
WINE ANALYSIS

PROJECT 4
TEAM 2

- Kimberly Childers
- Fernanda Valdez
- Misha Borunda

PROJECT SUMMARY

This project utilizes three different data sets to do a comprehensive analysis about wine.

Two of the chosen datasets will be used for Machine Learning to train a model and make predictions about red and white wines based on their physicochemical composition.

In addition to the Machine Learning portion of the project, there will be an analysis and visualization done on Tableau to portray insights from the datasets and create an interactive way for users to learn about wine types and quality and the variables that affect it.

At the end of the project, there will be a Full Stack Web Application combining our ML Research, Tableau dashboards, and reports about wine, its quality and other characteristics.

DATA SETS

- **Wine Quality Data Sets**
  - One data set is for red wine only and the other one is for white wine only
  - Both Datasets include the same physicochemical variables
    - https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009?resource=download
    - https://www.kaggle.com/datasets/piyushagni5/white-wine-quality

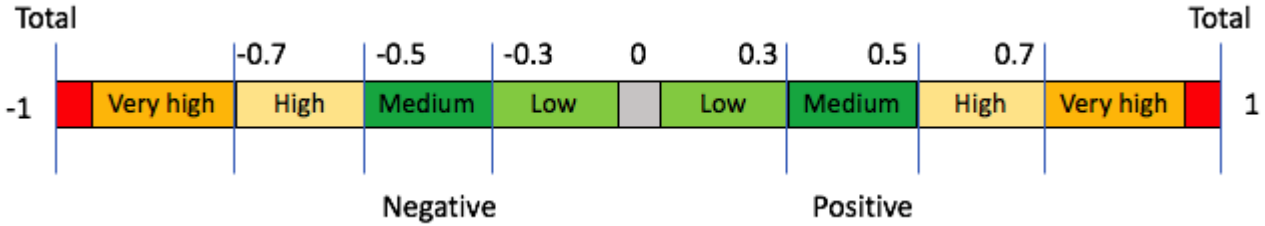
- **Wine Reviews Data Set**
  - includes wine regions, points and prices
    - https://www.kaggle.com/datasets/zynicide/wine-reviews

DATA CLEANING

- Two data sets were merged for the ML portion of the analysis to be able to create a model to predict wine types (Red vs White) based on physical and chemical characteristics
- There were no Null-Values, therefore rows of data did not need to be dropped
- A new column “wine_type” was created to identify wine types in the combined data frame
- One-Hot encoding was used to classify red wines as “1” and white wines as “0”
- We had “Unbalanced” Data since 75% of wines were white and only 25% were red
- SMOTE (over sampling technique) and NEAR MISS (under sampling technique) were used to balance the data

DEFINITIONS

- **Fixed Acidity**
  - Is the set of the wine's natural acids
  - It preserves the wine's natural qualities, as well as its color
- **Volatile Acidity**
  - Is a measure of a wine's gaseous acids
  - The amount of VA in wine is often considered an indicator of spoilage
- **Citric Acid**
  - It can be added to finished wines to increase acidity and give a "fresh" flavortive or additive to food or drink to add a sour taste
- **Residual Sugar**
  - It is the natural grape sugars leftover in a wine after the alcoholic fermentation finishes
- **Chlorides**
  - Depend on the geographic, geologic and climatic conditions of vine culture
  - The content is increased in wines coming from vineyards which are near the sea coast
- **Free Sulfur Dioxide**
  - Sulfur Dioxide Ions that are not chemically bonded to other chemicals
  - Too much sulfur in wine can cause danger to health if drank excessively
- **Density**
  - Mass/Volume – How "heavy" the wine is
- **PH**
  - Influences the taste of the final product, but also its color, oxidation and chemical stability.
- **Alcohol**

