Red Wine Quality

Presented by

Faymin Chen

Introduction:

The wine industry is highly competitive and saturated with many different varieties and blends for the consumer to choose from. As such, many wine rating websites have popped up such as Wine Enthusiast and Vivino, which help customers separate the gems from the chaff. For a winery, it's important for them to invest in wines that are going to get high ratings so that they can recoup their investments.

Objective:

This project will use Red Wine Quality Data Set, available on the UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/wine+quality/winequality-red.csv to perform binary classification for wine quality using ML Algorithm.

The goal of this project is to determine wine quality based on its chemical properties.

It is important especially for small brewery to correctly identified high quality wine from the rest.

We want to minimize incorrectly classify low quality wine as high quality.

We wanted a model that perform better than average so the wine breweries can stay ahead of competition. It will help the executive, directors, managers to understand their business and change strategy, make decision using a data driven approach.

We summarized our findings and insights gained from analyzing the red wine dataset, and evaluate model performance of both Logistic Regression and Random Forest Classifier. Finally, we choose the best model for our wine quality prediction

Methodology:

Load the Dataset: The Red Wine Quality dataset is loaded using the pd.read_csv() function. The head() and info() methos are used to display the first few rows and get information about the dataset respectively.
Knowing the Dataset: Basic information about the dataset is generated; there are no categorical attributes, they are all numeric.
Data Cleaning: no duplicate values found, and no missing values
Data Visualization: Seaborn library is used to visualize the data
Data Preprocessing: The target variable 'Quality' is further categorized and renamed to 'hi_quality'. Red wines with quality score higher than 6 will be classified as hi_quality = 1, score >=6 will be classified as hi_quality = 0 since we are performing binary classification. We scaled our features using Standard Scaler since the data has varying scales.
- **Splitting the Dataset**: The dataset is split into training and testing sets using the train_test_split() method from scikit-learn.
Implementing Machine Learning Algorithms: Logistic Regression and Random Forest classifiers are initialized and trained using the training data.
Model Evaluation: AUC ROC score and confusion matrix (precision, recall, F1 score) are computed to evaluate the performance of each algorithm on the testing data. We also performed threshold adjustment which is preferred method over undersampling or oversampling when dealing with imbalanced data
Results: The results, including the AUC ROC score and confusion matrix, are printed for each algorithm.

Data Overview:

The dataset related to red wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests (see [Cortez et al., 2009], http://www3.dsi.uminho.pt/pcortez/wine/).

Additional Information

The dataset is related to red variants of the Portuguese "Vinho Verde"* wine. For more details, consult: http://www.vinhoverde.pt/en/ or the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.). These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

*Portuguese wine that originated in the historic Minho province in the far north of the country

Features:

- Fixed acidity
- Volatile acidity
- Citric acid
- Residual sugar
- Chlorides
- Free sulfur dioxide
- Total sulfur dioxide
- Density
- PH
- Sulphates
- Alcohol

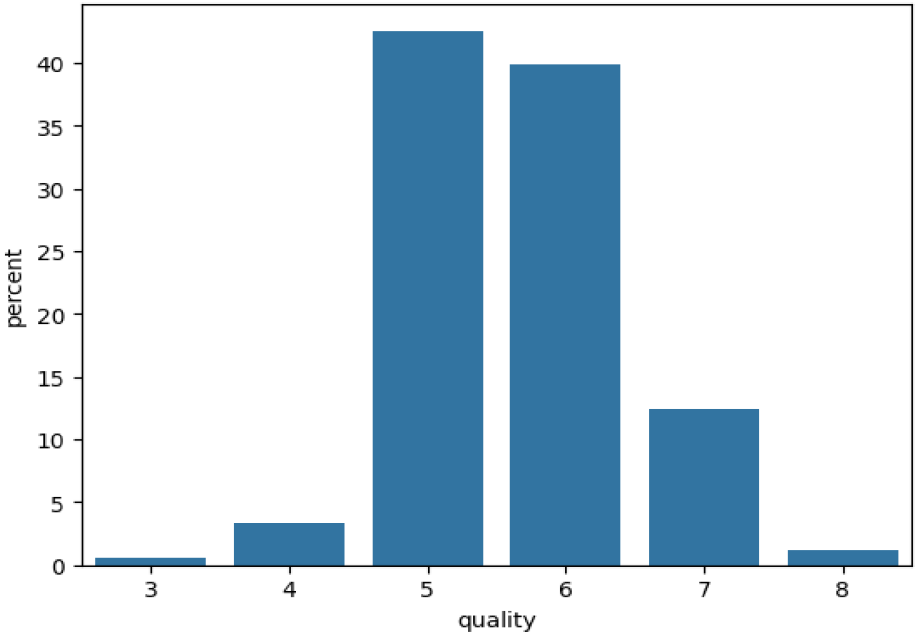
```
wine.isnull().sum()
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
```

