

```
In [ ]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

Vamos analisar vinhos!

Usaremos um dataset composto por propriedades físico-químicas de vinhos brancos. Temos 1599 amostras e um total de 11 variáveis independentes, descritas abaixo:

- `fixed acidity` : a maioria dos ácidos envolvidos com vinho (não evaporam prontamente)
- `volatile acidity` : a quantidade de ácido acético no vinho, que em níveis muito altos pode levar a um gosto desagradável de vinagre
- `citric acid` : encontrado em pequenas quantidades, o ácido cítrico pode adicionar "leveza" e sabor aos vinhos
- `residual sugar` : a quantidade de açúcar restante após a fermentação é interrompida, é raro encontrar vinhos com menos de 1 grama / litro e vinhos com mais de 45 gramas / litro são considerados doces
- `chlorides` : a quantidade de sal no vinho

`free sulfur dioxide`: a forma livre de SO₂ existe em equilíbrio entre o SO₂ molecular (como gás dissolvido) e o íon bisulfito; impede o crescimento microbiano e a oxidação do vinho

- `total sulfur dioxide` : Quantidade de formas livres e encadernadas de SO₂; em baixas concentrações, o SO₂ é quase indetectável no vinho, mas nas concentrações de SO₂ acima de 50 ppm, o SO₂ se torna evidente no nariz e no sabor do vinho.
- `density` : a densidade do vinho é próxima a da água, dependendo do percentual de álcool e teor de açúcar
- `pH` : descreve se o vinho é ácido ou básico numa escala de 0 (muito ácido) a 14 (muito básico); a maioria dos vinhos está entre 3-4 na escala de pH
- `sulphates` : um aditivo de vinho que pode contribuir para os níveis de gás de dióxido de enxofre (SO₂), que age como um antimicrobiano e antioxidante
- `alcohol` : o percentual de álcool no vinho

Existe ainda uma variável chamada `quality`. Essa variável é uma nota de qualidade do vinho que varia de 0 a 10.

Nesse caso, todas as colunas, exceto a coluna "quality", são consideradas variáveis contínuas, pois representam medidas numéricas contínuas. A coluna "quality" é considerada uma variável categórica, pois representa as categorias de qualidade do vinho (pontuações de 0 a 10).

```
In [ ]: wine = pd.read_csv('E:\INFINET\PROJETO\INFINET\logisticregression\winequalityN.csv')
```

In []: wine

Out[]:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH
0	white	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00
1	white	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30
2	white	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26
3	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19
4	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19
...
6492	red	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45
6493	red	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52
6494	red	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42
6495	red	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57
6496	red	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39

6497 rows × 13 columns

In []: wine.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   type             6497 non-null   object 
 1   fixed acidity    6487 non-null   float64
 2   volatile acidity 6489 non-null   float64
 3   citric acid      6494 non-null   float64
 4   residual sugar   6495 non-null   float64
 5   chlorides        6495 non-null   float64
 6   free sulfur dioxide 6497 non-null   float64
 7   total sulfur dioxide 6497 non-null   float64
 8   density          6497 non-null   float64
 9   pH               6488 non-null   float64
 10  sulphates        6493 non-null   float64
 11  alcohol          6497 non-null   float64
 12  quality          6497 non-null   int64  
dtypes: float64(11), int64(1), object(1)
memory usage: 660.0+ KB

```

In []: wine.isnull().sum().sort_values(ascending= False)

```
Out[ ]: fixed acidity      10
          pH                 9
          volatile acidity   8
          sulphates         4
          citric acid        3
          residual sugar     2
          chlorides          2
          type                0
          free sulfur dioxide 0
          total sulfur dioxide 0
          density              0
          alcohol              0
          quality              0
          dtype: int64
```

defini que todos os vinhos acima de 5 são bons , logo = 1 e os abaixo ou igual a 5 são ruins = 0

```
In [ ]: number_of_wines = wine.shape[0]
wine['category'] = np.zeros((number_of_wines, 1))
wine.loc[wine.quality > 5, "category"] = 1
```

```
In [ ]: wine
```

```
Out[ ]:
```

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH
0	white	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00
1	white	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30
2	white	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26
3	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19
4	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19
...
6492	red	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45
6493	red	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52
6494	red	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42
6495	red	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57
6496	red	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39

6497 rows × 14 columns

```
In [ ]: w_wine = wine['type'] == 'white'
r_wine = wine['type'] == 'red'
```

```
In [ ]: fig, axes = plt.subplots(1, 3, figsize=(12, 4))

fig.suptitle("Qualidade dos Vinhos")
axes[0].set_title("Qualidade de todos vinhos")
```

```

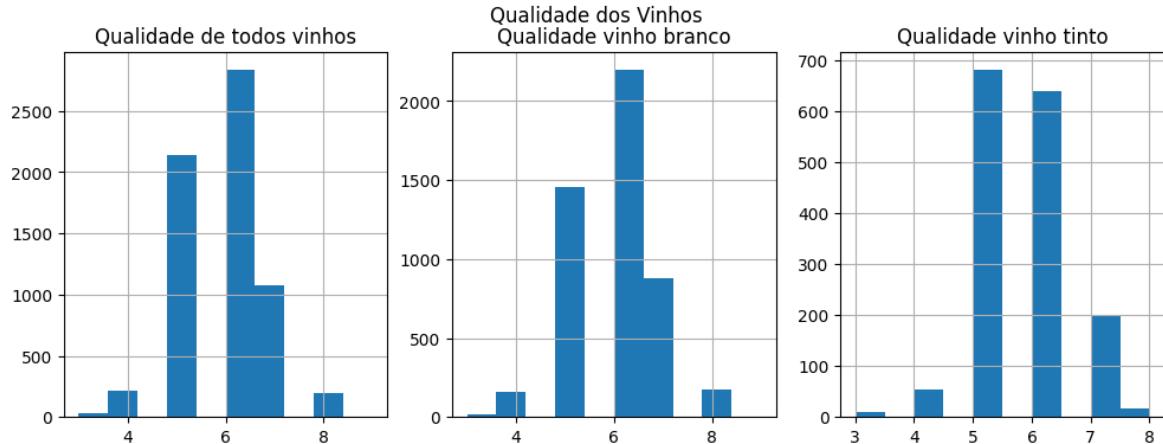
axes[1].set_title("Qualidade vinho branco")
axes[2].set_title("Qualidade vinho tinto")
wine.quality.hist(ax=axes[0])

wine[w_wine].quality.hist(ax=axes[1])

wine[r_wine].quality.hist(ax=axes[2])

```

Out[]: <Axes: title={'center': 'Qualidade vinho tinto'}>



In []: df_w_wine = wine[w_wine]
df_r_wine = wine[r_wine]

In []: df_w_wine.isnull().sum().sort_values(ascending=False)

Out[]: fixed acidity 8
volatile acidity 7
pH 7
citric acid 2
residual sugar 2
chlorides 2
sulphates 2
type 0
free sulfur dioxide 0
total sulfur dioxide 0
density 0
alcohol 0
quality 0
category 0
dtype: int64