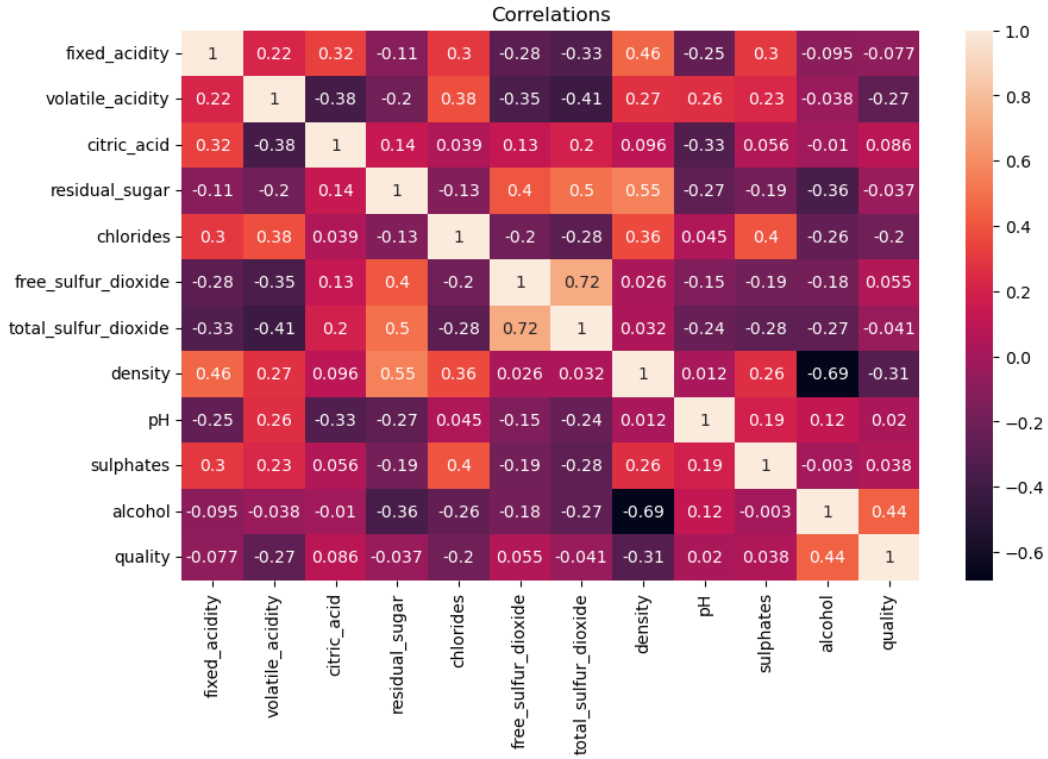


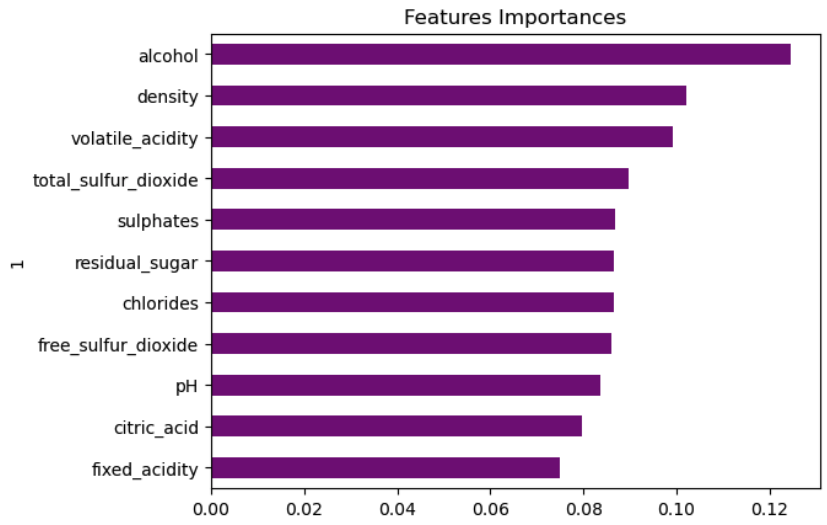
ML APPROACH 1

Predict Wine Quality based on its physical and chemical properties

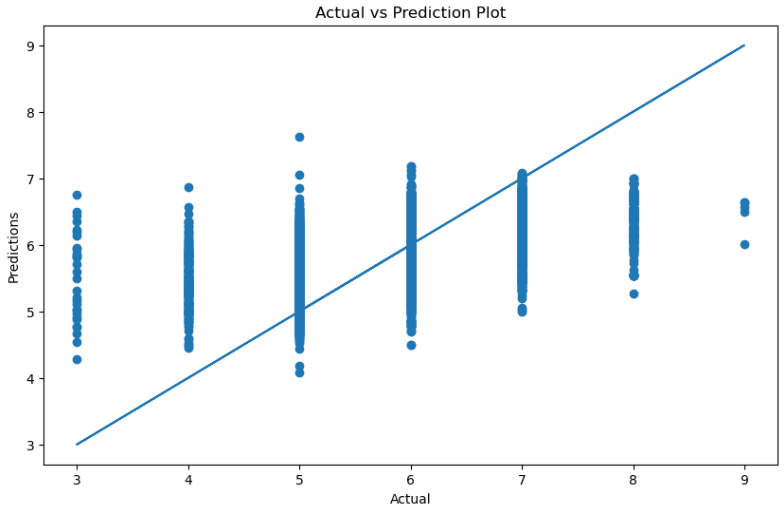
Alcohol, Density and Volatile Acidity were the only features that presented a "Moderate" Correlation

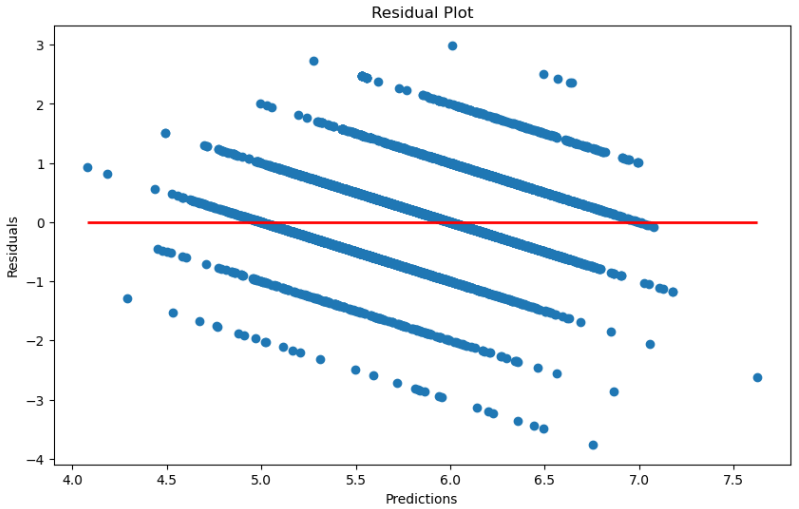
Since there were no strong correlations, we decided to change our model to predict wine type (Red vs White) instead





ML MODEL 1 FINDINGS





LINEAR REGRESSION

Accuracy Score : 0.6793846153846154