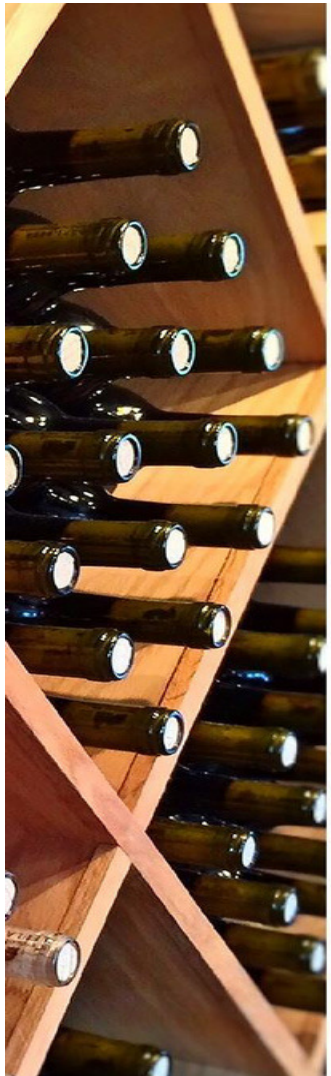
We have clean dataset, no duplicates and no missing values

Label Distribution

- Quality is represented by scores ranging from 0 to 10
- 0 is the worst and 10 is the best

