Wine Quality Prediction

Introduction

Problem Statement

Wine is the fermented juice of grapes, made in many varieties, such as red, white, sweet, dry, still, and sparkling, for use as a beverage, in cooking, in religious rites, etc., and usually having an alcoholic content of 14 percent or less. The wine industry shows a recent exponential growth as social drinking is on the rise. Nowadays, industry players are using product quality certifications to promote their products. This is a time-consuming process and requires the assessment given by human experts, which makes this process very expensive. Also, the price of wine depends on a rather abstract concept of wine appreciation by wine tasters, opinion among whom may have a high degree of variability. Another vital factor in wine certification and quality assessment is physicochemical tests, which are laboratory-based and consider factors like acidity, pH level, sugar, and other chemical properties. The wine market would be of interest if the human quality of tasting can be related to wine's chemical properties so that certification and quality assessment and assurance processes are more controlled. This project aims to determine which features are the best quality wine indicators and generate insights into each of these factors to our model's wine quality.

Technical Approach

The Cross-Industry Standard Process for Data Mining (CRISP-DM). This process model has 6 phases that naturally describe the data science life cycle for this project.

1. Business Understanding
2. Data Understanding

3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

During every phase of this project lifecycle, we might discover new aspects/finding which we will incorporate them in ways to improve the efficiency of our model.

Data Sources

The dataset that is selected has all the details that are needed for a model to predict the wine quality

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

Analysis

We are planning to perform feature reduction and dimensionality reduction to select the most relevant variables for our model. Many predictive algorithms assume the model variables follow a normal distribution. There are inherent advantages to using normally distributed variables, so our approach will focus on columns that closely follow this distribution. We will determine which variables are normally distributed by conducting the following analyses:

- Summary statistics on all variables: concentrating on mean and SD values.
- Identify the best model.

Requirement Development

For this project, we are planning to use python. To complete this project in python the following are required for our development:

IDE: Jupyter Notebook

Libraries:

1. Numpy – extensively used for data analysis to handle multidimensional arrays.
2. Pandas – high-performance data structures and analysis tools for the labeled data.
3. Matplotlib –powerful visualizations which create several stories with the data visualized.
4. SciPy – used for high-level technical computations.
5. Seaborn – interface for drawing attractive and informative statistical graphics.

Model Deployment

We will use feature selection techniques to finalize our feature list for models. Using the results from this step, we will build a couple of classification models and evaluate their performance. Below are some example models.

- **Random forest:** It builds multiple decision trees and merges them to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.
- **Decision Tree:** It belongs to the family of supervised learning algorithms. It can be used for solving regression and classification problems. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from training data. In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. Based on the comparison, we follow the branch corresponding to that value and jump to the next node.

Testing and Evaluation

We will be splitting the data into 70% training and 30% test dataset. Using the test dataset, we will test the model. confusion matrix, AUC, and F1 scores will be used for the evaluation. cross-validation will also be used to decide the best model.

Expected Results

Using this model we can predict the quality of wine.

Ethical Considerations

Ethical considerations in research are a set of principles that guide our research designs and practices. We must always adhere to a certain code of conduct when collecting data from people. The goals of human research often include understanding real-life phenomena, studying effective treatments, investigating behaviors, and improving lives in other ways. What you decide to research and how you conduct that research involve key ethical considerations.

These considerations work to

- Protect the rights of research participants
- Enhance research validity
- Maintain scientific integrity
- Since this is something people use for celebrations we have to be careful in manufacturing the wine without compromising on the quality.
- Money should not matter when compared with the lives of the people.

