Wine Study

Team 2: M. Borunda, K. Childers, and F. Valdez

Southern Methodist University

Data Science Bootcamp (09/2023)

Introduction

This study explores wine variants of “Vinho Verde” wine from Portugal. Team 2 was interested to find any correlation between these wine variants and their features which included fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality score. A database of these features was developed by combining two datasets for machine learning. The machine learning would help with developing a predictive model on wine type (red or white) based on the features. The results would educate the consumer on what aspects make a high-quality wine and its wine type. A third data set was used to bring in cost, region of origin, and rated points to enhance findings from the first and second datasets and determine price points and regions of high-quality wine.

Data Cleaning

The first two datasets were transformed for machine learning by first cleaning and combining them. The first data set brought in red wines and the second one consisted of white. To combine them, team two had to reformat the second set to match the first. Then a column had to be added to each data set to label the wines as red or white.

```
#See what is in the first dataset containing red wines
df_red = pd.read_csv("wine_quality.csv")
df_red.head()
```

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

```
#See what is in the second dataset containg white wines
df_white = pd.read_csv("winequality-white.csv")
df_white.head()
```

	fixed acidity;"volatile acidity";"citric acid";"residual sugar";"chlorides";"free sulfur dioxide";"total sulfur dioxide";"density";"pH";"sulphates";"alcohol";"quality"
0	7;0.27;0.36;20.7;0.045;45;170;1.001;3;0.45;8.8;6
1	6.3;0.3;0.34;1.6;0.049;14;132;0.994;3.3;0.49;9...
2	8.1;0.28;0.4;6.9;0.05;30;97;0.9951;3.26;0.44;1...
3	7.2;0.23;0.32;8.5;0.058;47;186;0.9956;3.19;0.4...
4	7.2;0.23;0.32;8.5;0.058;47;186;0.9956;3.19;0.4...

Figure 1: Original read in of dataset 1 & 2

```
#Added wine type column in first dataset to label all of these wines red
df_red["wine_type"]="red"
df_red.head()
```

fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality	wine_type
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	red
7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	red
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	red
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	red
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	red

```
#Added wine type column in second dataset to label all of these wines white
df_white["wine_type"]="white"
df_white.head()
```

fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality	wine_type
7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6	white
6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6	white
8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6	white
7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6	white
7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6	white

```
#Combining the first and second dataset of red and white wines
```

Figure 2: Added wine type columns to dataset 1 & 2

```
#Combining the first and second dataset of red and white wines
pd.concat([df_red, df_white], ignore_index=True)
```

fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality	wine_type
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	red
7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5	red
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5	red
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6	red
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	red
...	...	...	...	...	...	...	...	...	...	...	...	...
6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	0.50	11.2	6	white
6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	0.46	9.6	5	white
6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	0.46	9.4	6	white
5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	0.38	12.8	7	white
6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	0.32	11.8	6	white