In []: df_r_wine.isnull().sum().sort_values(ascending=False)

```
Out[ ]: fixed acidity      2
          pH                 2
          sulphates         2
          volatile acidity   1
          citric acid        1
          type                0
          residual sugar      0
          chlorides           0
          free sulfur dioxide 0
          total sulfur dioxide 0
          density              0
          alcohol              0
          quality              0
          category             0
          dtype: int64
```

para poder alterar de forma simplificada o gráfico da correlação, irei criar uma variável com os nomes das colunas

```
In [ ]: vars = [
          'fixed acidity',
          'volatile acidity',
          'citric acid',
          'residual sugar',
          'chlorides',
          'free sulfur dioxide',
          'total sulfur dioxide',
          'density',
          'pH',
          'sulphates',
          'alcohol',
          'category'
      ]
```

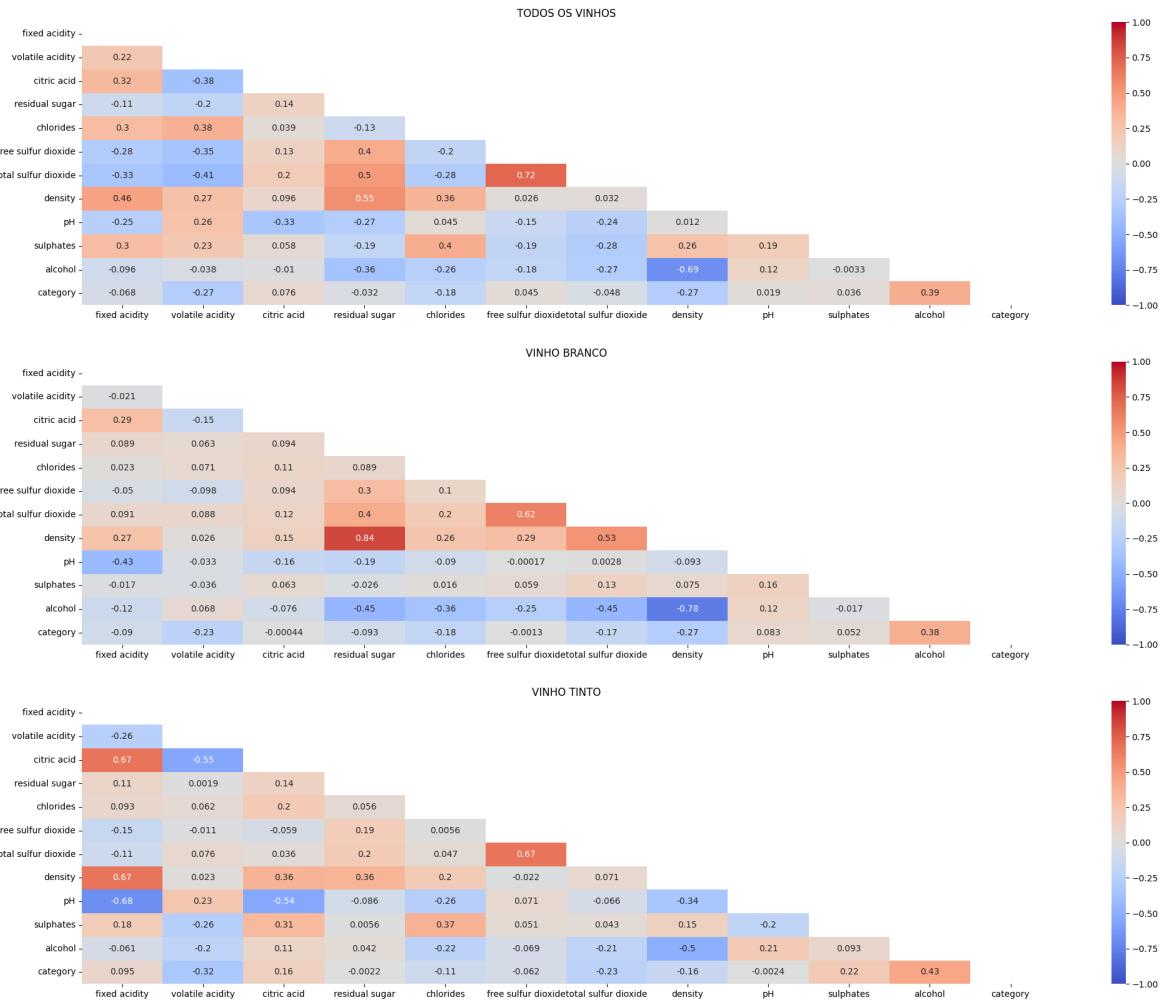
Vou criar uma máscara para plotar somente a parte inferior do heatmap e para facilitar e não ter que repetir código, irei fazer uma função mask com a função mask vou subplotar as correlações de todo dataframe, somente branco e somente tinto

```
In [ ]: def mask(DF):
          mask = np.triu(np.ones_like(DF, dtype=bool))
          return mask
```

```
In [ ]: fig, axes = plt.subplots(3, 1, figsize=(25, 20))
          sns.heatmap(wine[vars].corr(), vmax=1, vmin=-1, annot=True, ax=axes[0], cmap="copper")
          sns.heatmap(df_w_wine[vars].corr(), vmax=1, vmin=-1, annot=True, ax=axes[1], cmap="copper")
          sns.heatmap(df_r_wine[vars].corr(), vmax=1, vmin=-1, annot=True, ax=axes[2], cmap="copper")

          axes[0].set_title("TODOS OS VINHOS")
          axes[1].set_title("VINHO BRANCO")
          axes[2].set_title("VINHO TINTO")
```

```
Out[ ]: Text(0.5, 1.0, 'VINHO TINTO')
```



Com os graficos acima, ficou claro que nenhuma das colunas tem uma correlação forte com a minha categoria Logo os valores nulos não irão influenciar de forma drástica

```
In [ ]: wine_t = wine.fillna(wine.mean())
```

```
C:\Users\Diones\AppData\Local\Temp\ipykernel_2972\691893756.py:1: FutureWarning:
The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
```

```
wine_t = wine.fillna(wine.mean())
```

```
In [ ]: df_tr_wine= df_r_wine.fillna(df_r_wine.mean())
```

```
C:\Users\Diones\AppData\Local\Temp\ipykernel_2972\2938092119.py:1: FutureWarning:
The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
```

```
df_tr_wine= df_r_wine.fillna(df_r_wine.mean())
```

```
In [ ]: df_tw_wine = df_w_wine.fillna(df_w_wine.mean())
```

```
C:\Users\Diones\AppData\Local\Temp\ipykernel_2972\1451453130.py:1: FutureWarning:
The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
```

