Wine Quality Prediction using Machine Learning

1 INTRODUCTION

Wine is the most commonly used beverage globally, and its values are considered important in society. Wine is an alcoholic drink that is made up of fermented grapes. Quality of wine is important for its consumers, mainly for producers in the present competitive market to raise the revenue. Wine quality refers to the factors that go into producing a wine, as well as the indicators or characteristics that tell you if the wine is of high quality. Historically, wine quality used to be determined by testing at the end of the production.

1.1 Overview

In this project, we present a wine quality prediction technique that utilizes historical data to train simple machine learning models which are more accurate and can help us know the quality of wine. The models can be run on much less resource intensive environments. From this the best model is selected and saved in pkl format. We will be doing flask integration.

1.2 Purpose

1. To experiment with different classification methods to see which yields the highest accuracy.
2. To determine which features are the most indicative of a good wine quality.

2 LITERATURE SURVEY

If you have come across wine then you will notice that wine has also their type, they are red and white wine. According to experts, wine is differentiated according to its smell, flavour, and colour, but we are not wine experts to say that wine is good or bad. Every person has their own opinion about the tastes, so identifying a quality based on a person’s taste is challenging. Judging the quality of wine manually is a really tough task, even the professional wine tasters have the accuracy of 71%.

2.1 Existing problem

For this project, I used winequalityN dataset to build various classification models to predict whether the particular red wine is "good quality" or not. Each wine dataset is given a "quality" score between 0 and 10. For the purpose of this project, I converted the output to a binary output where each wine is either "good quality" (a score of 7 or higher) or not (a score below 7). The quality of wine by 12 input variables:

1. type
2. fixed acidity
3. volatile acidity
4. citric acid
5. residual sugar
6. chlorides
7. free sulphur dioxide
8. total sulphur dioxide
9. density
10. pH
11. sulfates
12. alcohol

2.2 Proposed solution

To accomplish this, we have to complete all the activities and tasks listed below

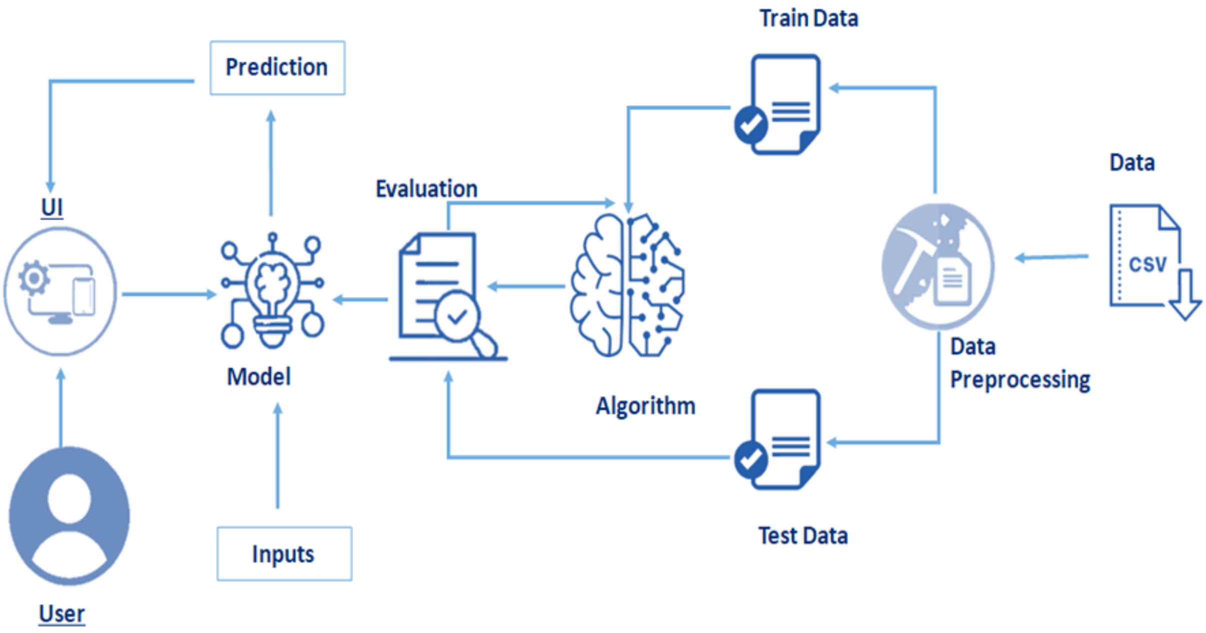
1. Data Collection
    a. Collect the dataset or Create the dataset
2. Visualizing and analyzing data
    a. Univariate analysis
    b. Bivariate analysis

   c. Multivariate analysis
   d. Descriptive analysis
3. Data Pre-processing
   a. Drop unwanted features
   b. Checking for null values
   c. Handling categorical data
   d. Splitting data into train and test
   e. Feature Scaling
4. Model Building
   a. Import the model building Libraries
   b. Initializing the model
   c. Training and testing the model
   d. Evaluation of Model
   e. Save the Model
5. Application Building
   a. Create an HTML file
   b. Build a Python Code

3 THEORITICAL ANALYSIS

First we need to collect the dataset and then perform Data Visualization, Data Preprocessing, Model Building and then we need to build an application for prediction of wine quality.