Challenges/Issues

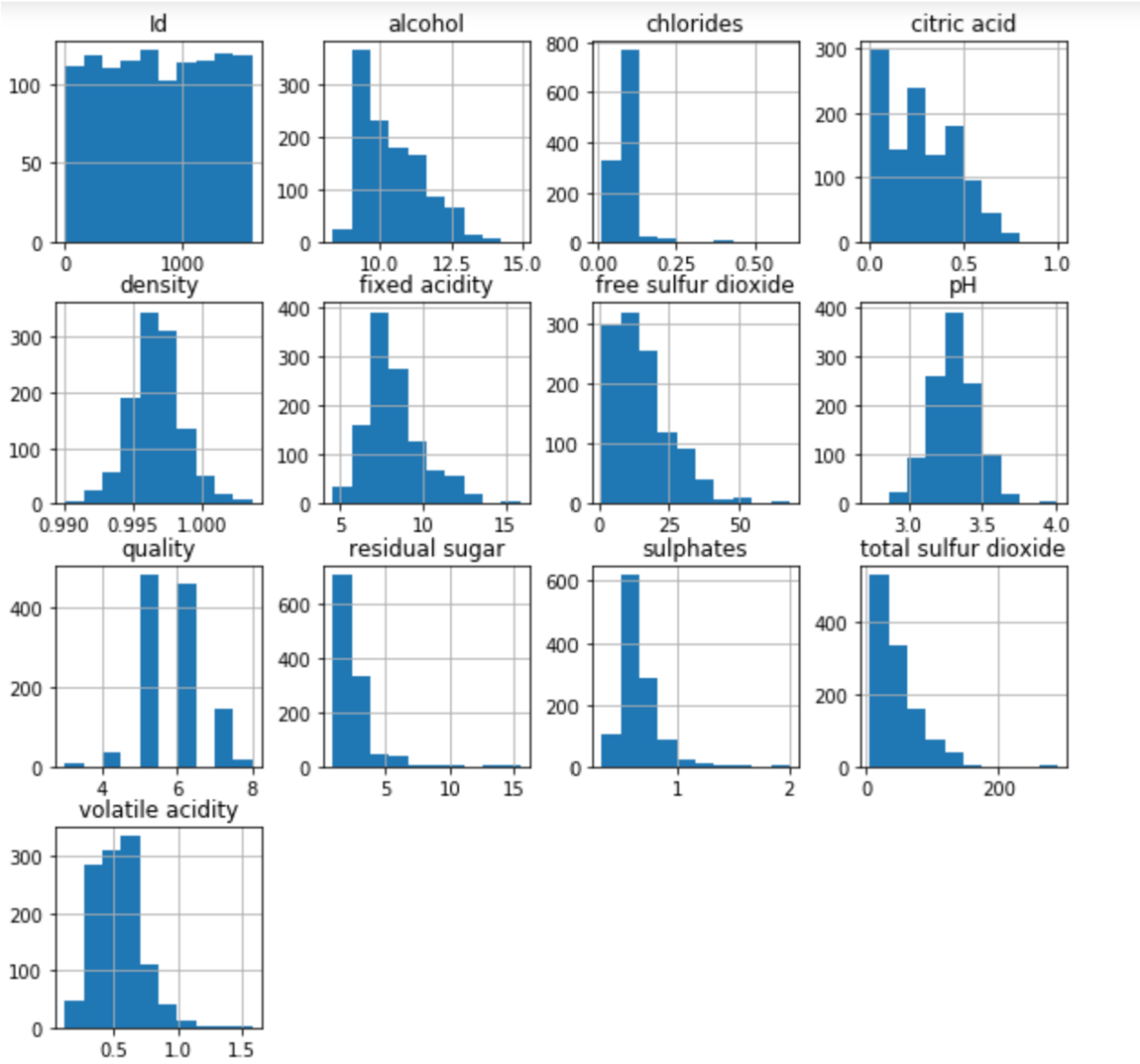
The biggest challenge is remove the features that are not required to improve the efficiency of the model.