ows × 13 columns

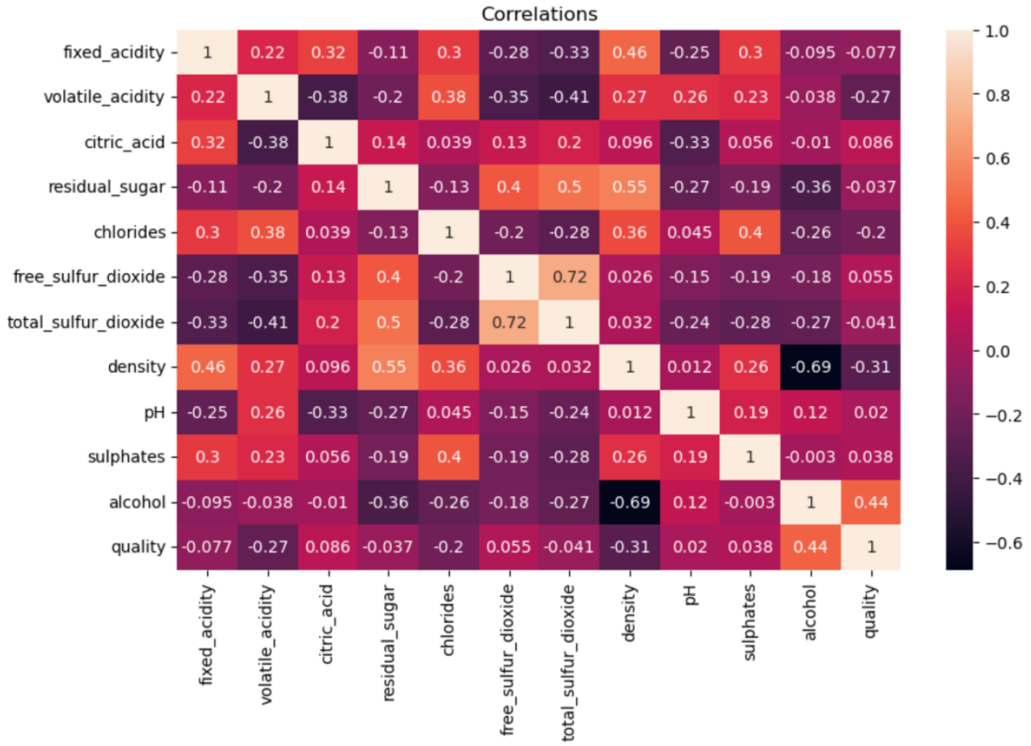
```
#Name new combined data frame
df_combined = pd.concat([df_red, df_white], ignore_index=True)
```

```
#Export to csv file for Tableau
df_combined.to_csv("combined_datasets.csv")
```

Figure 3: Combined datasets 1 & 2

The range of the data was not extreme, and scaling was not needed. It did look like there was a large skew due to the number of white wines. After analysis, about 93% of the data fell between the quality rating 5-7. The features “fixed_acidity” and “alcohol” seemed to better determents of wine type. When looking for correlation between the wine features and quality the most correlation that could be seen (it was from low to moderate) was “alcohol”, “density”, “volatile_acidity”, and “chlorides” as a

close runner up with -0.2.




```
quality                1.000000
alcohol                0.444319
density                0.305858
volatile_acidity       0.265699
chlorides              0.200666
citric_acid            0.085532
fixed_acidity          0.076743
free_sulfur_dioxide    0.055463
total_sulfur_dioxide   0.041385
sulphates              0.038485
residual_sugar         0.036980
pH                     0.019506
Name: quality, dtype: float64
```

Figure 4: Correlation between wine features

Findings/Analysis

Machine learning was performed on the dataset to create a predictive model. Team 2 first performed various analysis to determine the best model to predict quality of wine. The first analysis utilized linear regression. The values that came out showed that it was not the best model to use. The R2 value was low (below 0.4), meaning there was low correlation. The MSE value was slightly higher

than team 2 would prefer because the closer to 0, the better the model. In this case it was around 0.54. In addition, the RMSE was kind of high, the model is better the lower it is. Finally, the MAE value should be lower for a better fit model, but it was kind of high as well.

```
R2: 0.2921368850399051
MSE: 0.5397154672785655
RMSE: 0.7346532973304928
MAE: 0.5683159023114154
```

