1219
macro avg	0.63	0.43	0.43	1219
weighted avg	0.57	0.54	0.53	1219

SVC (Support Vector Classifier)

Métricas de TREINO



Métricas de TESTE



SVC (Support Vector Classifier)

```
Grid best parameters: {'svc__gamma': 0.5000000000000001, 'svc__C': 4}
Grid best Score: 0.5618257261410788
```

Métricas de TREINO

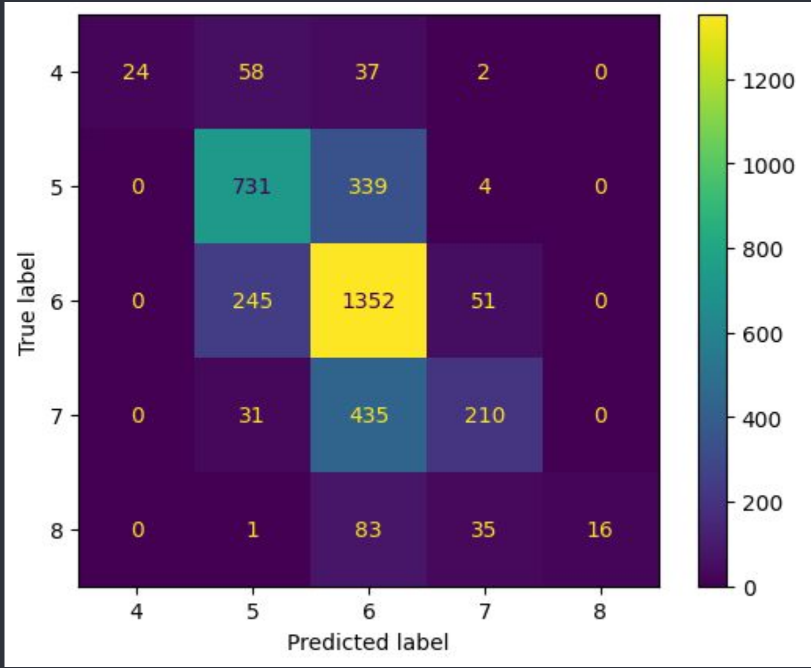
	precision	recall	f1-score	support
4	0.70	0.98	0.82	121
5	0.85	0.73	0.79	1074
6	0.83	0.74	0.78	1648
7	0.70	0.96	0.81	676
8	0.83	0.95	0.88	135
accuracy			$\boxed{0.80}$	3654
macro avg	0.78	0.87	0.82	3654
weighted avg	0.81	0.80	0.79	3654

Métricas de TESTE

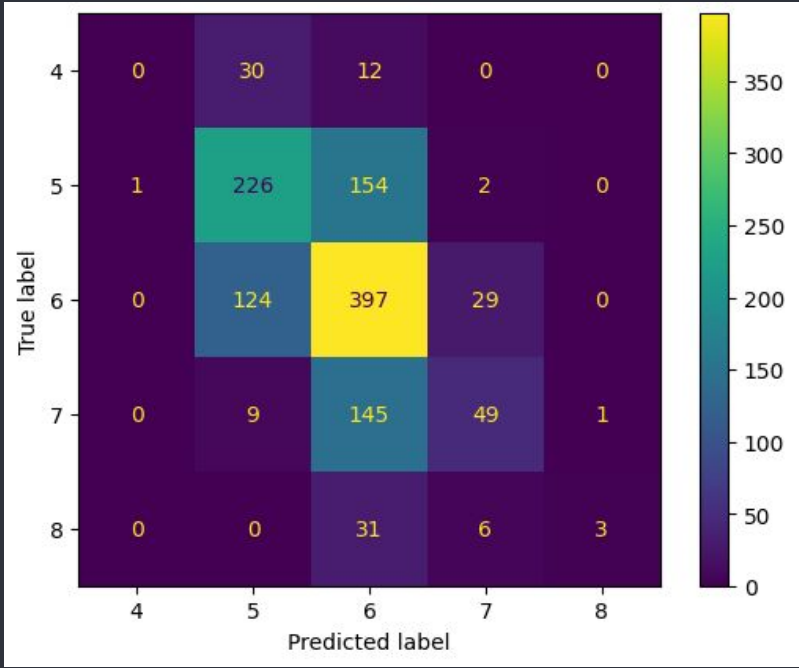
	precision	recall	f1-score	support
4	0.37	0.40	0.39	42
5	0.62	0.54	0.58	383
6	0.59	0.57	0.58	550
7	0.51	0.67	0.58	204
8	0.33	0.38	0.35	40
accuracy			$\boxed{0.57}$	1219
macro avg	0.49	0.51	0.50	1219
weighted avg	0.57	0.57	0.57	1219