3.1 Block diagram



3.2 Hardware / Software designing

Hardware Requirement :

1. A PC
2. Keyboard
3. CPU
4. Mouse

Software Requirement:

1. Anaconda Navigator
2. Python Packages

1. Open anaconda prompt as administrator:

1. Type **“pip install numpy”** and click enter.
2. Type **“pip install pandas”** and click enter.
3. Type **“pip install scikit-learn”** and click enter.
4. Type **”pip install matplotlib”** and click enter.
5. Type ”**pip install pickle-mixin”** and click enter.
6. Type **”pip install seaborn”** and click enter.
7. Type **“pip install Flask”** and click enter.

4 EXPERIMENTAL ANALYSIS

1. User interacts with the UI to enter the input.
2. Entered input is analyzed by the model which is integrated.
3. Once model analyses the input the prediction is showcased on the UI

4.1 Dataset Collection

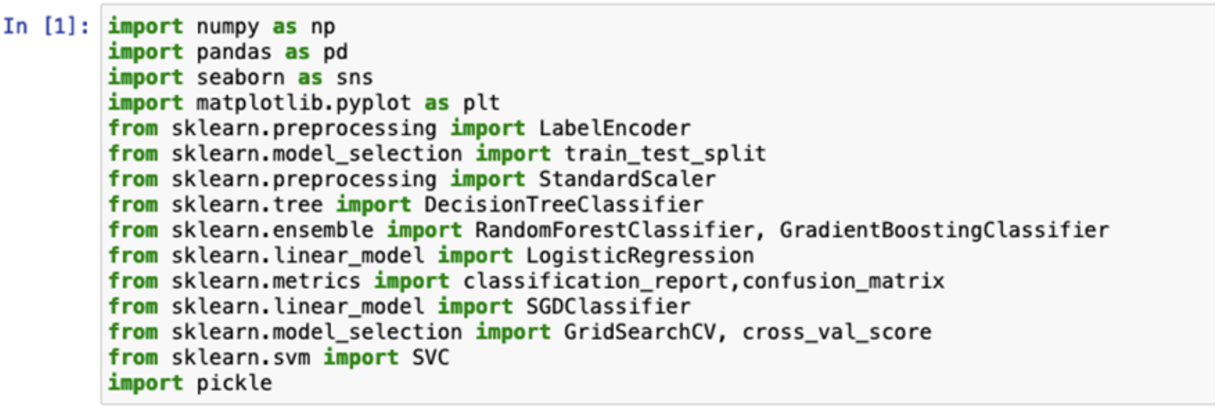
ML depends heavily on data, it is the most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.I collected my "winequalityN.csv" dataset from smart internz website.

4.2 Visualizing and Analysing the data

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analyzing techniques.

1. Importing The Libraries :

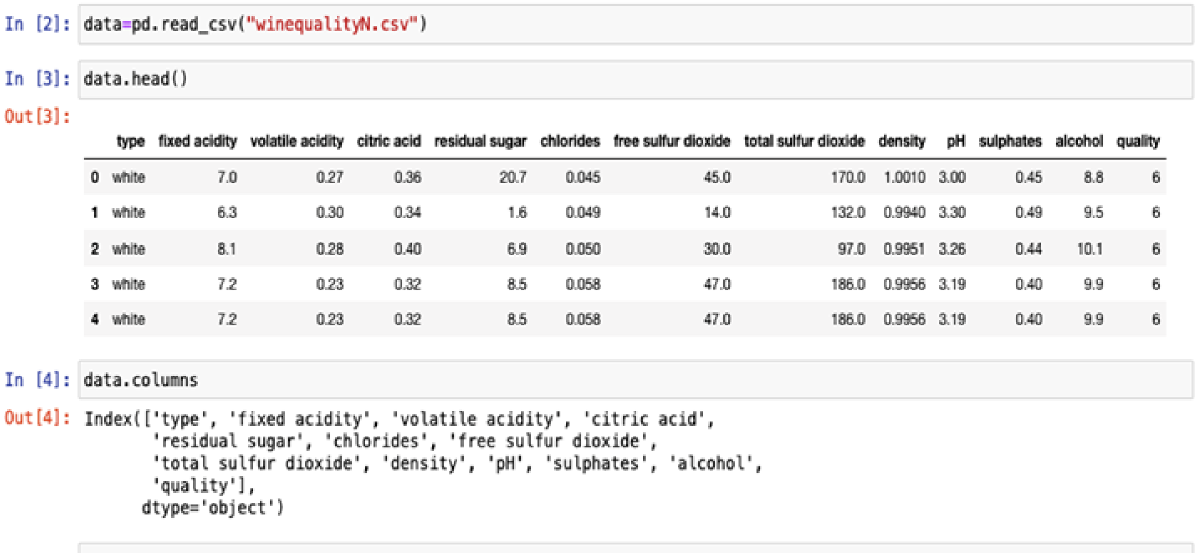
Import the necessary libraries as shown in the image.



```
In [1]: import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        from sklearn.preprocessing import LabelEncoder
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import classification_report,confusion_matrix
        from sklearn.linear_model import SGDClassifier
        from sklearn.model_selection import GridSearchCV, cross_val_score
        from sklearn.svm import SVC
        import pickle
```