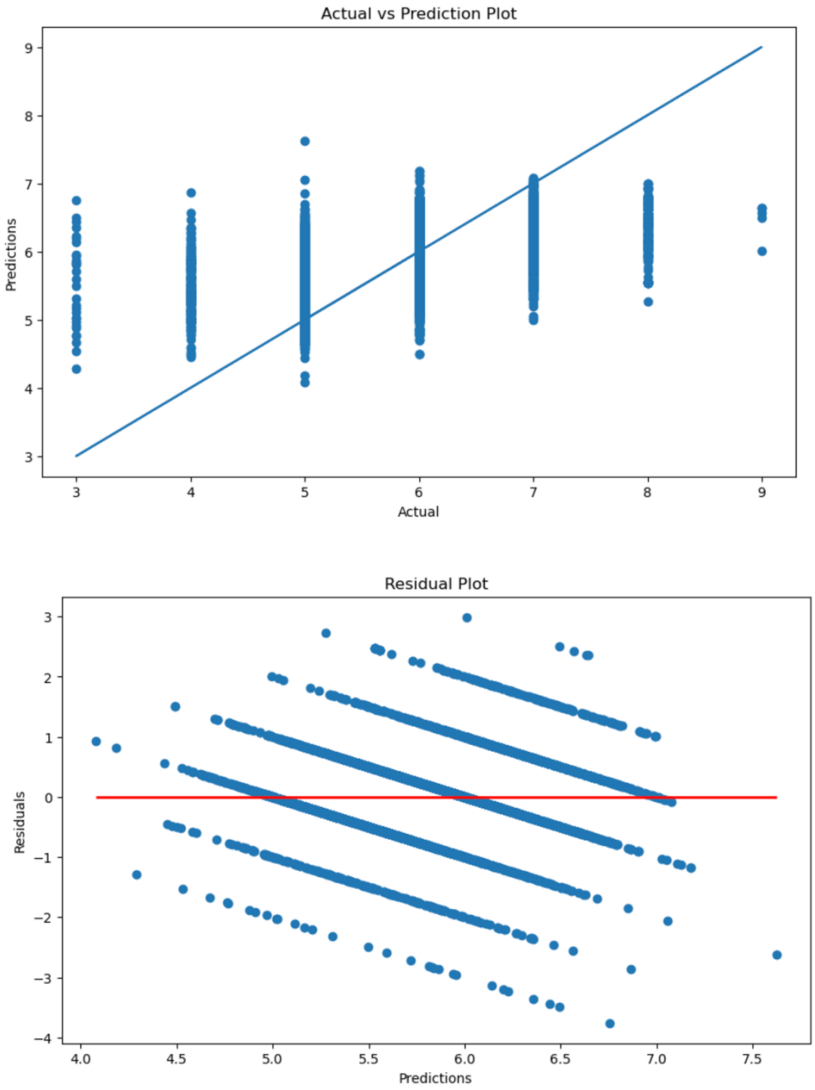


Figure 5: Linear regression analysis to predict wine quality

Random Forest was utilized next to determine if it would be a better fit model to predict wine quality. It had an accuracy score of 68%. It did verify that alcohol, density, and volatile were the best determinants of wine quality as seen in figure 7 below.

Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Predicted 6
Actual 0	0	0	7	2	0	0	0
Actual 1	0	6	30	14	0	0	0
Actual 2	0	0	393	146	4	0	0
Actual 3	0	1	115	546	45	0	0
Actual 4	0	0	5	113	135	3	0
Actual 5	0	0	0	15	20	24	0
Actual 6	0	0	0	0	1	0	0

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

Figure 6: Random Forest analysis

```
# Feature Importance
importances = rf_model.feature_importances_
# We can sort the features by their importance
sorted(zip(rf_model.feature_importances_, X.columns), reverse=True)
```

```
[(0.12460578951843379, 'alcohol'),
 (0.1020438172629673, 'density'),
 (0.09918870542344221, 'volatile_acidity'),
 (0.0897416811311875, 'total_sulfur_dioxide'),
 (0.08681526334547943, 'sulphates'),
 (0.08666350180583153, 'residual_sugar'),
 (0.08651101661700283, 'chlorides'),
 (0.08608903621030604, 'free_sulfur_dioxide'),
 (0.08370213965418391, 'pH'),
 (0.07960195051325691, 'citric_acid'),
 (0.07503709851790852, 'fixed_acidity')]
```