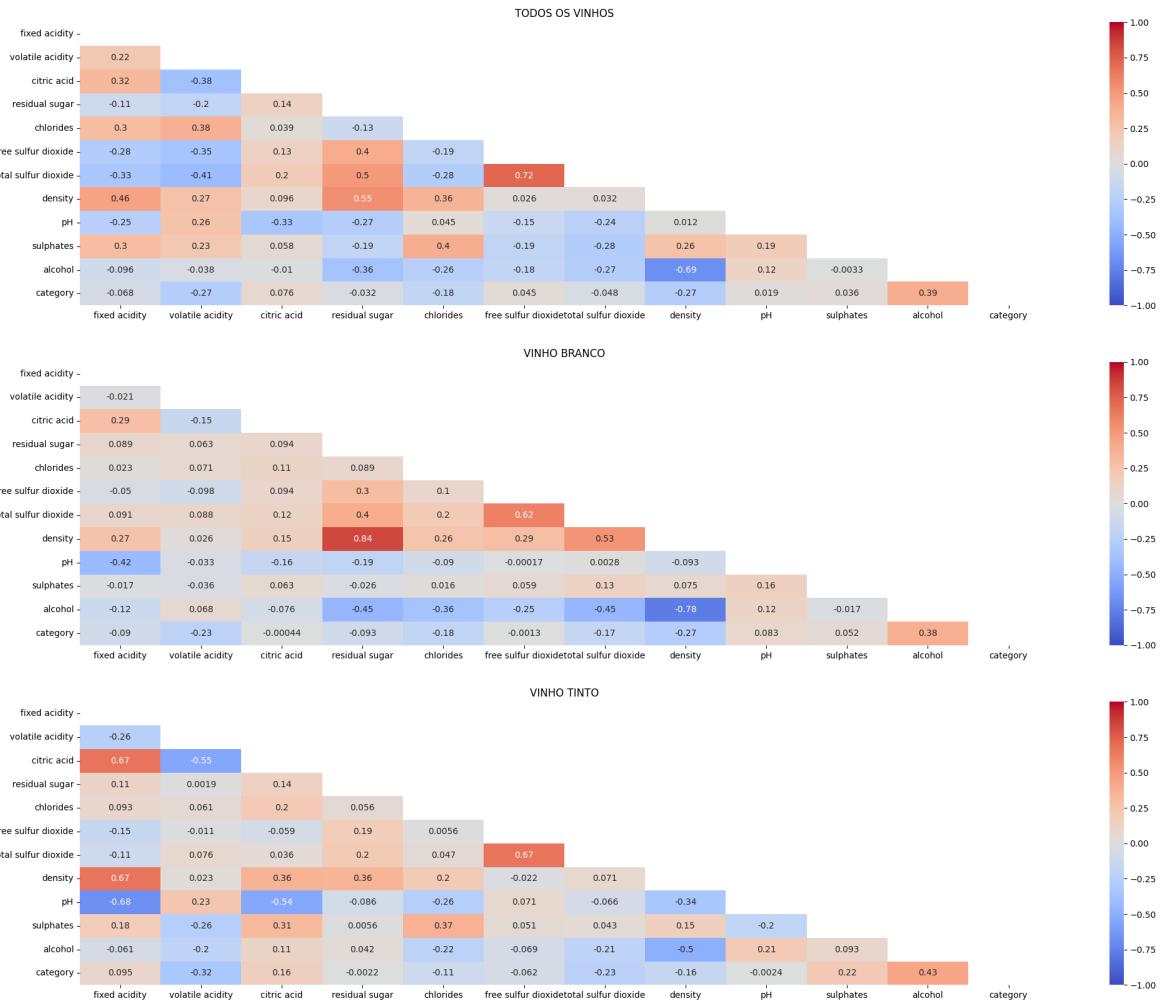
```
df_tw_wine = df_w_wine.fillna(df_w_wine.mean())
```

agora gerando a correlação com os datasets tratados para comparar e conferir se algo mudou nas correlações

```
In [ ]: fig, axes = plt.subplots(3, 1, figsize=(25, 20))
sns.heatmap(wine_t[vars].corr(), vmax=1, vmin=-1, annot=True, ax=axes[0], cmap="RdYlBu")
sns.heatmap(df_tw_wine[vars].corr(), vmax=1, vmin=-1, annot=True, ax=axes[1], cmap="RdYlBu")
sns.heatmap(df_tr_wine[vars].corr(), vmax=1, vmin=-1, annot=True, ax=axes[2], cmap="RdYlBu")

axes[0].set_title("TODOS OS VINHOS")
axes[1].set_title("VINHO BRANCO")
axes[2].set_title("VINHO TINTO")
```

Out[]: Text(0.5, 1.0, 'VINHO TINTO')



como pode ser visto , mantiveram as correlações, assim como usarei os dados de regressão logistica para classificar os vinhos, esses valores médios encontrados não causarão muitos desvios.

In []: #fazendo um pairplot

```
vars = [
    'fixed acidity',
    'volatile acidity',
    'citric acid',
    'residual sugar',
    'chlorides',
    'free sulfur dioxide',
    'total sulfur dioxide',
```

```

'density',
'pH',
'sulphates',
'alcohol',
'quality',
'category'
]
sns.pairplot(wine_t[vars], hue = 'category', height= 2, kind="reg", )

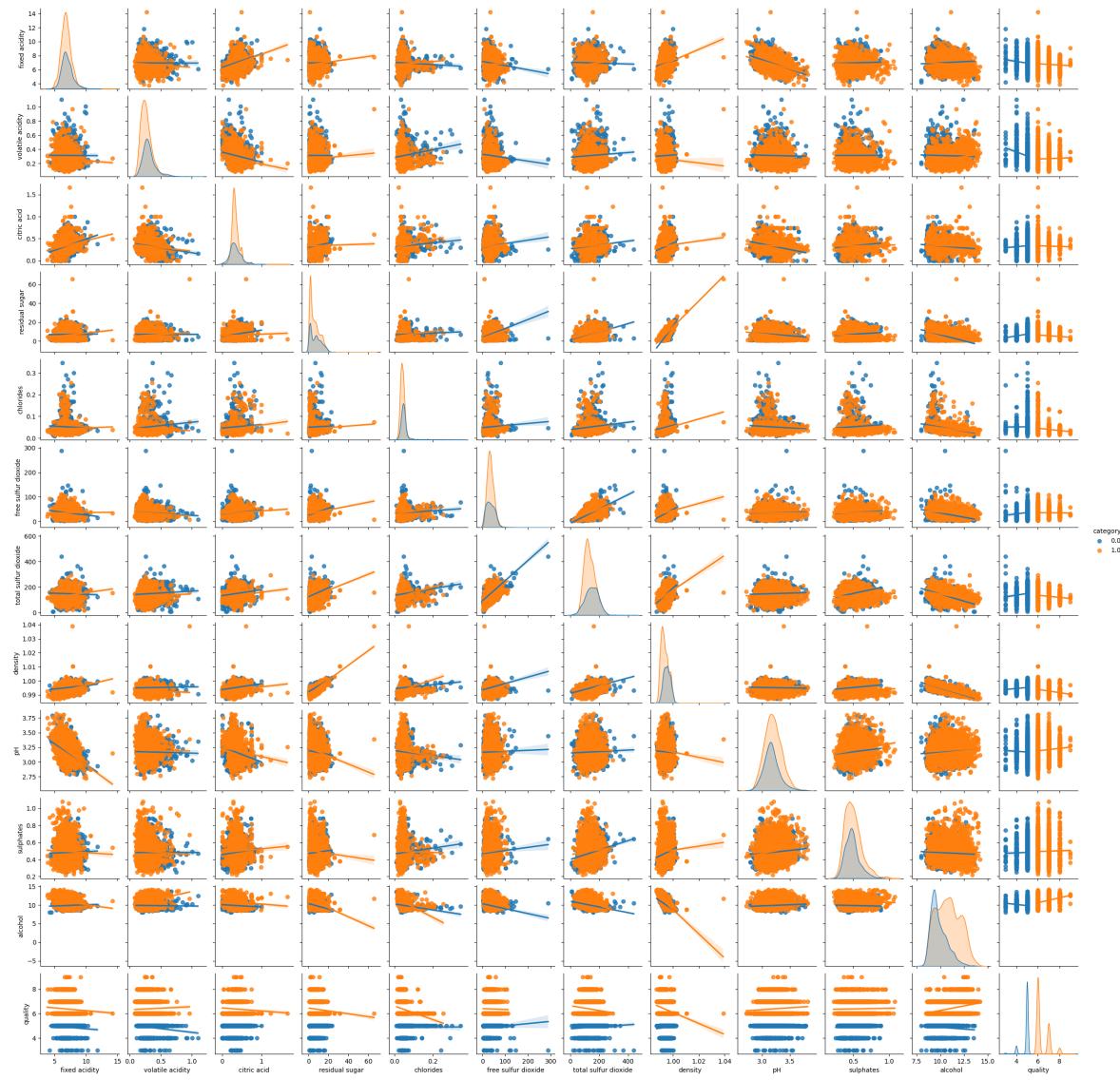
```