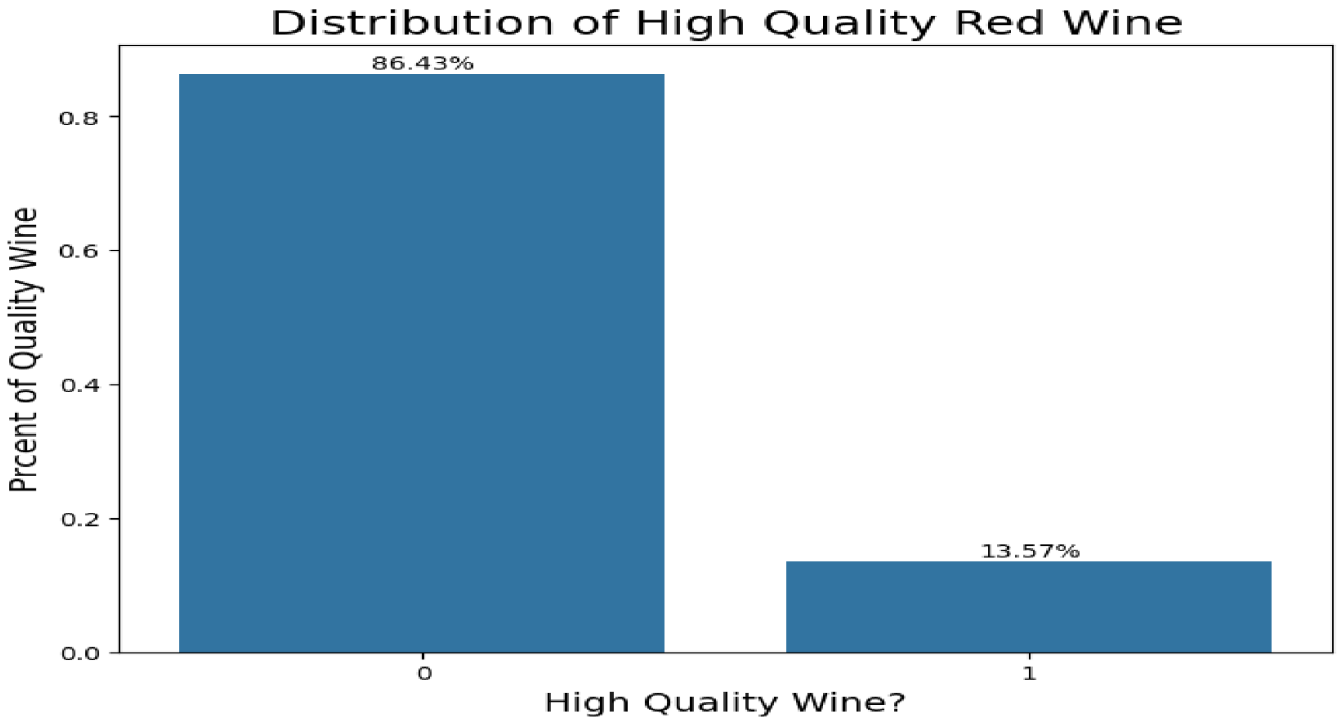


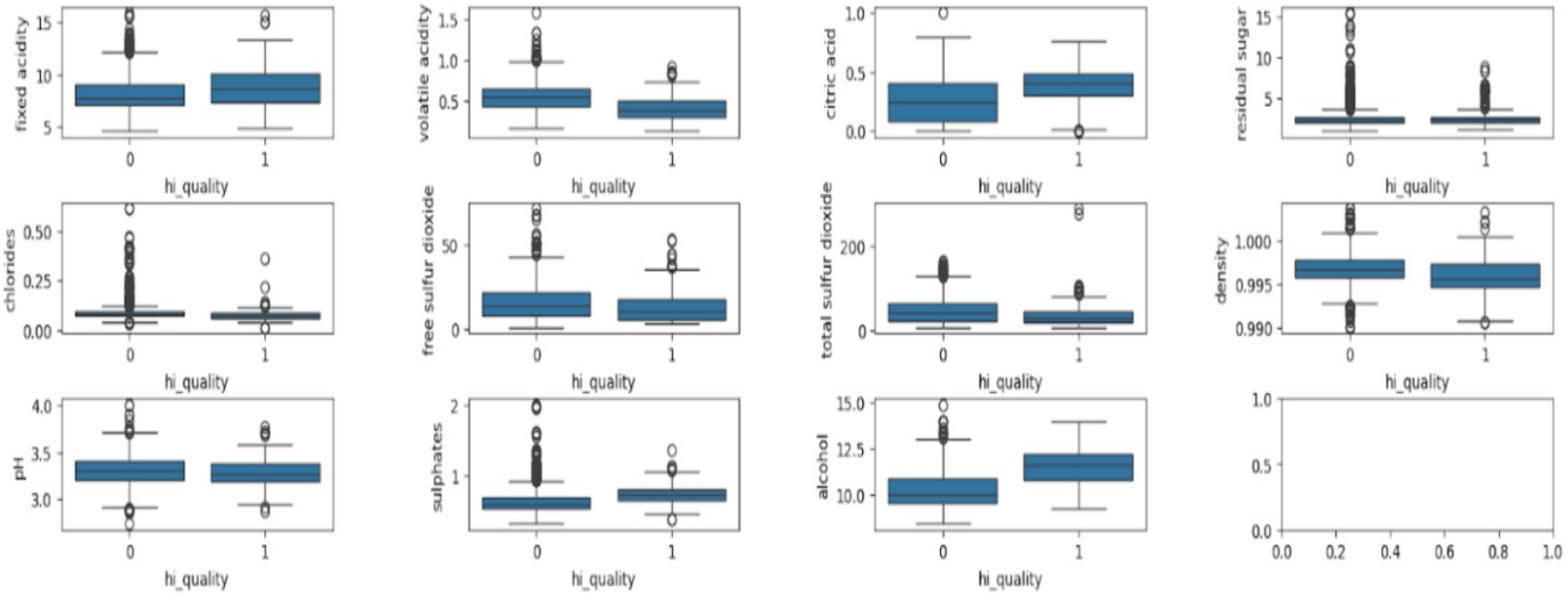
The dataset consists of mostly mediocre wine quality