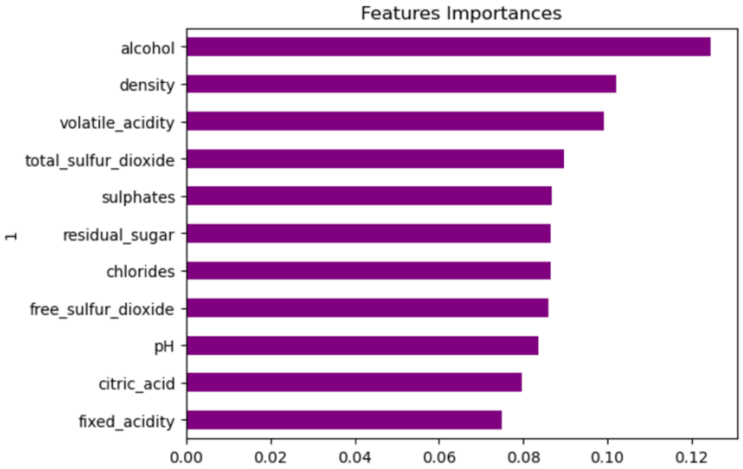


Figure 7: Feature importance

KNN analysis was the next analysis and was only a 54% accuracy. PCA was the next technique and came out okay as a predictive model.

```
R2: 1.0
MSE: 1.3004002300934333e-30
RMSE: 1.1403509240989957e-15
MAE: 8.380073438143402e-16
```

Figure 8: PCA analysis

Decision Tree Analysis was performed, and accuracy was about 61% as seen in the figure 9 below.

Confusion Matrix

	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Predicted 6
Actual 0	0	0	5	4	0	0	0
Actual 1	1	13	17	12	6	1	0
Actual 2	2	18	356	138	25	4	0
Actual 3	2	12	135	464	82	12	0
Actual 4	0	4	24	83	136	9	0
Actual 5	0	0	2	18	13	26	0
Actual 6	0	0	0	0	1	0	0

```
Accuracy Score : 0.6123076923076923
Classification Report
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         9
           4       0.28      0.26      0.27        50
           5       0.66      0.66      0.66       543
           6       0.65      0.66      0.65       707
           7       0.52      0.53      0.52       256
           8       0.50      0.44      0.47        59
           9       0.00      0.00      0.00         1

    accuracy                           0.61      1625
   macro avg       0.37      0.36      0.37      1625
weighted avg       0.61      0.61      0.61      1625
```

Figure 9: Decision tree analysis

In conclusion, team two decided that quality did not have strong enough correlations to the other wine features to use in a predictive model. Therefore, attention was switched to focus on predicting wine type as red or white based on wine features (jupyter notebook title "machine_learning_classification"). One-hot encoding was implemented to id the wine type as either 1 for red or 0 for white. Predicting wine type had much stronger correlations as seen in figure 11.

```
#One-Hot Encoding
df_combined2["wine_type"] = np.where(df_combined2["wine_type"]=="red", 1, 0)
df_combined2.head()

#df_combined2["wine_type"] = df_combined2.wine_type.astype(str)
##df_combined2.head()
```

fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality	wine_type
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	1
7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	1
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	1
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	1
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	1

```
df_combined2.tail()
```

fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality	wine_type
6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	0.50	11.2	6	0
6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	0.46	9.6	5	0
6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	0.46	9.4	6	0
5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	0.38	12.8	7	0
6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	0.32	11.8	6	0