Out[]: <seaborn.axisgrid.PairGrid at 0x2ba70aa7970>



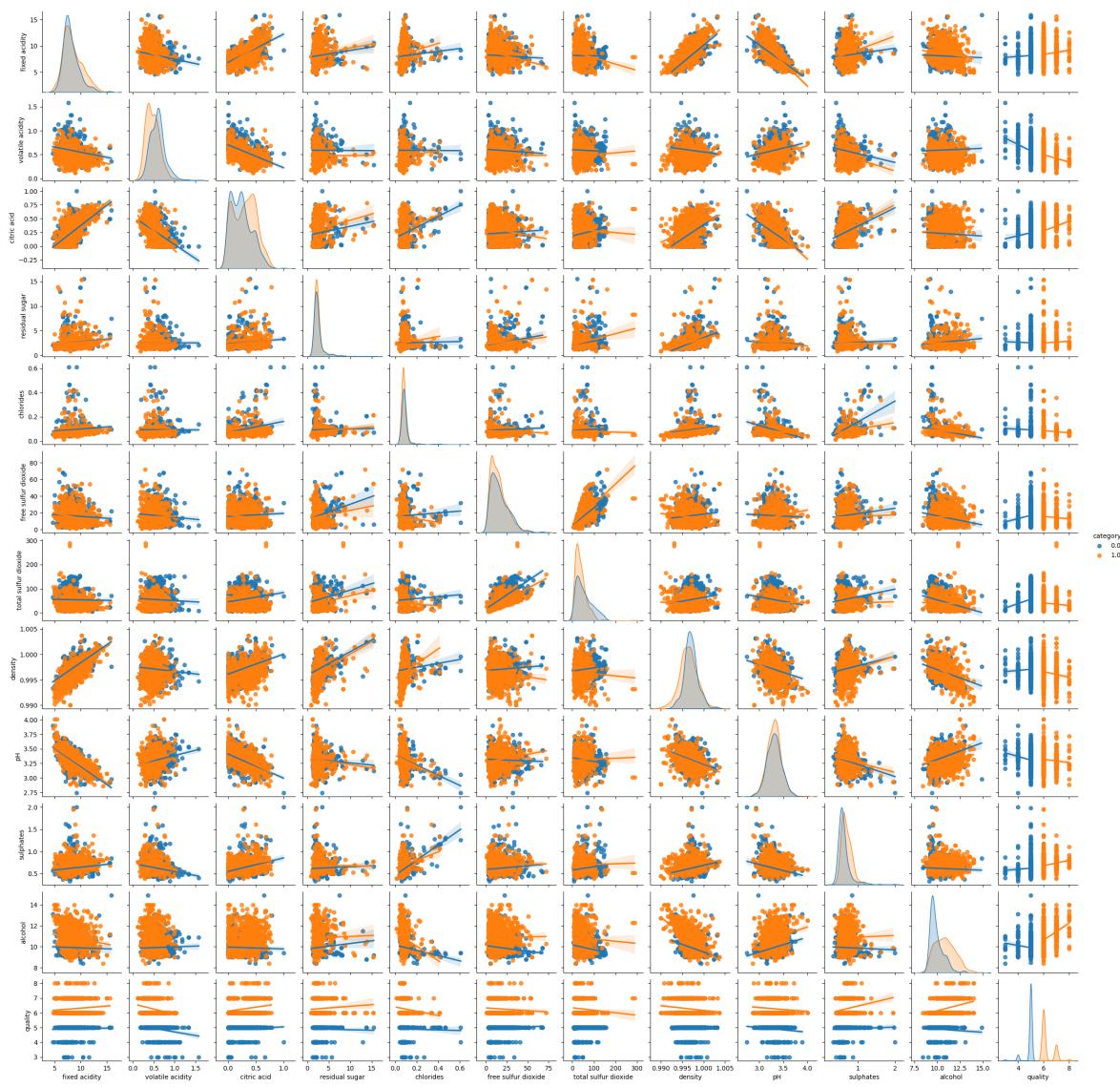
In []: sns.pairplot(df_tw_wine[vars], hue = 'category', height= 2, kind="reg",)

Out[]: <seaborn.axisgrid.PairGrid at 0x2ba7227cc70>



```
In [ ]: sns.pairplot(df_tr_wine[vars], hue = 'category', height= 2, kind="reg", )
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x2ba06fb93f0>
```



Utilizando a base de vinho branco vou utilizar para fazer o modelo de validação cruzada k-folds com $k = 10$ na regressão logistica.

In []: #reimportando as bibliotecas necessárias, algumas ja temos, outras ainda não.

```
# Import necessary packages
import pandas as pd
import random
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from copy import deepcopy as cp
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

from sklearn.tree import (
    DecisionTreeClassifier,
    plot_tree
)

from sklearn.model_selection import (
    train_test_split,
    StratifiedKFold)
from sklearn.metrics import (
    accuracy_score,
```

```

precision_score,
recall_score,
f1_score,
classification_report,
confusion_matrix,
roc_curve,
auc,
RocCurveDisplay
)
from sklearn.svm import SVC