Random Forest - Feature Selection

Métricas de TREINO



Métricas de TESTE



Random Forest - Feature Selection

```
Grid best parameters: {'rf__n_estimators': 180, 'rf__max_features': 2, 'rf__max_depth': 7}
Grid best Score: 0.5673291414027997
```

Métricas de TREINO

	precision	recall	f1-score	support
4	1.00	0.20	0.33	121
5	0.69	0.68	0.68	1074
6	0.60	0.82	0.69	1648
7	0.70	0.31	0.43	676
8	1.00	0.12	0.21	135
accuracy			0.64	3654
macro avg	0.80	0.43	0.47	3654
weighted avg	0.67	0.64	0.61	3654

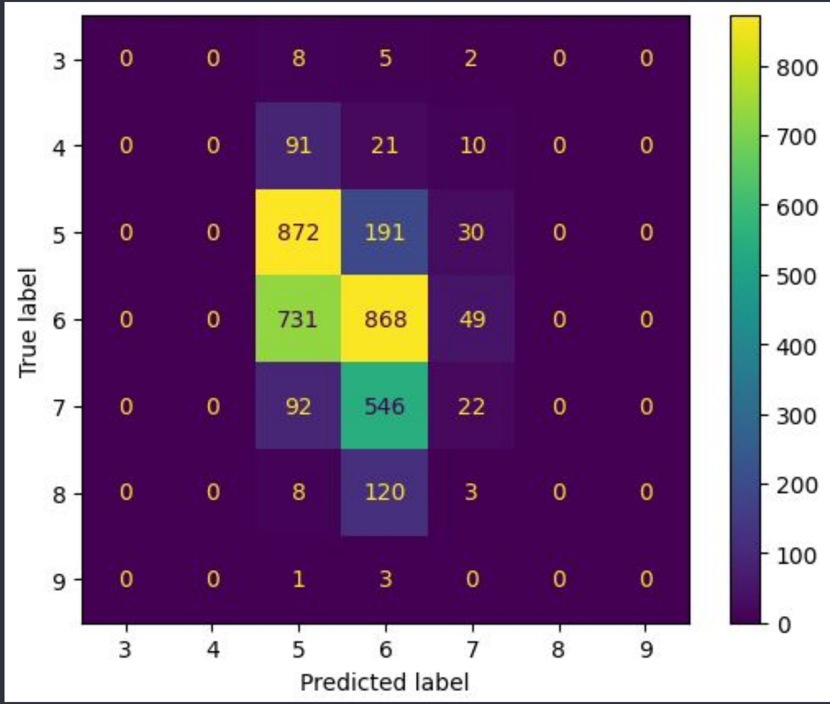
Métricas de TESTE

	precision	recall	f1-score	support
4	0.00	0.00	0.00	42
5	0.58	0.59	0.59	383
6	0.54	0.72	0.62	550
7	0.57	0.24	0.34	204
8	0.75	0.07	0.14	40
accuracy			0.55	1219
macro avg	0.49	0.33	0.34	1219
weighted avg	0.54	0.55	0.52	1219

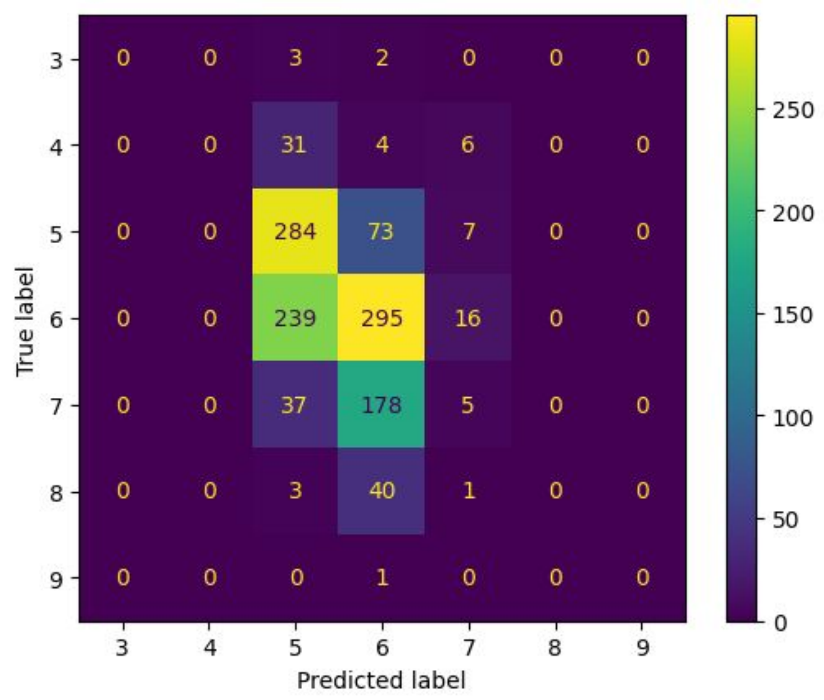
```
Index(['volatile acidity', 'free sulfur dioxide', 'density', 'alcohol'], dtype='object')
```