Figure 10: Converting wine types to 1 for red and 0 for white

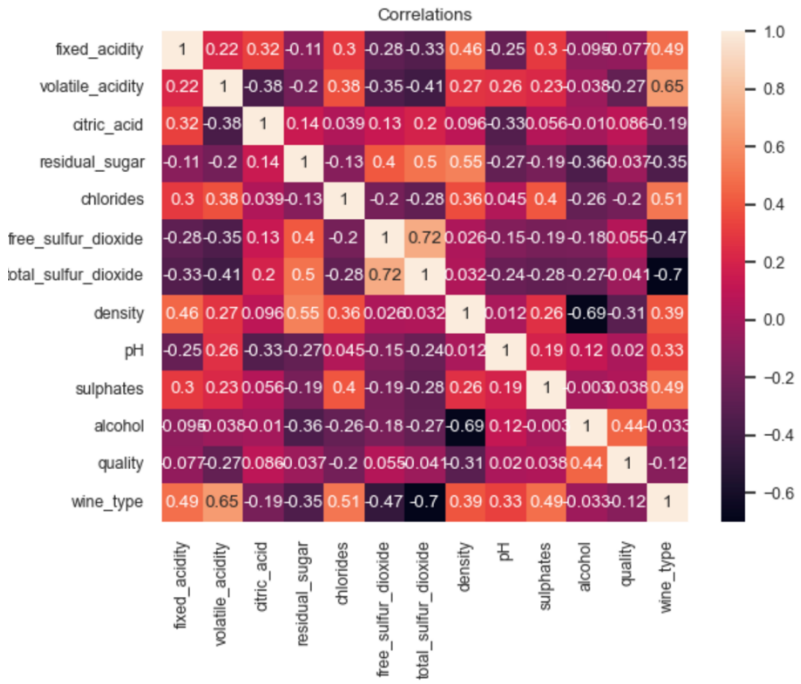


Figure 11: Correlation with wine_type

Team 2 implemented SMOTE and NearMiss techniques to balance out the data due to the higher count of white wines (white: 4,898; red: 1,599). After performing SMOTE and re-sampling the data with logistic regression analysis the accuracy skyrocketed to 98%.

	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

Figure 12: Analysis after SMOTE

Team two explored some neural network analysis for fun and found it came out with an accuracy level of 96% for one node, 97% for two, and 99% for three. Further analysis showed that specifically, “total_sulrfur_dioxide”, “volatile_acidity”, and “chlorides” were the best indicator of wine type.

	feature	importance
6	total_sulfur_dioxide	0.407177
4	chlorides	0.395523
1	volatile_acidity	0.067747
9	sulphates	0.026457
7	density	0.022694
8	pH	0.020120
0	fixed_acidity	0.019838
2	citric_acid	0.013681
10	alcohol	0.013008
3	residual_sugar	0.010134
5	free_sulfur_dioxide	0.003622

wine_type	count	mean	std	min	25%	50%	75%	max
red	1599.0	0.527821	0.179060	0.12	0.39	0.52	0.64	1.58
white	4898.0	0.278241	0.100795	0.08	0.21	0.26	0.32	1.10

Figure 13: Stats on volatile acidity and wine type

The XGBClassifier model was the best fit overall and Team 2 decided to use that technique as the final predictive model to determine wine type. It did seem like a bit of an overfit, but overall, it was the best model to use after comparing Logistic Regression (no imbalance of data techniques used), SMOTE, and NearMiss.

```
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
```

Figure 14: XGBClassifier results

Dashboard

The Dashboard was created to have an approachable way for people to explore wine and its varying features that make it either a great or bad wine. The Tableau dashboard also incorporates wine regions, prices, and ratings to make the average drinker knowledgeable on what aspects make a highly rated wine. Team 2 organized Tableau into four main dashboards.

The first dashboard focused on wine types, exploring the distinct difference with alcohol, density, and citric acid that can determine if a wine is white or red.

Figure 15: Wine type dashboard

The second dashboard provided an overview of wine quality. A user can interface with the dashboard by selecting a quality to see how alcohol, volatile acidity, pH, and sulphates effected it.

Although there were not strong correlations, some could be found with these features to determine a wine quality.