```

```
In [ ]: df_tw_wine.rename(columns={'category':'opinion'}, inplace = True)
```

```
In [ ]: def specificity_score(y, y_pred):
cm = confusion_matrix(y, y_pred)
specificity = (cm[0,0] / (cm[0, 0] + cm[0 ,1]))
return specificity
```

```
In [ ]: def interpolation(fpr, tpr):
interp_fpr = np.linspace(0, 1, 100)
interp_tpr = np.interp(interp_fpr, fpr, tpr)
interp_tpr[0] = 0.
return interp_fpr, interp_tpr
```

```
In [ ]: var = [
'fixed acidity',
'volatile acidity',
'citric acid',
'residual sugar',
'chlorides',
'free sulfur dioxide',
'total sulfur dioxide',
'density',
'pH',
'sulphates',
'alcohol',
]

X = df_tw_wine[var]
y = df_tw_wine['opinion']
stratify = y
random_state = 42
test_size = 0.1
```

```
In [ ]: X_train_cv, X_test, y_train_cv, y_test = train_test_split(X.values,
y.values,
test_size = 0.1,
random_state = 42,
stratify = y)
```

```
def train(X, y, model_klass, model_kwargs = {}):
cv = StratifiedKFold(n_splits=10)
f1_score_val_list = []
f1_score_train_list = []
accuracy_score_val_list = []
```

```

accuracy_score_train_list = []
recall_score_val_list = []
recall_score_train_list = []
precision_score_val_list = []
precision_score_train_list = []

fig, ax = plt.subplots(1, 1, figsize=(8, 8))
fprs_list = []
tprs_list = []
auc_list = []

model_list = []
scaler_list = []

# Validação cruzada só em Training Data
for fold, (train_idx, val_idx) in enumerate(cv.split(X, y)):
    X_train = X[train_idx, :]
    y_train = y[train_idx]
    X_val = X[val_idx, :]
    y_val = y[val_idx]

    # Escala
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_val_scaled = scaler.transform(X_val)

    scaler_list.append(scaler)

# Treino
model = model_klass(**model_kwargs)
model.fit(X_train_scaled, y_train)
y_pred = model.predict(X_train_scaled)

y_pred_val = model.predict(X_val_scaled)
print(f"===== FOLD {fold} =====")
print(f"Meu resultado para treino de F1-Score é {f1_score(y_train, y_pred)}")
print(f"Meu resultado para treino de Acurácia é de {accuracy_score(y_train, y_pred)}")
print(f"Meu resultado para treino de Recall é de {recall_score(y_train, y_pred)}")
print(f"Meu resultado para treino de Precision é de {precision_score(y_train, y_pred)}")
f1_score_val_list.append(f1_score(y_val, y_pred_val))
f1_score_train_list.append(f1_score(y_train, y_pred))
accuracy_score_val_list.append(accuracy_score(y_val, y_pred_val))
accuracy_score_train_list.append(accuracy_score(y_train, y_pred))
recall_score_val_list.append(recall_score(y_val, y_pred_val))
recall_score_train_list.append(recall_score(y_train, y_pred))
precision_score_val_list.append(precision_score(y_val, y_pred_val))
precision_score_train_list.append(precision_score(y_train, y_pred))
model_list.append(model)
viz = RocCurveDisplay.from_estimator(
    model,
    X_val_scaled,
    y_val,
    ax=ax,
    alpha=0.3,
    lw=1
)
interp_fpr, interp_tpr = interpolation(viz.fpr, viz.tpr)
fprs_list.append(interp_fpr)
tprs_list.append(interp_tpr)
auc_list.append(viz.roc_auc)

```

```

print()
mean_val = np.mean(f1_score_val_list)
std_val = np.std(f1_score_val_list)
print(f"Meu resultado de F1-Score Médio de treino é {np.mean(f1_score_train_")
print(f"Meu resultado de accuracy_score Médio de treino é {np.mean(accuracy_"
print(f"Meu resultado de recall_score Médio de treino é {np.mean(recall_scor
print(f"Meu resultado de precision_score Médio de treino é {np.mean(precisic

best_model_idx = np.argmax(f1_score_val_list)
print(f"Meu melhor fold é: {best_model_idx} ")
best_model = model_list[best_model_idx]

# Fazer a inferência em Test Data
best_scaler = scaler_list[best_model_idx]
X_test_scaled = best_scaler.transform(X_test)
y_pred_test = model.predict(X_test_scaled)

# Fazer a Curva ROC
mean_fpr = np.mean(fprs_list, axis=0)
mean_tpr = np.mean(tprs_list, axis=0)
mean_auc = np.mean(auc_list)
std_auc = np.std(auc_list)

ax.plot(
    mean_fpr,
    mean_tpr,
    color='blue',
    lw=2,
    label=r"Mean ROC (AUC = %.2f $\pm$ %.2f)" %(mean_auc, std_auc)
)

```



```

ax.plot(np.linspace(0, 1, 100),
        np.linspace(0, 1, 100),
        color='g',
        ls=":",
        lw=0.5)
ax.legend()

print()
print(f"""
    Meu resultado de F1-Score para o conjunto de teste é : {f1_score(y_t
    Meu resultado de Accuracy para o conjunto de teste é : {accuracy_scc
    Meu resultado de Recall para o conjunto de teste é : {recall_score(
    Meu resultado de Precision para o conjunto de teste é: {precision_sc
return best_model, mean_val, std_val, best_scaler

```

```

In [ ]: config = [
    (LogisticRegression, {}),
    (DecisionTreeClassifier, {'min_samples_leaf': 50}),
    (SVC, {'kernel': 'rbf', 'gamma': 2}),
]

#results = []
for model_class, setting in config:
    print(model_class.__name__)
    best_model, mean_val, std_val, best_scaler = train(X_train_cv, y_train_cv, n

```

LogisticRegression**===== FOLD 0 =====**

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.81

Meu resultado para treino de Acurácia é de 0.76, Meu resultado para validação de Acurácia é de 0.74

Meu resultado para treino de Recall é de 0.89, Meu resultado para validação de R ecall é de 0.86

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.77

===== FOLD 1 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.82

Meu resultado para treino de Acurácia é de 0.75, Meu resultado para validação de Acurácia é de 0.75

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de R ecall é de 0.88

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.77

===== FOLD 2 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.82

Meu resultado para treino de Acurácia é de 0.76, Meu resultado para validação de Acurácia é de 0.75

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de R ecall é de 0.87

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.78

===== FOLD 3 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.84

Meu resultado para treino de Acurácia é de 0.76, Meu resultado para validação de Acurácia é de 0.77

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de R ecall é de 0.91

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.78

===== FOLD 4 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.81

Meu resultado para treino de Acurácia é de 0.75, Meu resultado para validação de Acurácia é de 0.74

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de R ecall é de 0.85

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.78

===== FOLD 5 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.84

Meu resultado para treino de Acurácia é de 0.75, Meu resultado para validação de Acurácia é de 0.77

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de R ecall é de 0.91

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.78

===== FOLD 6 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.81

Meu resultado para treino de Acurácia é de 0.76, Meu resultado para validação de Acurácia é de 0.73

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de Recall é de 0.86

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.76

===== FOLD 7 =====

Meu resultado para treino de F1-Score é 0.82, Meu resultado para validação de F1-Score é 0.82

Meu resultado para treino de Acurácia é de 0.75, Meu resultado para validação de Acurácia é de 0.75

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de Recall é de 0.88

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.77

===== FOLD 8 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.83

Meu resultado para treino de Acurácia é de 0.76, Meu resultado para validação de Acurácia é de 0.75

Meu resultado para treino de Recall é de 0.88, Meu resultado para validação de Recall é de 0.88

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.78

===== FOLD 9 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.83

Meu resultado para treino de Acurácia é de 0.75, Meu resultado para validação de Acurácia é de 0.77

Meu resultado para treino de Recall é de 0.89, Meu resultado para validação de Recall é de 0.88

Meu resultado para treino de Precision é de 0.78, Meu resultado para validação de Precision é de 0.8

Meu resultado de F1-Score Médio de treino é 0.83 +- 0.0016, Meu resultado de F1-Score Médio de validação é 0.82 +- 0.011

Meu resultado de accuracy_score Médio de treino é 0.76 +- 0.0022, Meu resultado de accuracy_score Médio de validação é 0.75 +- 0.014

Meu resultado de recall_score Médio de treino é 0.88 +- 0.0031. Meu resultado de recall_score Médio de validação é 0.88 +- 0.019

Meu resultado de precision_score Médio de treino é 0.78 +- 0.002, Meu resultado de precision_score Médio de validação é 0.78 +- 0.0091