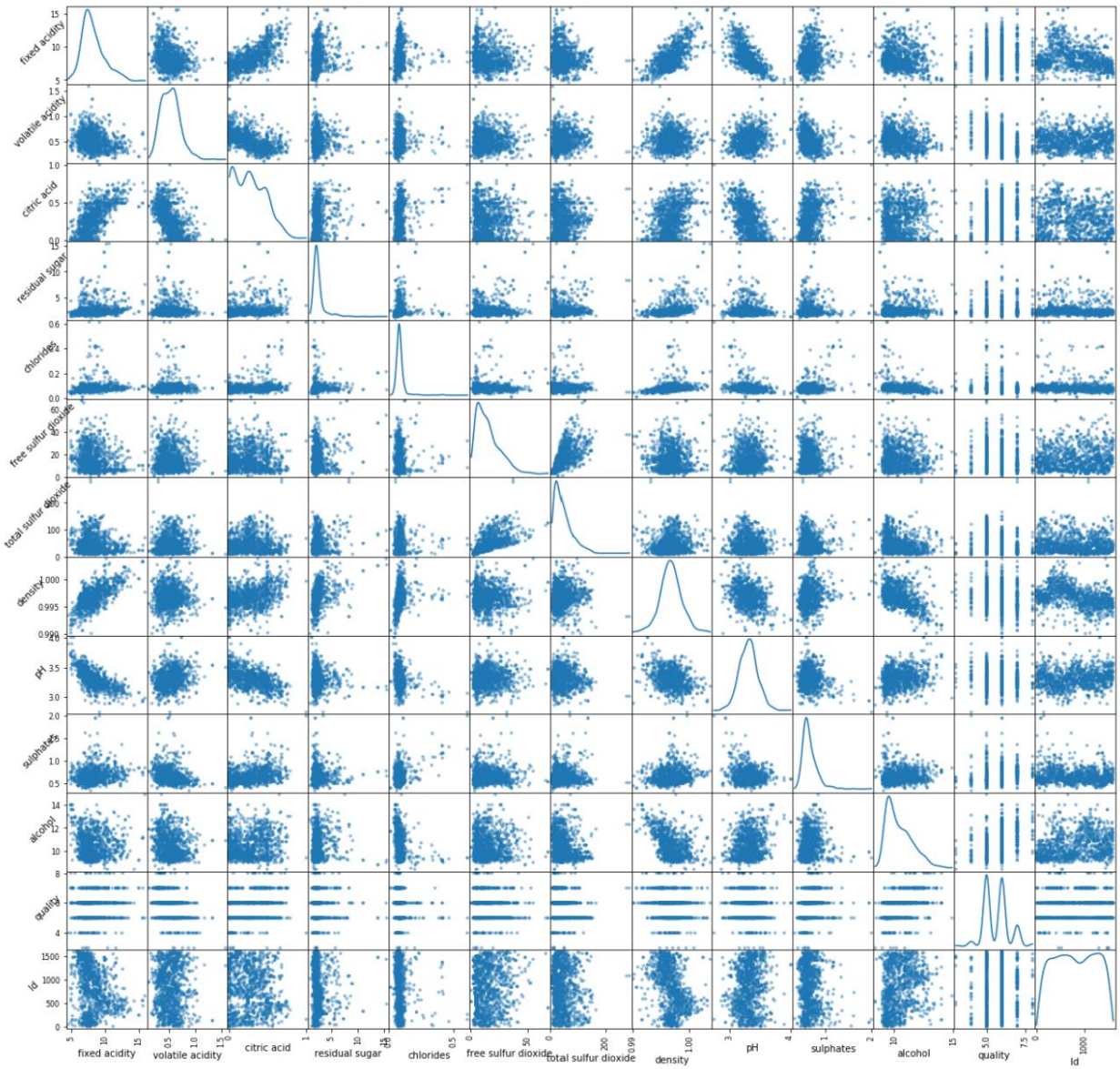
Reference:

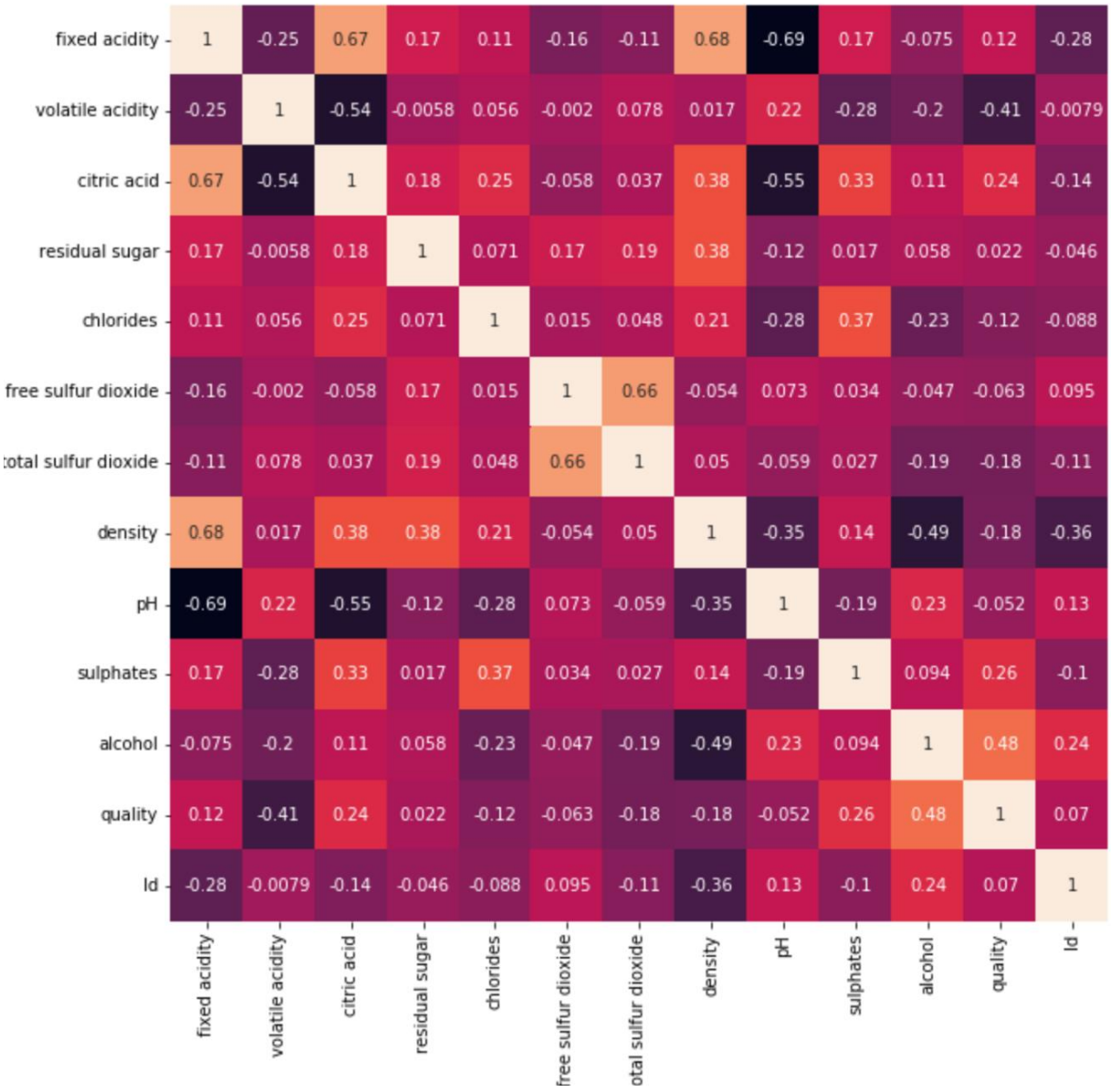
- https://www.kaggle.com/yasserh/wine-quality-dataset

- https://towardsdatascience.com/red-wine-quality-prediction-using-regression-modeling-and-machine-learning-7a3e2c3e1f46
- https://www.scirp.org/journal/paperinformation.aspx?paperid=107796

Illustration:




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



We have loaded the Wine Quality dataset into a dataframe

```
# Load data into a dataframe
wine_df = pd.read_csv("WineQT.csv")
wine_df.head(10)
```