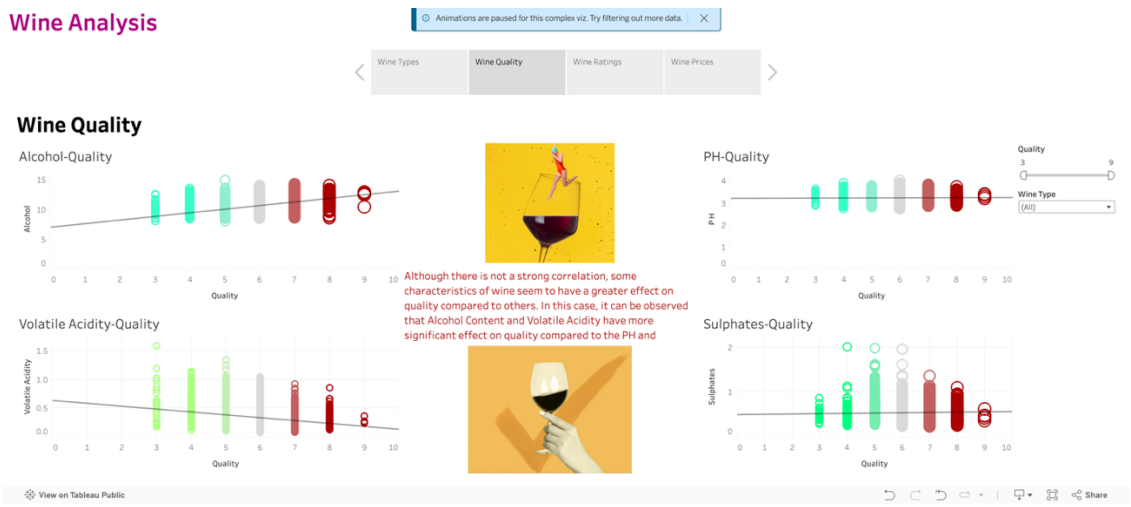


Figure 16: Wine quality dashboard

The third dashboard covered wine ratings from the third dataset. A user of the interface can view the average wine points per Country and which countries had the highest points for wine.

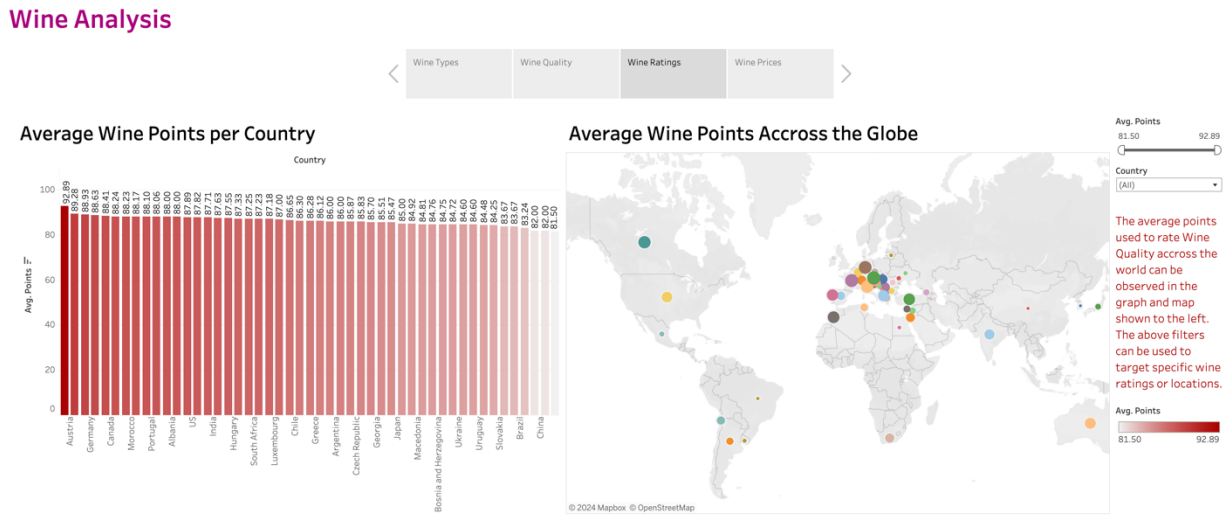


Figure 17: Wine ratings dashboard

The final dashboard displayed wine price, point ratings, and average cost of wine per country. A user can enter a price range and points to see which wine would best fit their criteria.