Meu melhor fold é: 3

Meu resultado de F1-Score para o conjunto de teste é : 0.82

Meu resultado de Accuracy para o conjunto de teste é : 0.74

Meu resultado de Recall para o conjunto de teste é : 0.89

Meu resultado de Precision para o conjunto de teste é: 0.76

DecisionTreeClassifier

===== FOLD 0 =====

Meu resultado para treino de F1-Score é 0.85, Meu resultado para validação de F1-Score é 0.82

Meu resultado para treino de Acurácia é de 0.79, Meu resultado para validação de Acurácia é de 0.76

Meu resultado para treino de Recall é de 0.86, Meu resultado para validação de Recall é de 0.83

Meu resultado para treino de Precision é de 0.83, Meu resultado para validação de Precision é de 0.82

===== FOLD 1 =====

Meu resultado para treino de F1-Score é 0.84, Meu resultado para validação de F1-Score é 0.81

Meu resultado para treino de Acurácia é de 0.78, Meu resultado para validação de Acurácia é de 0.74

Meu resultado para treino de Recall é de 0.87, Meu resultado para validação de R ecall é de 0.84

Meu resultado para treino de Precision é de 0.82, Meu resultado para validação de Precision é de 0.78

===== FOLD 2 =====

Meu resultado para treino de F1-Score é 0.85, Meu resultado para validação de F1-Score é 0.83

Meu resultado para treino de Acurácia é de 0.79, Meu resultado para validação de Acurácia é de 0.76

Meu resultado para treino de Recall é de 0.87, Meu resultado para validação de R ecall é de 0.85

Meu resultado para treino de Precision é de 0.82, Meu resultado para validação de Precision é de 0.81

===== FOLD 3 =====

Meu resultado para treino de F1-Score é 0.83, Meu resultado para validação de F1-Score é 0.83

Meu resultado para treino de Acurácia é de 0.78, Meu resultado para validação de Acurácia é de 0.77

Meu resultado para treino de Recall é de 0.83, Meu resultado para validação de R ecall é de 0.86

Meu resultado para treino de Precision é de 0.83, Meu resultado para validação de Precision é de 0.8

===== FOLD 4 =====

Meu resultado para treino de F1-Score é 0.84, Meu resultado para validação de F1-Score é 0.83

Meu resultado para treino de Acurácia é de 0.79, Meu resultado para validação de Acurácia é de 0.76

Meu resultado para treino de Recall é de 0.86, Meu resultado para validação de R ecall é de 0.85

Meu resultado para treino de Precision é de 0.83, Meu resultado para validação de Precision é de 0.81

===== FOLD 5 =====

Meu resultado para treino de F1-Score é 0.85, Meu resultado para validação de F1-Score é 0.83

Meu resultado para treino de Acurácia é de 0.79, Meu resultado para validação de Acurácia é de 0.76

Meu resultado para treino de Recall é de 0.87, Meu resultado para validação de R ecall é de 0.87

Meu resultado para treino de Precision é de 0.82, Meu resultado para validação de Precision é de 0.8

===== FOLD 6 =====

Meu resultado para treino de F1-Score é 0.84, Meu resultado para validação de F1-Score é 0.8

Meu resultado para treino de Acurácia é de 0.79, Meu resultado para validação de Acurácia é de 0.73

Meu resultado para treino de Recall é de 0.85, Meu resultado para validação de R ecall é de 0.8

Meu resultado para treino de Precision é de 0.84, Meu resultado para validação de Precision é de 0.8

===== FOLD 7 =====

Meu resultado para treino de F1-Score é 0.84, Meu resultado para validação de F1-Score é 0.85

Meu resultado para treino de Acurácia é de 0.78, Meu resultado para validação de Acurácia é de 0.8

Meu resultado para treino de Recall é de 0.85, Meu resultado para validação de R ecall é de 0.87

Meu resultado para treino de Precision é de 0.82, Meu resultado para validação de Precision é de 0.84

===== FOLD 8 =====

Meu resultado para treino de F1-Score é 0.84, Meu resultado para validação de F1-Score é 0.83

Meu resultado para treino de Acurácia é de 0.78, Meu resultado para validação de Acurácia é de 0.77

Meu resultado para treino de Recall é de 0.85, Meu resultado para validação de Recall é de 0.83

Meu resultado para treino de Precision é de 0.83, Meu resultado para validação de Precision é de 0.83

===== FOLD 9 =====

Meu resultado para treino de F1-Score é 0.84, Meu resultado para validação de F1-Score é 0.81

Meu resultado para treino de Acurácia é de 0.79, Meu resultado para validação de Acurácia é de 0.75

Meu resultado para treino de Recall é de 0.85, Meu resultado para validação de Recall é de 0.8

Meu resultado para treino de Precision é de 0.84, Meu resultado para validação de Precision é de 0.82

Meu resultado de F1-Score Médio de treino é 0.84 +- 0.0043, Meu resultado de F1-Score Médio de validação é 0.82 +- 0.014

Meu resultado de accuracy_score Médio de treino é 0.79 +- 0.0046, Meu resultado de accuracy_score Médio de validação é 0.76 +- 0.017

Meu resultado de recall_score Médio de treino é 0.85 +- 0.012. Meu resultado de recall_score Médio de validação é 0.84 +- 0.023

Meu resultado de precision_score Médio de treino é 0.83 +- 0.0063, Meu resultado de precision_score Médio de validação é 0.81 +- 0.015

Meu melhor fold é: 7

Meu resultado de F1-Score para o conjunto de teste é : 0.82

Meu resultado de Accuracy para o conjunto de teste é : 0.76

Meu resultado de Recall para o conjunto de teste é : 0.84

Meu resultado de Precision para o conjunto de teste é: 0.8

SVC

===== FOLD 0 =====

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.85

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.78

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.97

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.76

===== FOLD 1 =====

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.87

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.8

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.97

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.79

===== FOLD 2 =====

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.86

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.79

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.98

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.77
===== FOLD 3 ======

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.86

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.78

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.97

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.77
===== FOLD 4 ======

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.86

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.79

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.98

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.77
===== FOLD 5 ======

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.86

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.78

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.97

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.77
===== FOLD 6 ======

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.87

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.8

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.99

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.77
===== FOLD 7 ======

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.86

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.79

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.99

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.77
===== FOLD 8 ======

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.86

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de Acurácia é de 0.79

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.99

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.76
===== FOLD 9 ======

Meu resultado para treino de F1-Score é 0.99, Meu resultado para validação de F1-Score é 0.85

Meu resultado para treino de Acurácia é de 0.99, Meu resultado para validação de

Acurácia é de 0.78

Meu resultado para treino de Recall é de 1.0, Meu resultado para validação de Recall é de 0.97

Meu resultado para treino de Precision é de 0.99, Meu resultado para validação de Precision é de 0.76

Meu resultado de F1-Score Médio de treino é 0.99 +- 0.00057, Meu resultado de F1-Score Médio de validação é 0.86 +- 0.0049

Meu resultado de accuracy_score Médio de treino é 0.99 +- 0.00077, Meu resultado de accuracy_score Médio de validação é 0.79 +- 0.0079

Meu resultado de recall_score Médio de treino é 1.0 +- 0.00055. Meu resultado de recall_score Médio de validação é 0.98 +- 0.007