AdaBoost

Métricas de TREINO



Métricas de TESTE



AdaBoost

```
Grid best parameters: {'adb__n_estimators': 9, 'adb__learning_rate': 0.97, 'adb__algorithm': 'SAMME'}
Grid best Score: 0.4816179954981637
```

Métricas de TREINO

	precision	recall	f1-score	support
3	0.00	0.00	0.00	15
4	0.00	0.00	0.00	122
5	0.48	0.80	0.60	1093
6	0.49	0.53	0.51	1648
7	0.19	0.03	0.06	660
8	0.00	0.00	0.00	131
9	0.00	0.00	0.00	4
accuracy			0.48	3673
macro avg	0.17	0.19	0.17	3673
weighted avg	0.40	0.48	0.42	3673

Métricas de TESTE

	precision	recall	f1-score	support
3	0.00	0.00	0.00	5
4	0.00	0.00	0.00	41
5	0.48	0.78	0.59	364
6	0.50	0.54	0.52	550
7	0.14	0.02	0.04	220
8	0.00	0.00	0.00	44
9	0.00	0.00	0.00	1
accuracy			0.48	1225
macro avg	0.16	0.19	0.16	1225
weighted avg	0.39	0.48	0.41	1225

Nova abordagem {

Após o teste de diferentes técnicas, ficou constatada a baixa acurácia para os diferentes modelos. Como possível solução, o problema foi dividido em duas etapas:

- Classificar o vinho como bom ou ruim;
- Diferenciar a classificação do vinho bom.

}

Novos Algoritmos {

O primeiro algoritmo irá classificar entre vinho bom e vinho ruim:

```
qualidade_vinho = {3:'bad', 4: 'bad', 5: 'bad', 6: 'bad', 7: 'good', \
8: 'good', 9:'good'}
```

O segundo algoritmo irá diferenciar a classe dos vinhos bons:

```
df_winewhite_good = df_winewhite[df_winewhite ['quality'].isin([7, 8])]
```

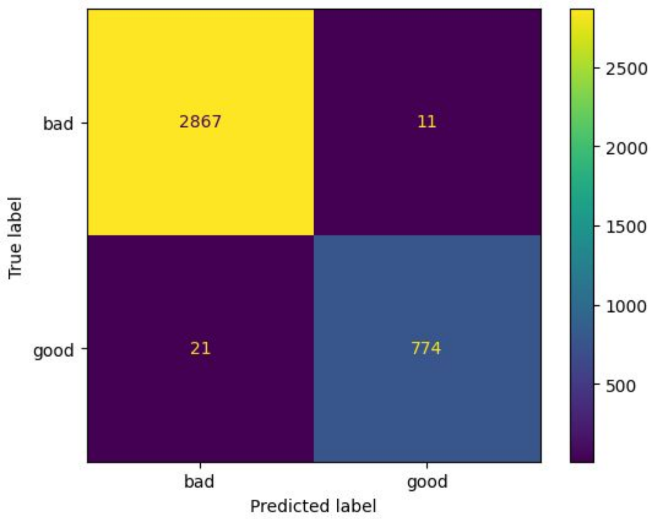
<"Obs.: nota 9 não utilizada, pois apresenta poucas amostras" >

}

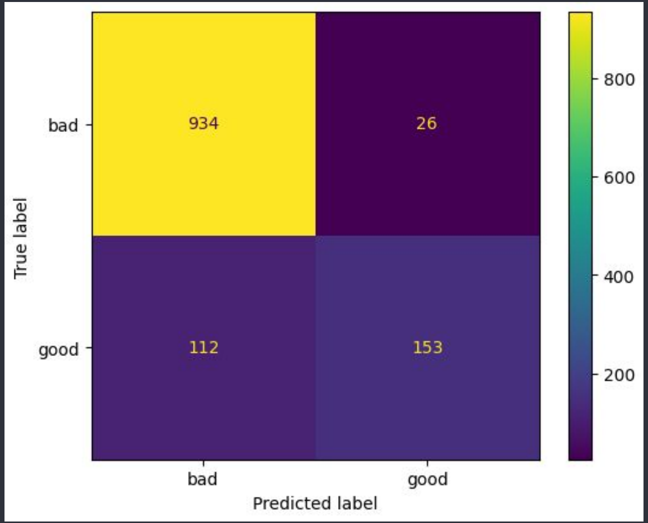
SVC Classificando Vinhos bons x Vinhos ruins

```
Grid best parameters: {'svc__gamma': 0.9000000000000001, 'svc__C': 2}
Grid best Score: 0.8559797417367611
```

