



WINE ANALYSIS PRESENTATION

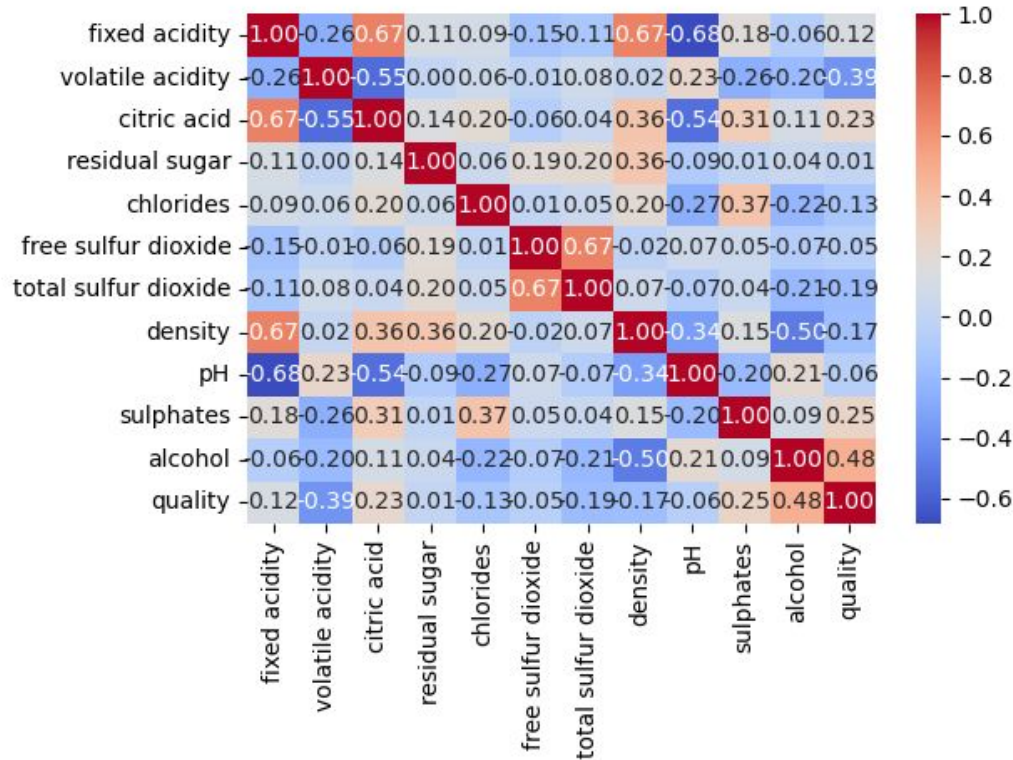
By: Justine Pile
Nico Barzotti
Dykie Smith

WINE VARIABLES

- The purpose of this analysis was to use a machine learning model to predict the quality rating of a wine.
- Several variables are listed in this dataset and we set out to see which variables have a positive or negative relationship to the quality rating that a wine receives.

	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

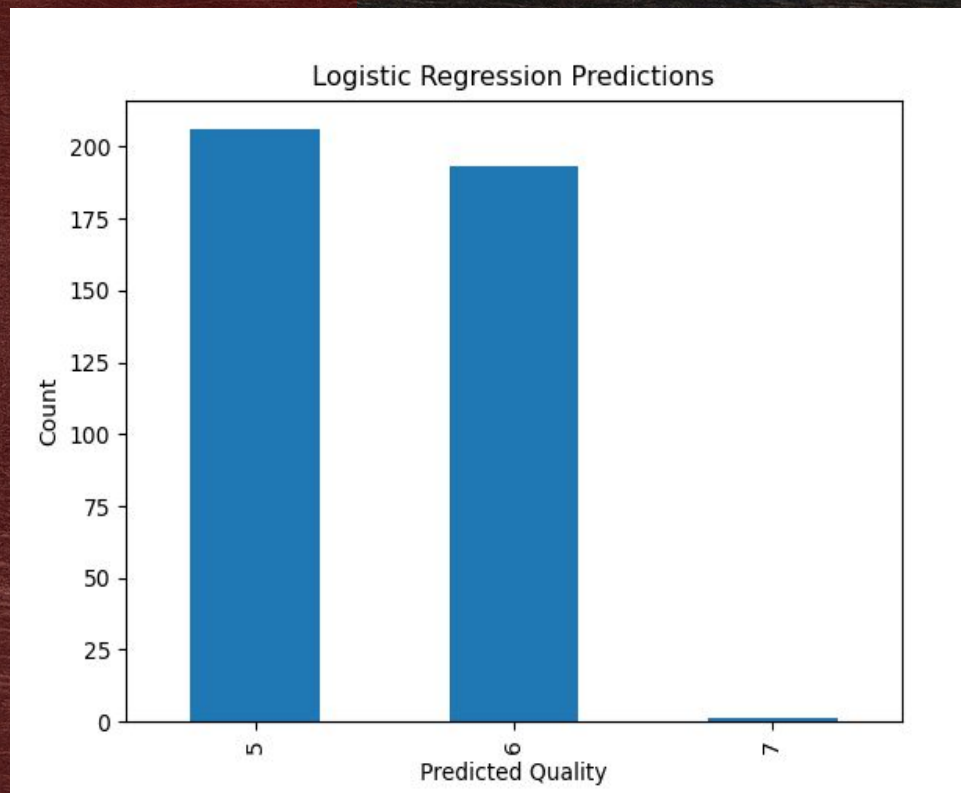
HEATMAP



- There is a negative correlation between alcohol and volatile acidity, as well as between volatile acidity and citric acid
- There is a positive correlation between alcohol and quality

LOGISTIC REGRESSION MODELING

- A logistic regression model was used to predict whether the wine quality would be above or below a rating of 6
- The accuracy of this model was 58%.
- 58% of the predictions made by the logistics regression model were correct.