Meu resultado de precision_score Médio de treino é 0.99 +- 0.0011, Meu resultado de precision_score Médio de validação é 0.77 +- 0.0067

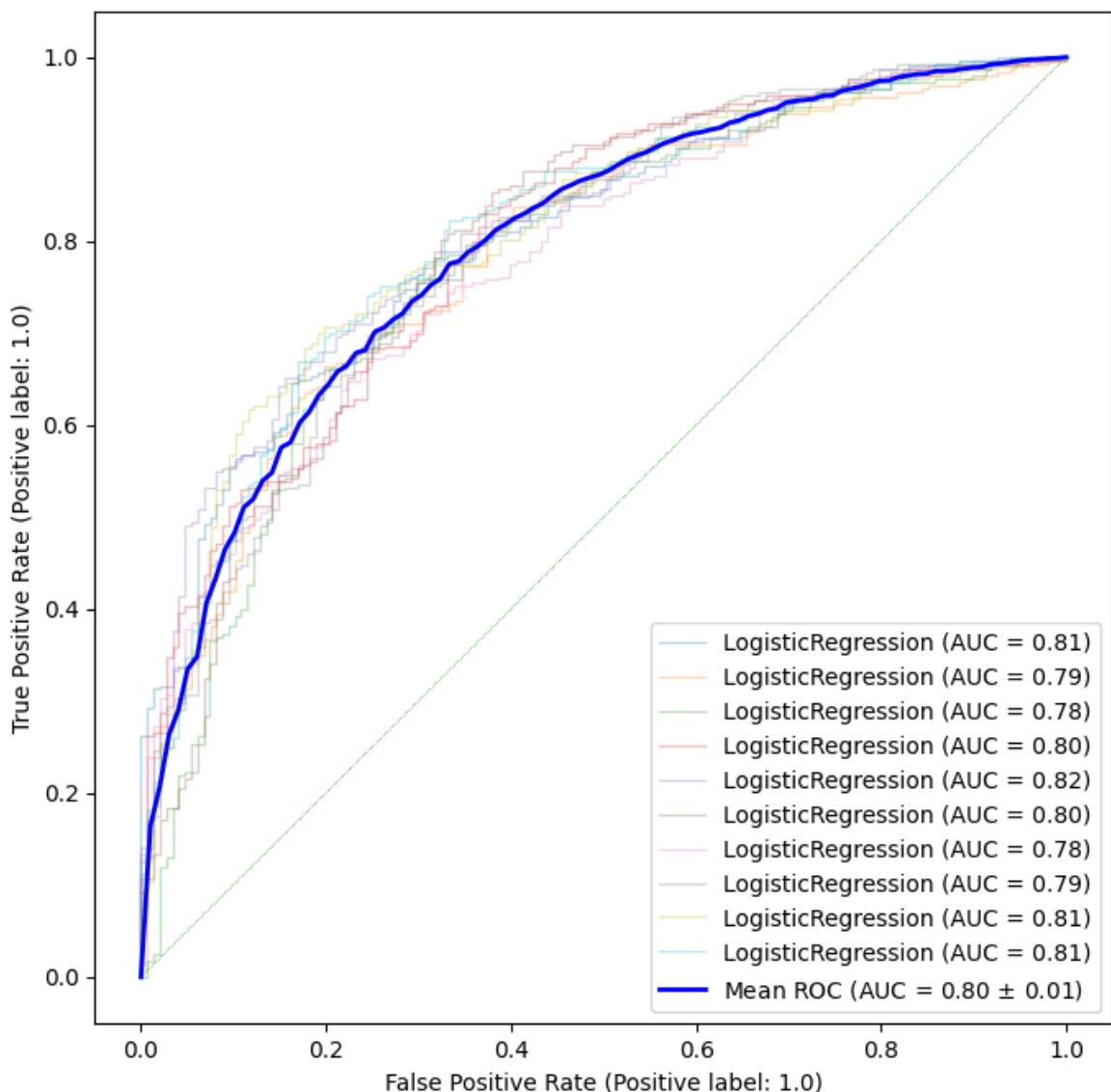
Meu melhor fold é: 6

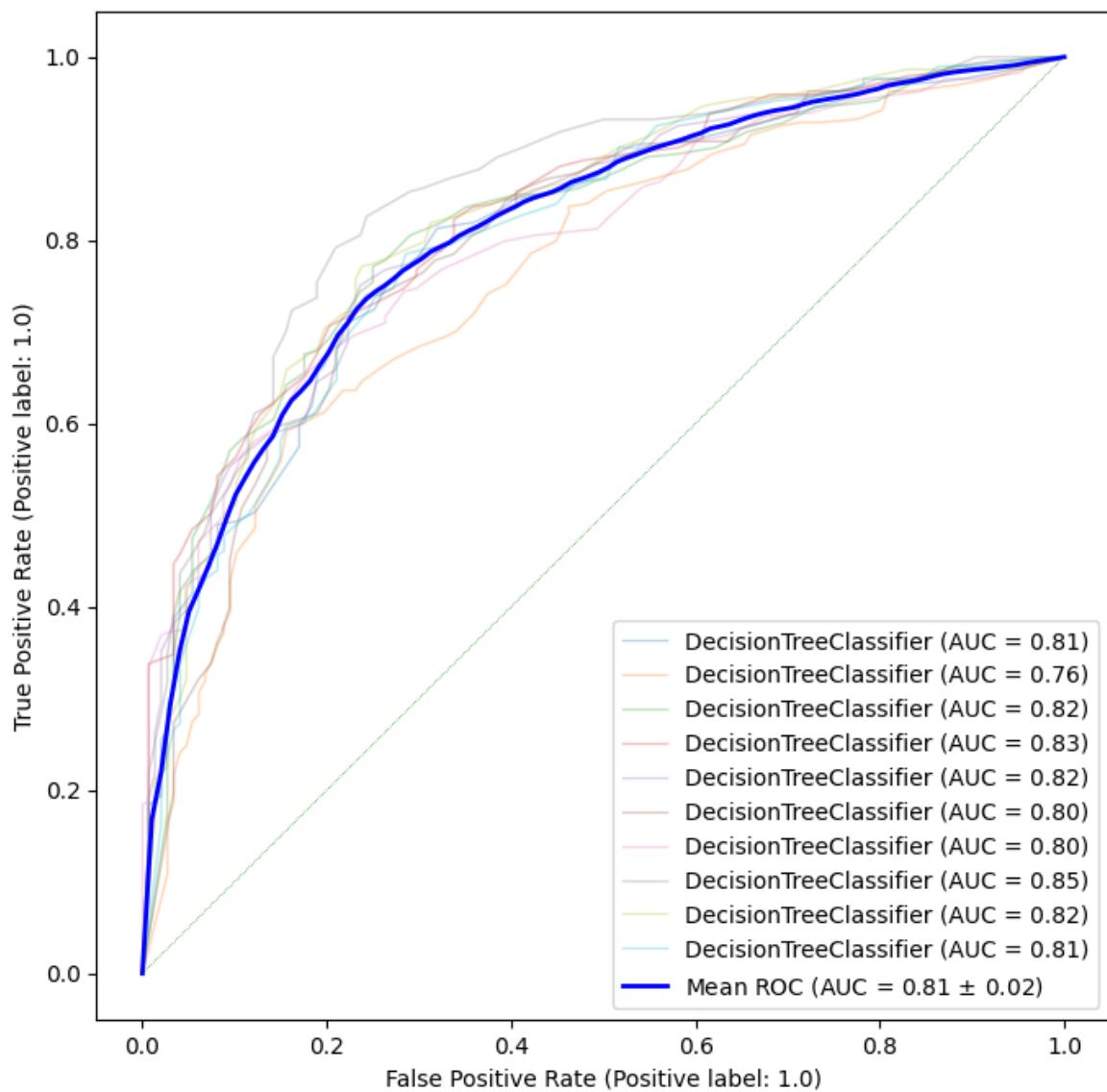
Meu resultado de F1-Score para o conjunto de teste é : 0.85