]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	0
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
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	2
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	3
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	4
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5	5
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	5	6
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	7	7
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	7	8
9	6.7	0.58	0.08	1.8	0.097	15.0	65.0	0.9959	3.28	0.54	9.2	5	10

This provides us the information about the columns that we have in the dataset.

```
wine_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1143 non-null   float64
 1   volatile acidity      1143 non-null   float64
 2   citric acid           1143 non-null   float64
 3   residual sugar        1143 non-null   float64
 4   chlorides             1143 non-null   float64
 5   free sulfur dioxide   1143 non-null   float64
 6   total sulfur dioxide  1143 non-null   float64
 7   density               1143 non-null   float64
 8   pH                    1143 non-null   float64
 9   sulphates             1143 non-null   float64
 10  alcohol               1143 non-null   float64
 11  quality               1143 non-null   int64  
 12  Id                    1143 non-null   int64  
dtypes: float64(11), int64(2)
memory usage: 116.2 KB
```

We have to check if there are any null values in the datasets at column level. This confirms us that we do not have null values in the dataset

```
# Check if any missing values
wine_df.isnull().sum()
```

```
5]: 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
    Id                      0
    dtype: int64
```

Since our target variable is quality, we are dropping it from the wine_df dataframe.

```
9]: y = wine_df['quality']
    x = wine_df.drop('quality', axis = 1)
```

We used selectkbest to identify the best feature from the dataset which can be used to predict the target variable quality.

```
# Feature selection using SelectKBest feature selection
skbest = SelectKBest(k=12)
skbest.fit(x,y)
x_skbest=skbest.transform(x)
x_skbest.shape
```

```
]: (1143, 12)
```

```
# 10 best features using SelectKBest
best_features = SelectKBest(score_func=f_classif, k=10)
fit = best_features.fit(x,y)
df_scores = pd.DataFrame(fit.scores_)
df_columns = pd.DataFrame(x.columns)
feature_scores = pd.concat([df_columns, df_scores],axis=1)
feature_scores.columns = ['Feature_Name','Score']  # name output columns
print(feature_scores.nlargest(12,'Score'))         # print 10 best features
```

```
            Feature_Name      Score
10               alcohol  82.747058
1        volatile acidity  47.937979
9                sulphates  18.049074
2             citric acid  17.705465
6   total sulfur dioxide  15.270771
7                density   8.544380
0          fixed acidity   4.314182
8                     pH   4.149650
4              chlorides   3.690383
11                    Id   3.288904
5    free sulfur dioxide   2.692791
3         residual sugar   1.234693
```

Based on the scores from seleckbest we are dropping the columns from dataframe that doesn't have much importance to predict the target variable. Score less than 10 are being dropped from the dataframe.

```
x = x.drop(columns=['fixed acidity','pH','free sulfur dioxide','chlorides', 'density','residual sugar','Id'])
```

We are splitting the test and train data with a test size of 30% data from the dataframe.

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3,random_state=42)
```

```
# Details of training dataset
print("Shape of x_train dataset: ", x_train.shape)
print("Shape of y_train dataset: ", y_train.shape)
print("Shape of x_test dataset: ", x_test.shape)
print("Shape of y_test dataset: ", y_test.shape)
```

```
Shape of x_train dataset:  (800, 5)
Shape of y_train dataset:  (800,)
Shape of x_test dataset:  (343, 5)
Shape of y_test dataset:  (343,)
```

We have created a function that would help us in providing the score of the model that we want to use.

```
def evaluate_model(pipe, X, y):
    y_pred, y_true = np.empty(len(y)), np.empty(len(y))
    loo = LeaveOneOut()
    for i, (train_idx, test_idx) in tqdm(enumerate(loo.split(X)), total=len(y)):
        X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
        y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
        y_pred[i] = pipe.fit(X_train, y_train).predict(X_test)[0]
        y_true[i] = y_test
    return r2_score(y_true, y_pred), np.sqrt(mean_squared_error(y_true, y_pred))
```

We ran this for multiple models to choose the best from the list.