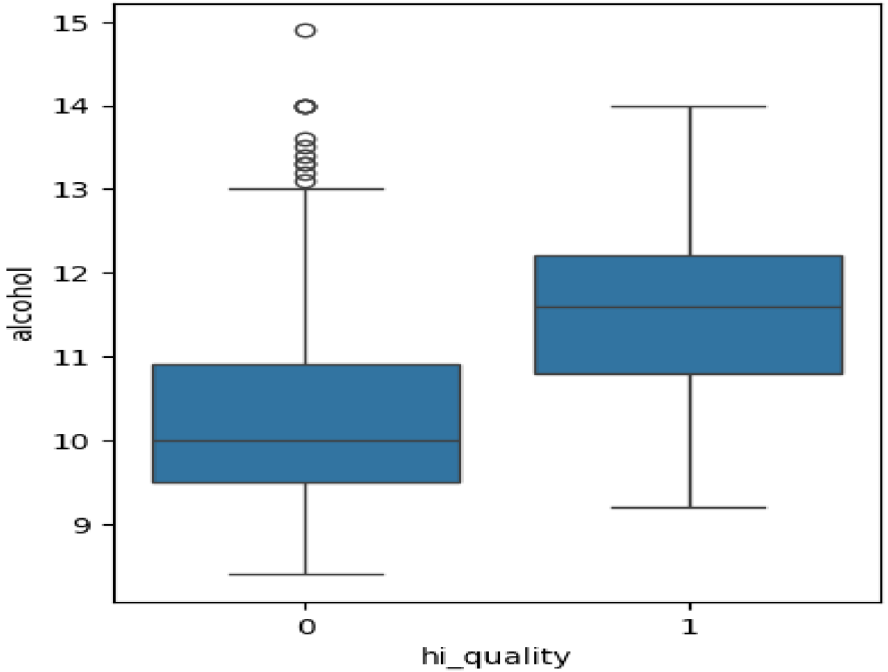
Labels and Encoding

- Binning:
  - score under 6 → “Low quality
  - Score above 6 → High Quality



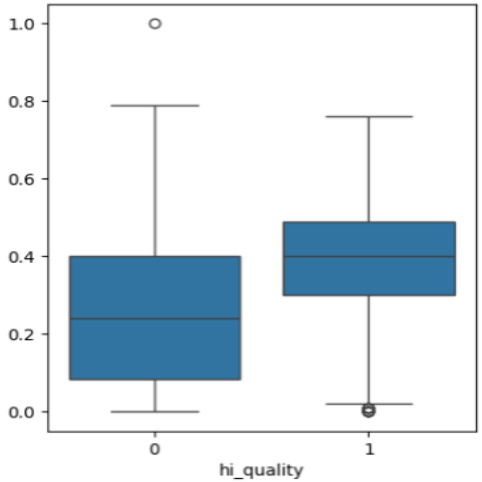

fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
hi_quality

hi_quality vs alcohol



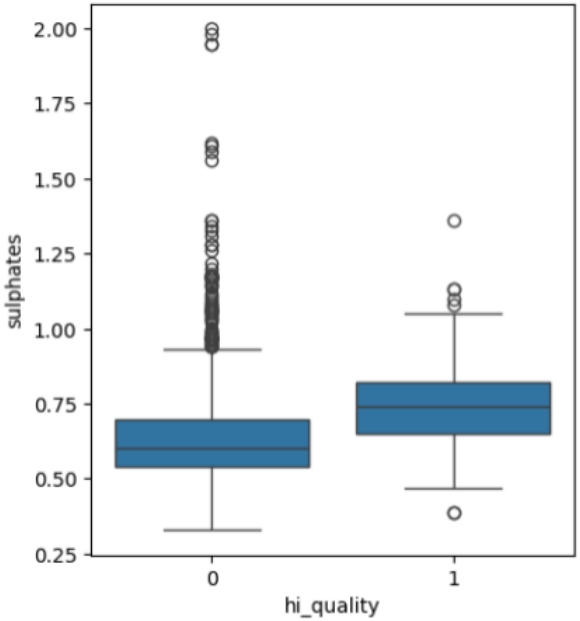
High quality wines have about 1.5% more alcohol on average than low quality wines. While it's unclear exactly why this might be the case, it may simply be due to general taste preferences of the raters lean more towards wine that is alcoholic. This was shown to be statistically significant via a t-test (p-value < .001)

hi_quality vs citric acid



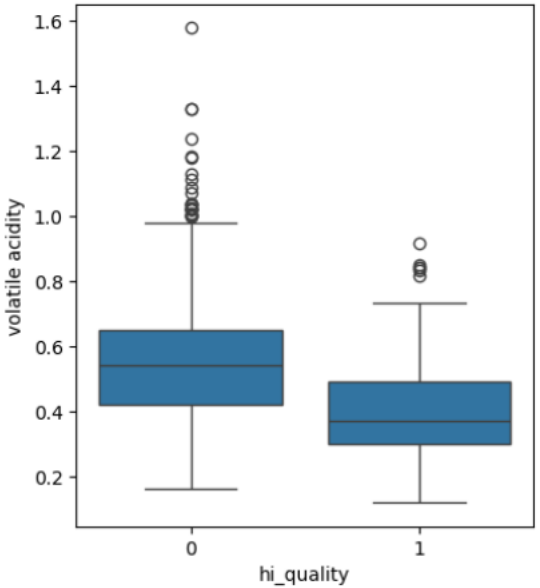
Higher quality wines tend to have more citric acid. Citric acid can give the wine a "fresh" taste, enhancing its flavor. Citric acid is often used as a stabilizer in food and beverages. This relationship was shown to be statistically significant via a t-test (p-value < 0.05)