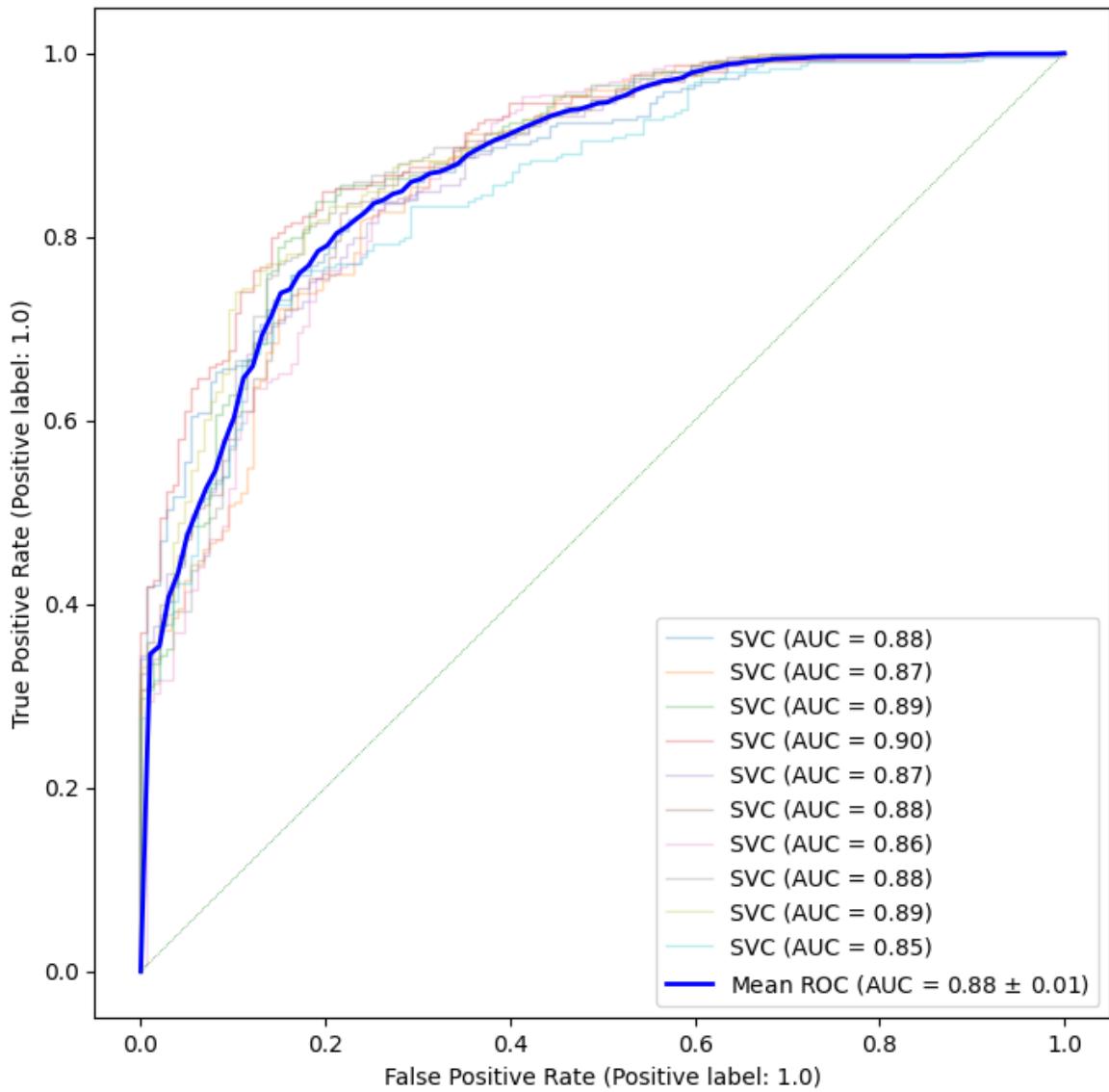
Meu resultado de Accuracy para o conjunto de teste é : 0.77

Meu resultado de Recall para o conjunto de teste é : 0.97

Meu resultado de Precision para o conjunto de teste é: 0.75







Agora vou utilizar o retorno da função train (bestmodel e best scaler) para atribuir esses valores a uma base não treinada que chamarei de base real (será o dataframe dos vinhos tintos) e assim estabelecer o f1 score, acurácia , recall e precisão para a base real, em relação a base testada.

In []: best_model

Out[]: ▾ SVC

SVC(gamma=2)

Base de vinho tinto, eu estava usando o nome como categoria e vou mudar para aderir ao padrão do exercício.

In []: df_tr_wine.rename(columns={'category': 'opinion'}, inplace = True)

Utilizando Xreal e Y real para os dados do vinho tinto. as funções best model e best scaler são retornos da função train(onde foi feito o treino).

```
In [ ]: X_real = df_tr_wine[var]
y_real = df_tr_wine['opinion']
X_real_scaled = best_scaler.transform(X_real)
y_pred_real = best_model.predict(X_real_scaled)
```

e:\miniconda3\envs\Bootcamp\lib\site-packages\sklearn\base.py:432: UserWarning: X has feature names, but StandardScaler was fitted without feature names
warnings.warn(

```
In [ ]: print(f"A acurácia real é {100 * accuracy_score(y_real, y_pred_real):.2f} %")
print(f"A recall real é {100 * recall_score(y_real, y_pred_real):.2f} %")
print(f"A precisão real é {100* precision_score(y_real, y_pred_real):.2f} %")
print(f>O F1 Score real = {f1_score(y_real, y_pred_real):.2f} ")
print(classification_report(y_real, y_pred_real))
```

A acurácia real é 53.53 %

A recall real é 100.00 %

A precisão real é 53.50 %

O F1 Score real = 0.70

	precision	recall	f1-score	support
0.0	1.00	0.00	0.00	744
1.0	0.54	1.00	0.70	855
accuracy			0.54	1599
macro avg	0.77	0.50	0.35	1599
weighted avg	0.75	0.54	0.37	1599

como podemos ver , para os vinhos tintos temos uma redução da precisão , da acurácia e do f1 score cairam bastante, isso se dá pela diferença das características de vinhos, como o treinamento foi feito com vinho branco e o real com vinho tinto, algumas das características são diferentes, isso mostra pelo gráfico de distribuição das qualidades, que foi feito na etapa de análise exploratória e nas correlações entre as variáveis, em que no tinto diferem um pouco em relação ao do branco, levando a um treinamento enviesado quando tentamos colocar a base de tinto.

O melhor treinamento seria com a Base total , para que essas características diferentes fossem equalizadas e tivessemos um valor médio entre os 2 tipos de vinho.