Classification Report

	precision	recall	f1-score	support
3	0.00	0.00	0.00	9
4	0.86	0.12	0.21	50
5	0.71	0.72	0.72	543
6	0.65	0.77	0.71	707
7	0.66	0.53	0.59	256
8	0.89	0.41	0.56	59
9	0.00	0.00	0.00	1
accuracy			0.68	1625
macro avg	0.54	0.36	0.40	1625
weighted avg	0.69	0.68	0.67	1625

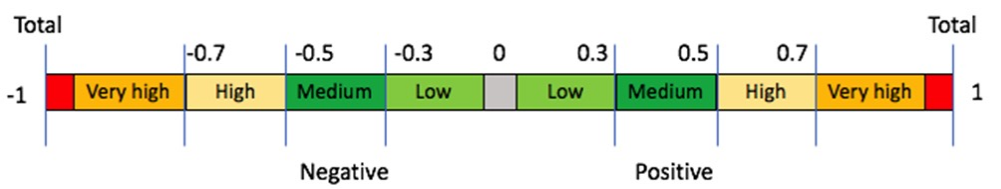
RANDOM FOREST

KNN

	precision	recall	f1-score	s
3	0.20	0.11	0.14	
4	0.15	0.15	0.15	
5	0.59	0.55	0.57	
6	0.58	0.57	0.57	
7	0.46	0.54	0.49	
8	0.18	0.28	0.22	
9	0.00	0.00	0.00	
accuracy			0.54	
macro avg	0.31	0.31	0.31	
ighted avg	0.54	0.54	0.54	