Métricas de TREINO



Métricas de TESTE



SVC Classificando Vinhos bons x Vinhos Ruins

Métricas de TREINO

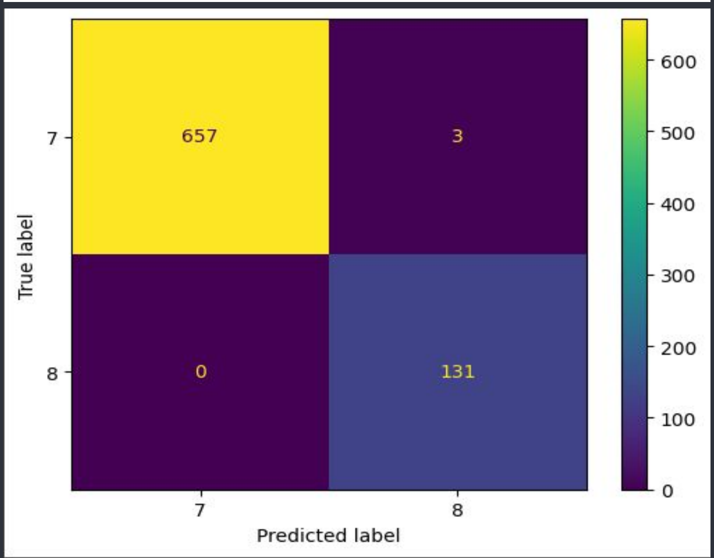
	precision	recall	f1-score	support
bad	0.99	1.00	0.99	2878
good	0.99	0.97	0.98	795
accuracy			$\boxed{0.99}$	3673
macro avg	0.99	0.98	0.99	3673
weighted avg	0.99	0.99	0.99	3673

Métricas de TESTE

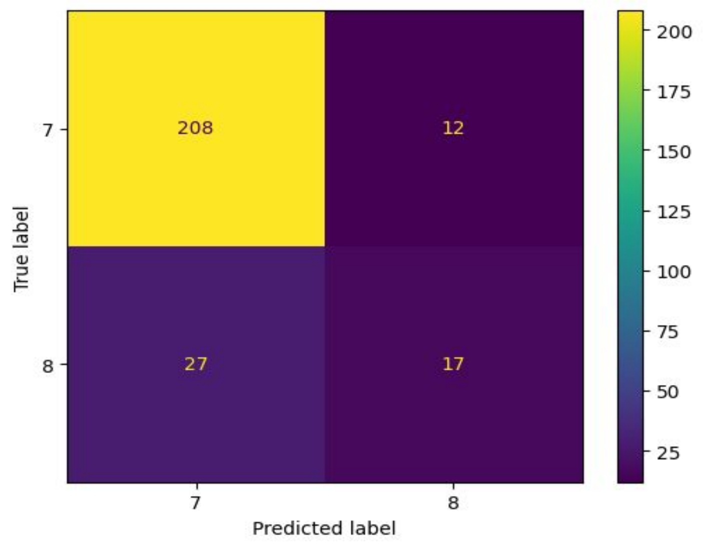
	precision	recall	f1-score	support
bad	0.89	0.97	0.93	960
good	0.85	0.58	0.69	265
accuracy			$\boxed{0.89}$	1225
macro avg	0.87	0.78	0.81	1225
weighted avg	0.88	0.89	0.88	1225

SVC para Vinhos de nota 7 x Vinhos nota 8

Métricas de TREINO



Métricas de TESTE



SVC para Vinhos de nota 7 x Vinhos nota 8

```
Grid best parameters: {'rf__n_estimators': 180, 'rf__max_features': 2, 'rf__max_depth': 7}
Grid best Score: 0.8282134818149508
```

Métricas de TREINO

	precision	recall	f1-score	support
7	1.00	1.00	1.00	660
8	0.98	1.00	0.99	131
accuracy			$\boxed{1.00}$	791
macro avg	0.99	1.00	0.99	791
weighted avg	1.00	1.00	1.00	791

Métricas de TESTE

	precision	recall	f1-score	support
7	0.89	0.95	0.91	220
8	0.59	0.39	0.47	44
accuracy			$\boxed{0.85}$	264
macro avg	0.74	0.67	0.69	264
weighted avg	0.84	0.85	0.84	264

Por fim, o modelo iria para produção?

Sim! Os resultados do projeto podem ser usados 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.

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.