hi_quality vs sulphates



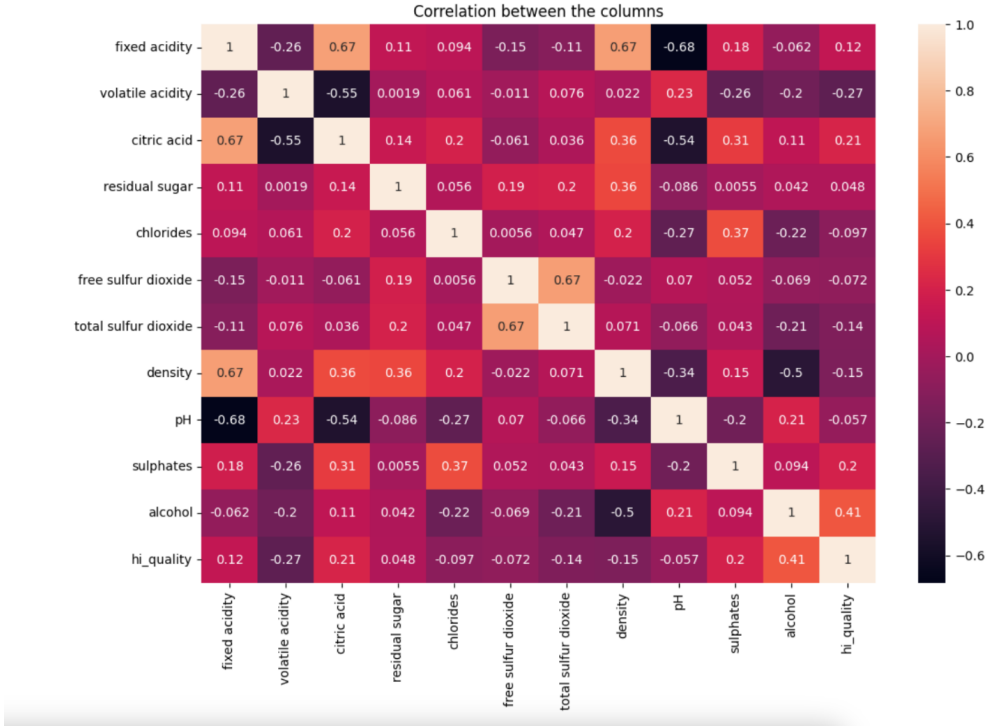
High quality wines have more sulphates than low quality wine. This relationship was shown to be statistically significant via a t-test (p-value < 0.05)

hi_quality vs volatile acidity



There is an inverse relationship between volatile acidity and wine quality: higher quality wine tends to have a lower level of volatile acidity. Volatile acids, similar to acetic acid, give a sour taste to our wine, degrading its quality. This relationship was shown to be statistically significant via a t-test (p-value < 0.01)