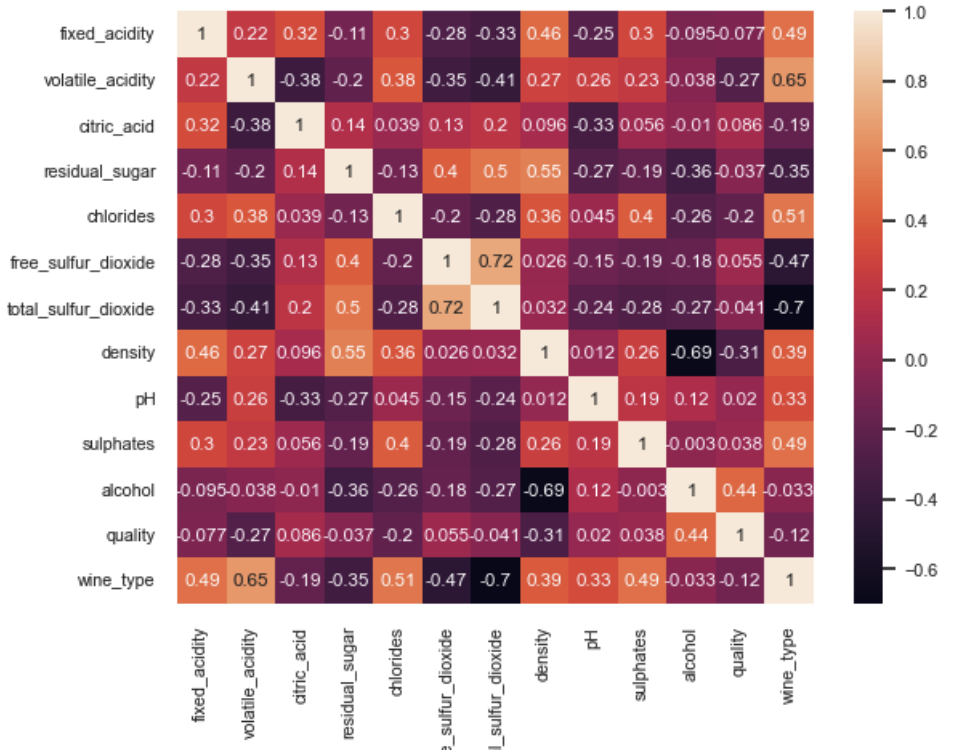
ML 2 FINDINGS

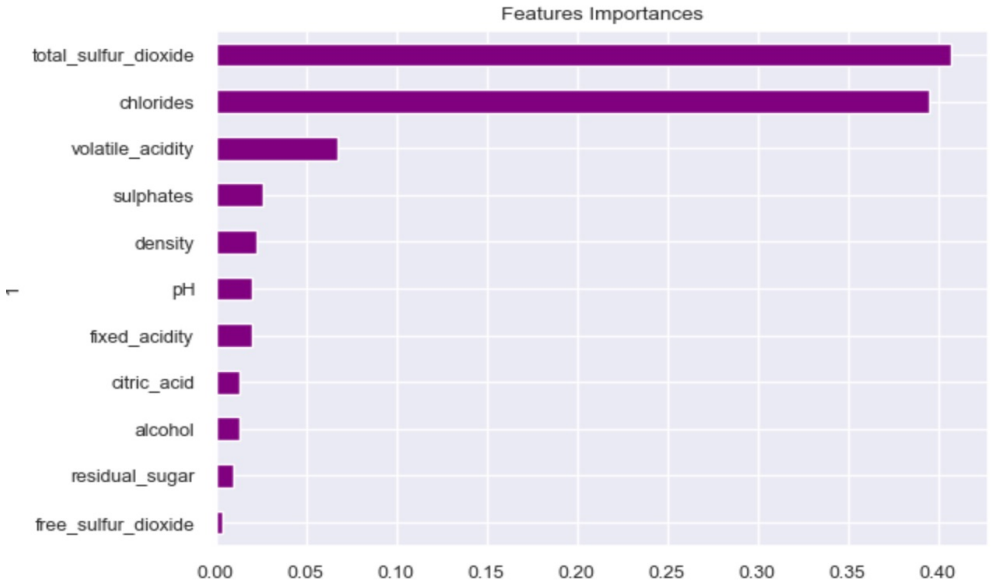
VS.