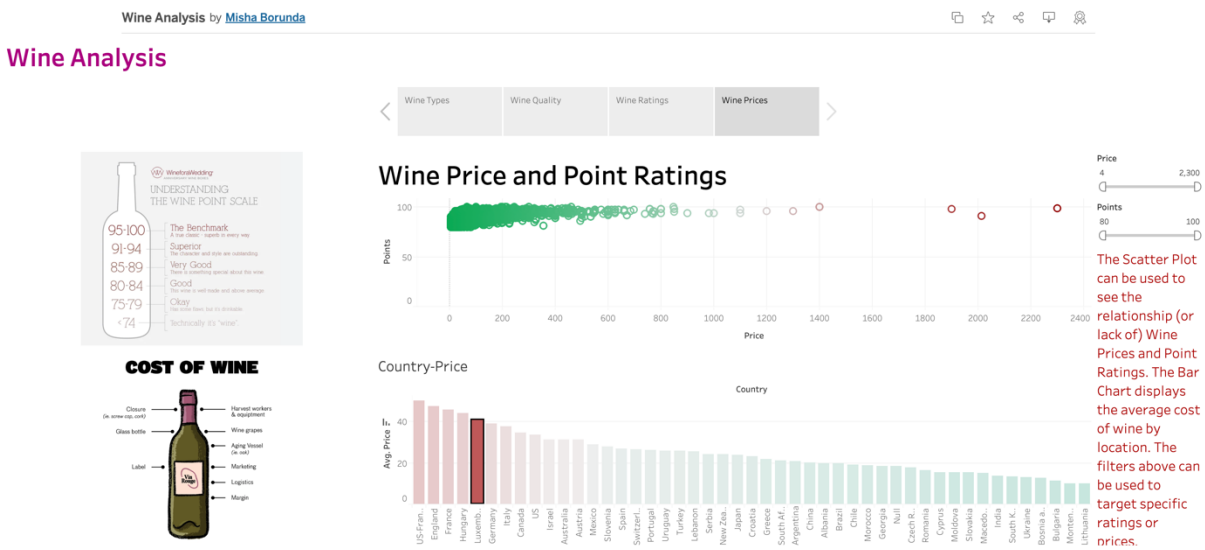


Figure 18: Wine prices dashboard

Conclusion

Overall XGBClassifier was used as the final predictive model. The logistic regression performed fine with the raw data. The team was not comfortable using it though, due to the high imbalance of red wine data points compared to white.

```
TESTING METRICS

    Test Confusion Matrix:
    [[1215   10]
 [  14  386]]

    Test Report:
                  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
```

Figure 19: Logistic regression results

Taking the imbalance into consideration, SMOTE was used to create synthetic data points to make-up the red wine data and match the white wine amount. That equated to an oversampling amount of 3,673 datapoints each. The logistic regression performed with that data came out better as seen in the figure below.

```
sm = SMOTE(random_state = 2)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train.ravel())

print('After OverSampling, the shape of train_X: {}'.format(X_train_res.shape)
print('After OverSampling, the shape of train_y: {} \n'.format(y_train_res.sha

print("After OverSampling, counts of label '1': {}".format(sum(y_train_res ==
print("After OverSampling, counts of label '0': {}".format(sum(y_train_res ==
```

```
After OverSampling, the shape of train_X: (7346, 11)
After OverSampling, the shape of train_y: (7346,)

After OverSampling, counts of label '1': 3673
After OverSampling, counts of label '0': 3673
```

```
# 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
```