```
pipe = get_pipeline(PCA(n_components=5), model=LinearRegression())
model_score, score_std = get_model_score(pipe=pipe, data_x=x, data_y=y, cv=5)
r2, rmse = evaluate_model(pipe, x, y)
print('Linear Regression Model Score: ', model_score)
print('Linear Regression SD Score: ',score_std)
print('R Square: ', r2)
print('Root Mean Sqaure: ', rmse)
```

```
100%|██████████| 1143/1143 [01:12<00:00, 15.71it/s]

Linear Regression Model Score:  0.2915459444263656
Linear Regression SD Score:  0.029268572360845152
R Square:  0.3350070218364951
Root Mean Sqaure:  0.656838798694942
```

```
pipe = get_pipeline(PCA(n_components=5), model=RandomForestRegressor())
model_score, score_std = get_model_score(pipe=pipe, data_x=x, data_y=y, cv=5)
r2, rmse = evaluate_model(pipe, x, y)
print('Random Forest Regressor Model Score: ',model_score)
print('Random Forest Regressor SD Score: ',score_std)
print('R Square: ', r2)
print('Root Mean Sqaure: ', rmse)
```

```
100%|██████████| 1143/1143 [13:44<00:00,  1.39it/s]
```

```
Random Forest Regressor Model Score:  0.24549288362476976
Random Forest Regressor SD Score:  0.06126130522848122
R Square:  0.4197712193993324
Root Mean Sqaure:  0.6135499172311915
```

```
def model_evaluation(model, x_train, y_train, x_test, y_test):
    mod = model.fit(x_train, y_train)
    mod_pred = model.predict(x_test)
    return accuracy_score(y_test, mod_pred), np.sqrt(mean_squared_error(y_test, mod_pred))
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier()
model_score, score_std = model_evaluation(model, x_train, y_train, x_test, y_test)
#r2, rmse = evaluate_model(pipe, x, y)
print('Random Forest Classifier Model Score: ',model_score)
print('Root Mean Sqaure: ', rmse)
```

```
Random Forest Classifier Model Score:  0.6326530612244898
Root Mean Sqaure:  0.6135499172311915
```

```
model=DecisionTreeClassifier()
model_score, score_std = model_evaluation(model, x_train, y_train, x_test, y_test)
#r2, rmse = evaluate_model(pipe, x, y)
print('Decision Tree Classifier Model Score: ',model_score)
print('Root Mean Sqaure: ', rmse)
```

```
Decision Tree Classifier Model Score:  0.5451895043731778
Root Mean Sqaure:  0.6135499172311915
```

Model Name	Score
Linear Regression	29.15%
Random Forest Regresssor	24.54%
Random Forest Classifier	63.26%

Decision Tree Classifier	54.51%

Out of all the models, Random Forest classifier has the best score and we can use that to predict the quality of the wine. The quality and taste plays an important role in pricing of the wine. We have so many brands of wine where the price ranges from a min of $5 to some thousands of dollars. These models would help the companies when buying the wine from a manufacturer regarding the quality of the wine.

Appendix:

- **Random forest:** It builds multiple decision trees and merges them to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.
- **Logistic regression:** It is the appropriate regression analysis to conduct when the dependent variable is binary. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables.
- **Linear Regression:** Linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables. The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression.
- **Random Forest Regression:** Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. A Random Forest operates by

constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

- **Leave One Out:** The Leave-One-Out Cross-Validation, or LOOCV, procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.
- **Box Cox Transformation:** A Box Cox transformation is a transformation of non-normal dependent variables into a normal shape. Normality is an important assumption for many statistical techniques; if your data isn't normal, applying a Box-Cox means that you are able to run a broader number of tests.
- **Decision Tree:** It belongs to the family of supervised learning algorithms. It can be used for solving regression and classification problems. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from training data. In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. Based on the comparison, we follow the branch corresponding to that value and jump to the next node.
- Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

- A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.