ML APPROACH 2

- Predicts wine type (Red vs White) based on physical and chemical properties
- Out of 11 features considered, 9 of them had moderate to strong correlations with wine type.
- The strongest correlations being Total Sulfur Dioxide (-0.70), Volatile Acidity (0.65), Chlorides (0.51), and Sulfates (0.49)





LOGISTIC REGRESSION - SMOTE

BEFORE BALANCING DATA

```python
# Non-resampled data
# logistic regression object
lr = LogisticRegression()

# train the model on train set
lr.fit(X_train, y_train)

predictions = lr.predict(X_test)

# print classification report
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1225
           1       0.97      0.96      0.97       400

    accuracy                           0.99      1625
   macro avg       0.98      0.98      0.98      1625
weighted avg       0.99      0.99      0.99      1625
```

AFTER BALANCING DATA

```python
# Resampled data
# logistic regression object
lr = LogisticRegression()

# train the model on train set
lr.fit(X_train_res, y_train_res)

predictions = lr.predict(X_test)

# print classification report
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99      1225
           1       0.95      0.99      0.97       400

    accuracy                           0.98      1625
   macro avg       0.97      0.98      0.98      1625
weighted avg       0.98      0.98      0.98      1625
```

LOGISTIC REGRESSION - NEARMISS

BEFORE BALANCING DATA

```
# Non-resampled data
# logistic regression object
lr = LogisticRegression()

# train the model on train set
lr.fit(X_train, y_train)

predictions = lr.predict(X_test)

# print classification report
print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	1225
1	0.97	0.96	0.97	400
accuracy			0.99	1625
macro avg	0.98	0.98	0.98	1625
weighted avg	0.99	0.99	0.99	1625

AFTER BALANCING DATA

```
# Resampled data
# train the model on training set
lr2 = LogisticRegression()
lr2.fit(X_train_miss, y_train_miss.ravel())
predictions = lr2.predict(X_test)

# print classification report
print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	1191
1	0.97	0.94	0.96	434
accuracy			0.98	1625
macro avg	0.98	0.97	0.97	1625
weighted avg	0.98	0.98	0.98	1625

NEURAL NETWORK NN1

```python
# Evaluate the model using test data
model_loss, model_accuracy = nn1.evaluate(X_test, y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
51/51 - 0s - 2ms/step - accuracy: 0.9434 - loss: 0.1691
Loss: 0.1690843552350998, Accuracy: 0.94338458776474
```

```python
# Compile, Train, and Evaluate model
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
nn1 = tf.keras.models.Sequential()

# Added first Dense layer
nn1.add(tf.keras.layers.Dense(units=5, activation="relu", input_dim=len(X.columns))) # we have 44 features

# Added second layer
nn1.add(tf.keras.layers.Dense(units=3, activation="relu"))

# Added output layer
nn1.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

# Check structure of the Sequential model
nn1.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 5)	60
dense_1 (Dense)	(None, 3)	18
dense_2 (Dense)	(None, 1)	4

Total params: 82 (328.00 B)

Trainable params: 82 (328.00 B)

Non-trainable params: 0 (0.00 B)

TRAINING METRICS

Train Confusion Matrix:

```
[[3598   75]
 [ 248  951]]
```

Train Report:

	precision	recall	f1-score	support
0	0.94	0.98	0.96	3673
1	0.93	0.79	0.85	1199
accuracy			0.93	4872
macro avg	0.93	0.89	0.91	4872
weighted avg	0.93	0.93	0.93	4872

TESTING METRICS

Test Confusion Matrix:

```
[[1206   19]
 [  73  327]]
```

Test Report:

	precision	recall	f1-score	support
0	0.94	0.98	0.96	1225
1	0.95	0.82	0.88	400
accuracy			0.94	1625
macro avg	0.94	0.90	0.92	1625
weighted avg	0.94	0.94	0.94	1625

NEURAL NETWORK **NN2**

```python
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.

nn2 = tf.keras.models.Sequential()

# Add first Dense layer, including the input layer
nn2.add(tf.keras.layers.Dense(units=15, activation="relu", input_dim=len(X.columns))) # we have 44 features