Figure 20: Model using SMOTE data

The Near Miss technique was then used which deleted the white wine datapoints to match the red wine ones. The team was not comfortable with this approach due to the deletion of so many datapoints.

```
#Trying with NearMiss
print("Before Undersampling, counts of label '1': {}".format(sum(y_train == 1)))
print("Before Undersampling, counts of label '0': {} \n".format(sum(y_train == 0)))

# apply near miss
from imblearn.under_sampling import NearMiss
nr = NearMiss()

X_train_miss, y_train_miss = nr.fit_resample(X_train, y_train.ravel())

print('After Undersampling, the shape of train_X: {}'.format(X_train_miss.shape))
print('After Undersampling, the shape of train_y: {} \n'.format(y_train_miss.shape))

print("After Undersampling, counts of label '1': {}".format(sum(y_train_miss == 1)))
print("After Undersampling, counts of label '0': {}".format(sum(y_train_miss == 0)))
```

```
Before Undersampling, counts of label '1': 1199
Before Undersampling, counts of label '0': 3673

After Undersampling, the shape of train_X: (2398, 11)
After Undersampling, the shape of train_y: (2398,)

After Undersampling, counts of label '1': 1199
After Undersampling, counts of label '0': 1199
```

```
# 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       0.99      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
```

Figure 22: Model using Near Miss data

Finally, looking at the XGBClassifier utilizing the SMOTE sampled dataset, it came out as the best model for predicting wine types.

```
TESTING METRICS

    Test Confusion Matrix:
    [[1220    5]
 [   1  399]]

    Test Report:
                  precision    recall  f1-score   support

           0       1.00      1.00      1.00      1225
           1       0.99      1.00      0.99       400

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

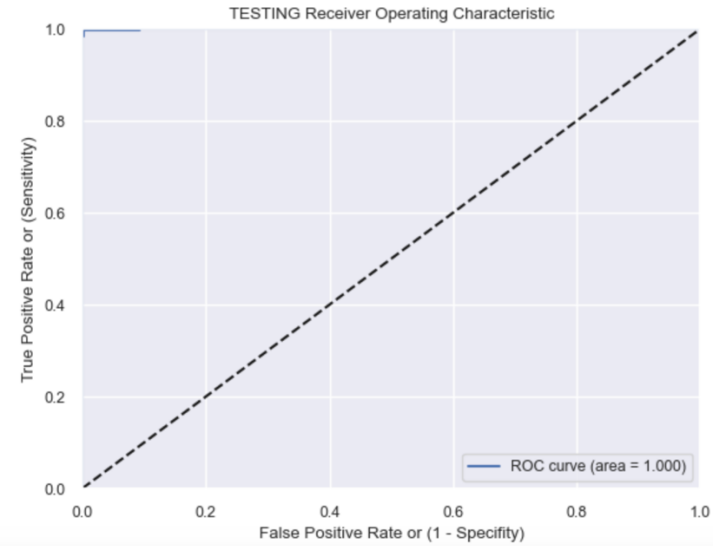


Figure 23: XGBClassifier model using SMOTE data

In conclusion, team 2 found the best predictive module and created a dashboard interface to guide the user on understanding what aspects effect wine. Naturally this study is intriguing because wine is within the top 10 beverages consumed globally. This study educated the average drinker on the various features of wine and how they can be used to predict wine type, price, and point ratings. Not every wine has to be super expensive to be a great drink and a multitude of aspects come together to create a perfect bottle of vino.

Works Cited

Bootswatch:

https://bootswatch.com/flatly/

Prediction of quality of Wine

https://www.kaggle.com/code/vishalyo990/prediction-of-quality-of-wine

Red Wine Quality:

https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009?resource=download

White Wine Quality:

https://www.kaggle.com/datasets/piyushagni5/white-wine-quality

Wine Reviews:

https://www.kaggle.com/datasets/zynicide/wine-reviews

W3:

https://www.w3schools.com/html/default.asp

Other Resources:

SMU Data Analytics Boot Camp Lessons

hhttps://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.

Terminology

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 seacoast.

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: Chemical found in beverages such as wine, liquor, and beer.