Variable Correlation Matrix



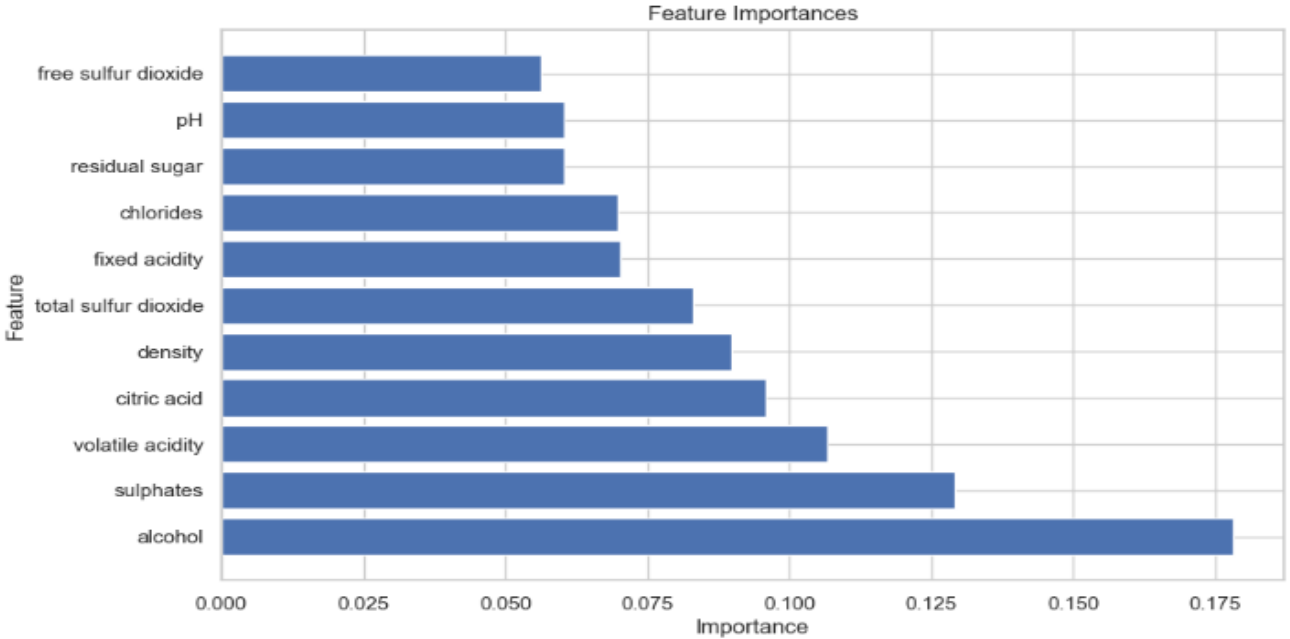
Variable Correlation Matrix

The heatmap shows that pH and volatile acidity have inverse correlation with hi_quality, as well as chlorides, density,total sulfur dioxide and free sulfur dioxide.

Multicollinearity: There is a strong positive correlation between fixed acidity and density (0.67). Also a strong positive correlation between total sulfur dioxide and free sulfur dioxide (0.67)
We will keep an eye on this going forward but for now will not drop any columns from the dataset

Some columns like citric acid, alcohol, sulphates are strongly and positively correlated whereas a lot of columns like pH, volatile acidity,chlorides, density,total sulfur dioxide and free sulfur dioxide have a negative correlation.
Highest correlation: alcohol ; Smallest correlation: residual sugar

Features Importance from Random Forest



The top 10 most important feature according to Random Forest Classifier:
Alcohol, sulphates, volatile acidity, citric acid, density, total sulfur dioxide, fixed acidity, chlorides, residual sugar, pH, free sulfur dioxide

Modeling Process

Split data into train and test set and perform scaling

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train2 = sc.fit_transform(X_train)
X_test2 = sc.transform(X_test)
```

Logistic Regression Model