# Second layer
nn2.add(tf.keras.layers.Dense(units=7, activation="relu"))

# Third layer
nn2.add(tf.keras.layers.Dense(units=5, activation="relu"))

# Output layer
nn2.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

# Check the Sequential model
nn2.summary()
```

```python
# Evaluate model with test data
model_loss, model_accuracy = nn2.evaluate(X_test, y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
51/51 - 0s - 2ms/step - accuracy: 0.9840 - loss: 0.0786
Loss: 0.07855309545993805, Accuracy: 0.984000027179718
```

TRAINING METRICS

Train Confusion Matrix:

```
[[3612   61]
 [  70 1129]]
```

Train Report:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	3673
1	0.95	0.94	0.95	1199
accuracy			0.97	4872
macro avg	0.96	0.96	0.96	4872
weighted avg	0.97	0.97	0.97	4872

TESTING METRICS

Test Confusion Matrix:

```
[[1212   13]
 [  13  387]]
```

Test Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	1225
1	0.97	0.97	0.97	400
accuracy			0.98	1625
macro avg	0.98	0.98	0.98	1625
weighted avg	0.98	0.98	0.98	1625

NEURAL NETWORK **NN3**

```python
# Define model - deep neural net, i.e., the number of input features and hidden nodes for each layer.

nn3 = tf.keras.models.Sequential()

# Add first Dense layer
nn3.add(tf.keras.layers.Dense(units=15, activation="relu", input_dim=len(X.columns))) # we have 44 features

# Second layer
nn3.add(tf.keras.layers.Dense(units=7, activation="relu"))

nn3.add(tf.keras.layers.Dense(units=5, activation="relu"))

# Output layer that uses a probability activation function
nn3.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

# Check the Sequential model
nn3.summary()
```

```python
# Evaluate model with test data
model_loss, model_accuracy = nn3.evaluate(X_test_scaled, y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
51/51 - 0s - 2ms/step - accuracy: 0.9951 - loss: 0.0304
Loss: 0.030397454276680946, Accuracy: 0.9950768947601318
```

```
Train Confusion Matrix:
[[3705    2]
 [  13 1152]]

Train Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3707
           1       1.00      0.99      0.99      1165

    accuracy                           1.00      4872
   macro avg       1.00      0.99      1.00      4872
weighted avg       1.00      1.00      1.00      4872
```

```
TESTING METRICS

Test Confusion Matrix:
[[1189    2]
 [   6  428]]

Test Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      1191
           1       1.00      0.99      0.99       434

    accuracy                           1.00      1625
   macro avg       1.00      0.99      0.99      1625
weighted avg       1.00      1.00      1.00      1625
```

TESTING METRICS

Test Confusion Matrix:

```
[[1223    2]
 [   1  399]]
```

Test Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1225
1	1.00	1.00	1.00	400
accuracy			1.00	1625
macro avg	1.00	1.00	1.00	1625
weighted avg	1.00	1.00	1.00	1625

TRAINING METRICS

Train Confusion Matrix:

```
[[3672    1]
 [   0 1199]]
```

Train Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3673
1	1.00	1.00	1.00	1199
accuracy			1.00	4872
macro avg	1.00	1.00	1.00	4872
weighted avg	1.00	1.00	1.00	4872

XGB

CONCLUSION

- Red and White wines are relatively “easy” to distinguish based on their properties and therefore it makes sense that our machine learning models have high accuracy
- Quality has only a moderate correlation with the physical and chemical properties of wine and therefore the machine learning models presented a lower level of accuracy

WEBSITE DEMONSTRATION

REFERENCES

- https://agrovin.com/en/techniques-for-correcting-wine-acidity/#:~:text=Fixed%20acidity%20is%20the%20set,of%20tartaric%20acid%20per%20litre.
- https://www.wineenthusiast.com/basics/drinks-terms-defined/volatile-acidity-wine/
- https://wineserver.ucdavis.edu/industry-info/enology/methods-and-techniques/common-chemical-reagents/citric-acid
- https://winefolly.com/deep-dive/what-is-residual-sugar-in-wine/
- https://wineamerica.org/the-magic-of-wine/wine-facts/
- https://www.randoxfood.com/why-is-testing-for-free-sulphite-so2-important-in-winemaking/#:~:text=The%20sulphur%20dioxide%20ions%20that,to%20health%20if%20drank%20excessively.