LOGISTIC REGRESSION MODELING

- The binary accuracy of this model was 75%.
- 75% of the testing data was correctly classified while the remainder was not. This produced a better result than the first accuracy model.

	Prediction	Actual
551	1	1
1413	0	0
1090	1	1
1369	0	0
536	0	0
...
624	0	0
1532	1	1
1073	1	1
839	0	0
40	0	0

K-NEAREST NEIGHBORS

KNN model with quality bins for 3-5 and 6-8

```
q = []
for i in wine['quality']:
    if i in (3, 4, 5):
        q.append('0')
    elif i in (6, 7, 8):
        q.append('1')
wine['quality_bins'] = q

wine['quality_bins'].value_counts()

1      855
0      744
Name: quality_bins, dtype: int64

x=wine.drop('quality_bins', axis=1)
y=wine['quality_bins']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 2)

sc = StandardScaler()

x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)

# Create a KNN model to predict quality from 'quality_bins'
knn_classifier = KNeighborsClassifier()

# Fit the model using training data
knn_classifier.fit(x_train, y_train)

KNeighborsClassifier
KNeighborsClassifier()
```

```
# Make prediction using the wine testing data
quality_predictions_KNN = knn_classifier.predict(x_test)
knn_df = pd.DataFrame({"Prediction": quality_predictions_KNN, "Actual": y_test})
knn_df
```

	Prediction	Actual
407	1	1
1220	1	1
1200	1	1
308	1	1
1328	0	0
...
724	0	0
188	0	0
1374	0	0
788	1	1
770	1	1

320 rows × 2 columns

```
# Analyze the accuracy of the predictions
knn_accuracy = knn_classifier.score(x_test, y_test)
print("KNN Accuracy:", knn_accuracy)
```

```
knn_df.to_csv('knn_model.csv', index=False)
```

KNN Accuracy: 0.959375


Key insights from the summary statistics include:

1. **Fixed Acidity:** Ranges from 4.6 to 15.9 with an average of 8.32.
2. **Volatile Acidity:** Ranges from 0.12 to 1.58 with an average of 0.53.
3. **Citric Acid:** Ranges from 0 to 1 with an average of 0.27.
4. **Residual Sugar:** Ranges from 0.9 to 15.5 with an average of 2.54.
5. **Chlorides:** Ranges from 0.012 to 0.611 with an average of 0.087.
6. **Free Sulfur Dioxide:** Ranges from 1 to 72 with an average of 15.87.
7. **Total Sulfur Dioxide:** Ranges from 6 to 289 with an average of 46.47.
8. **Density:** Ranges from 0.99007 to 1.00369 with an average of 0.9967.
9. **pH:** Ranges from 2.74 to 4.01 with an average of 3.31.
10. **Sulphates:** Ranges from 0.33 to 2 with an average of 0.66.
11. **Alcohol:** Ranges from 8.4 to 14.9 with an average of 10.42.
12. **Quality:** Ranges from 3 to 8 with an average of 5.64.

The standard deviation values indicate the degree of variation in the data. For example, the quality rating has a standard deviation of 0.81, indicating a relatively low level of variability in the wine quality ratings in this dataset.

This summary provides a comprehensive overview of the dataset and can be used to inform data preprocessing steps and guide the development of a machine learning model to predict wine quality.

The provided data is a statistical summary of a dataset related to wine quality. It includes 1599 entries, with no missing values, and 12 variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality.

A bottle of red wine and two glasses of red wine are positioned on a dark, textured wooden surface. The bottle is on the right, and the glasses are in the foreground, slightly to the left of the bottle. The background is a dark, textured wall. The text is overlaid on a semi-transparent dark red rectangular area.

This analysis aimed to use a machine learning model to predict the quality rating of red wine based on various characteristics or properties. The dataset used contained several variables, such as acidity, sugar content, alcohol percentage, Sulphates, and pH. The goal was to identify how these variables relate to the quality rating, whether positively (increasing the variable increases the quality) or negatively (increasing the variable decreases the quality).

For instance, a positive relationship between alcohol percentage and quality rating would suggest wines with higher alcohol content tend to have higher quality ratings. Conversely, a negative relationship between acidity and quality rating would suggest wines with higher acidity tend to have lower quality ratings.

The goal was to build a machine learning model that could accurately predict a red wine's quality rating based on these variables. Such a model could be beneficial for wine producers, allowing them to adjust their production processes to enhance wine quality. It could also be advantageous for consumers, providing them with a tool to select wines likely to be of high quality based on measurable characteristics.

Thank you.
Questions?