```
lr=LogisticRegression(solver='liblinear').fit(X_train2, y_train)
# Predicting Test Set
pred_test = lr.predict(X_test2) from sklearn.metrics import confusion_matrix, accuracy_score,
f1_score, precision_score, recall_score
test_accuracy = accuracy_score(y_test, pred_test)
test_cf_matrix=confusion_matrix(y_test, pred_test)
test_cl_report=classification_report(y_test,pred_test)
```

Logistic Regression Model

A logistic regression model is used for predicting classes using the probability of the target variable. Unlike linear regression, which uses expected values of the response model, logistic regression uses the probability or odds of the response variable to model based on the combination of values taken by the predictors. This model uses the sigmoid function that maps predicted values to probabilities.
It works well on linearly separable classes with easy implementation, making it a popular choice for classification problems.

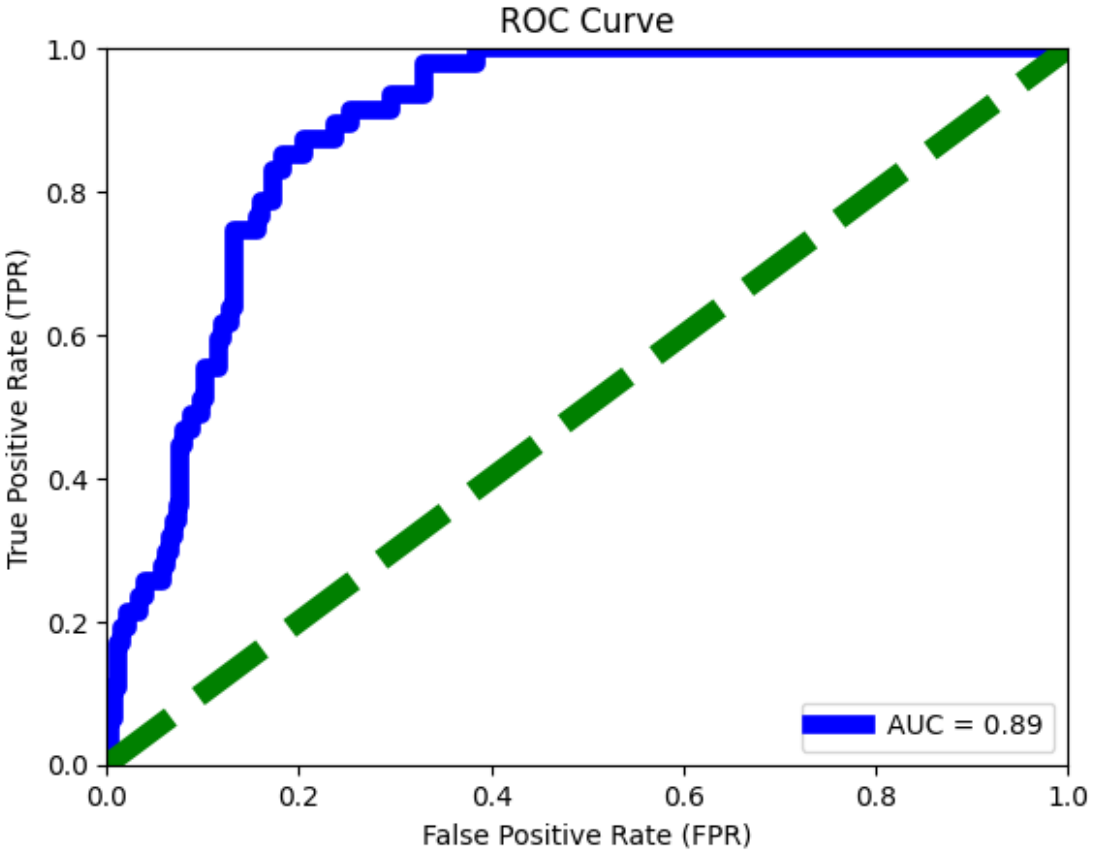
There are two types of logistic regression models for classification: binary and multinomial.
Binary logistic regression requires a dependent variable with only two possible outcomes whereas a multinomial requires three or more outcomes. In this case, the dataset is working with binary logistic regression since the target variable is binary (1 or 0). Logistic regression is applicable to this problem since we want to predict the probabilities and classify the red wine quality into two categories based on the explanatory variables. For the solver in the logistic regression model, liblinear is picked since it supports both L1 and L2 regularization.
We first tried to build a logistic regression model, which is specifically designed for binary classification problems and is the most straightforward model in our case. And we also

performed hyperparameter tuning via GridSearchCV to improve the result of our model. In addition, we adjusted the threshold to achieve a higher F1 score.

Logistic Regression Model

```
Best Parameters for Logistic Regression: {'C': 0.01, 'penalty': 'l2'}

Accuracy Tuned for Logistic Regression: 0.846875

Confusion Matrix Tuned for Logistic Regression:
 [[245  28]
 [ 21  26]]

Classification Report Tuned for Logistic Regression:
              precision    recall  f1-score   support

           0       0.92      0.90      0.91       273
           1       0.48      0.55      0.51        47

    accuracy                           0.85       320
   macro avg       0.70      0.73      0.71       320
weighted avg       0.86      0.85      0.85       320
```

Logistic Regression Model



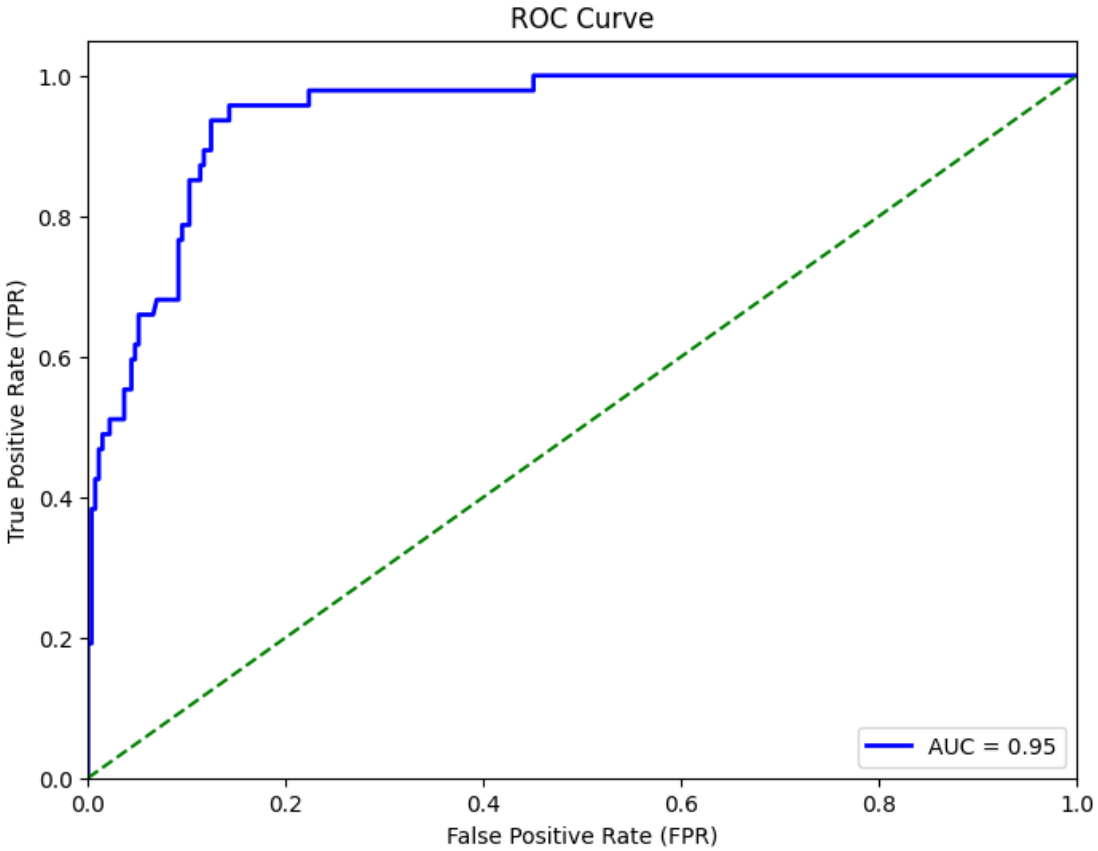
Random Forest Model

```
RF_model=RandomForestClassifier(n_estimators=100, bootstrap=False)
RF_model.fit(X_train2,y_train) y_test_pred=RF_model.predict(X_test2)
y_train_pred=RF_model.predict(X_train2)
train_accuracy=accuracy_score(y_train,y_train_pred)
train_cf_matrix=confusion_matrix(y_train,y_train_pred)
train_cl_report=classification_report(y_train,y_train_pred)
test_accuracy=accuracy_score(y_test,y_test_pred)
test_cf_matrix=confusion_matrix(y_test,y_test_pred)
test_cl_report=classification_report(y_test,y_test_pred)
```

Random Forest Model

```
Fitting 3 folds for each of 81 candidates, totalling 243 fits

Best Parameters (Random Forest): {'max_depth': 15, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n
_estimators': 500}
Accuracy (Tuned Random Forest): 0.89375
Confusion Matrix (Tuned Random Forest):
 [[255  18]
 [ 16  31]]
Classification Report (Tuned Random Forest):
              precision    recall  f1-score   support

           0       0.94      0.93      0.94       273
           1       0.63      0.66      0.65        47

    accuracy                           0.89       320
   macro avg       0.79      0.80      0.79       320
weighted avg       0.90      0.89      0.89       320
```

Random Forest Model



Conclusion

Our Random Forest model perform much better than the Logistic Regression model, suggesting there is a complex, non-linear relationship between the features and hi quality target variable.

Discussion

When assessing wine quality, precision is generally considered more important than recall because it's crucial to accurately identify high-quality wines, meaning a false positive (classifying a low-quality wine as high quality) is more detrimental than a false negative (missing a truly high-quality wine) in most scenarios.
Impact of False Positives: If a model incorrectly labels a low-quality wine as high quality, it could mislead consumers and damage the reputation of a producer, which is a more significant concern than missing a few good wines.
Focus on Quality Assurance: Wine quality assessment aims to ensure that the wines being marketed as high quality truly meet the standard, so prioritizing precision aligns with this goal.
However, context matters:

- Market Analysis : If the goal is to identify all potential high-quality wines in a large market, even at the cost of including some lower-quality options, then recall might be more important.
- Specific Quality Levels: Depending on the quality tier, the importance of precision might vary.

Key Takeaway: For most wine quality assessments, prioritizing precision is generally considered better practice as it minimizes the risk of misrepresenting lower-quality wines as high quality.