1. Read The Dataset :

1. Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.In pandas we have a function called **read_csv()** to read the dataset. As a parameter we have to give the directory of csv file.



```
In [2]: data=pd.read_csv("winequalityN.csv")

In [3]: data.head()

Out[3]:
```

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	white	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	white	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	white	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	white	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	white	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

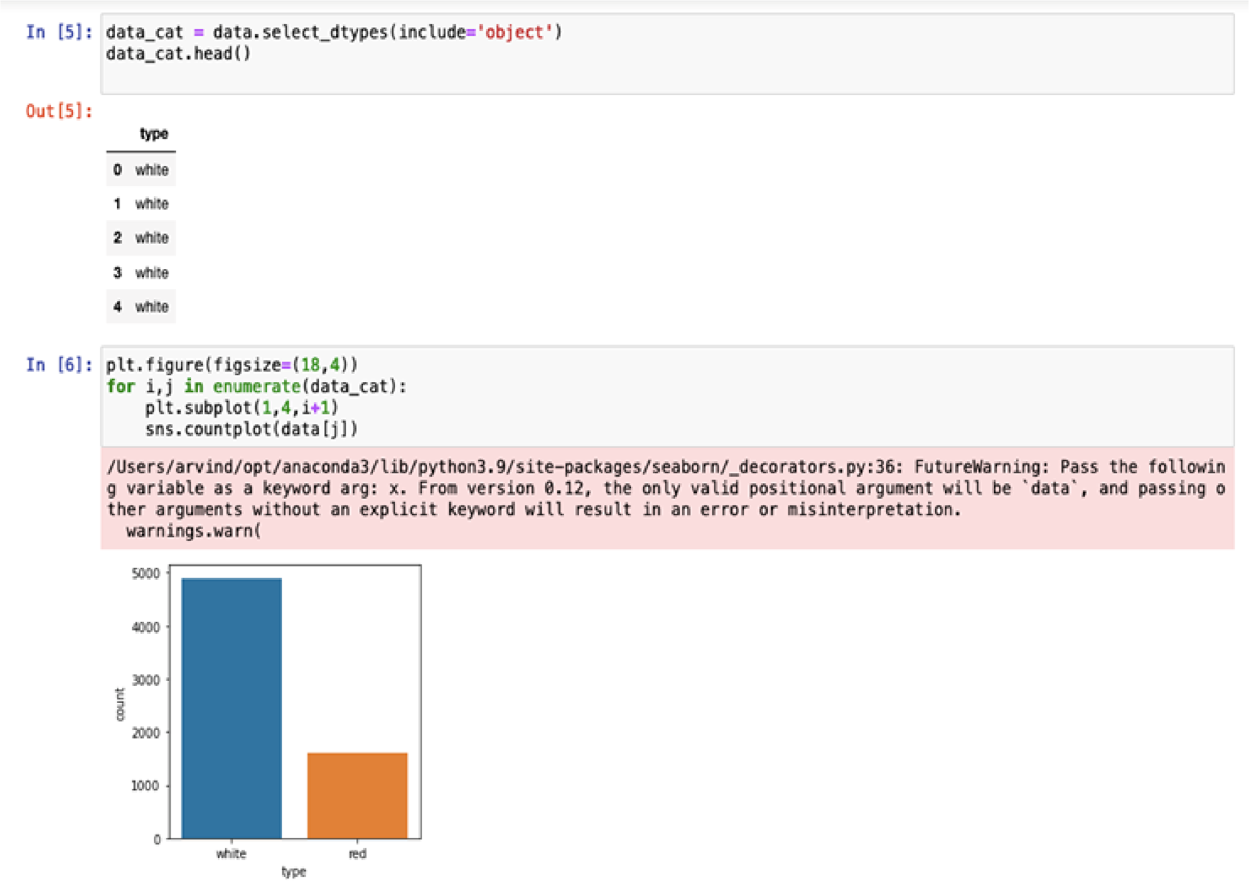
```
In [4]: data.columns

Out[4]: Index(['type', 'fixed acidity', 'volatile acidity', 'citric acid',
               'residual sugar', 'chlorides', 'free sulfur dioxide',
               'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol',
               'quality'],
              dtype='object')
```

1. Uni-variate Analysis :

1. In simple words, univariate analysis is understanding the data with single feature.

Here we have displayed two different graphs such as pie plot, box plot and count plot.

1. Seaborn package provides a wonderful function distplot. The distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution. With the help of distplot, we can find the distribution of the feature. We have used distplot here to check whether alcohol is normally distributed or is skewed.
1. In our dataset we have a categorical features. With the countplot function, we are going to count the unique category in that feature. With for loop and subplot we have plotted this below graph.
2. From the plot we came to know, count of white wine observations is much more than the red wine.
1. The hist() function in pyplot module of matplotlib library is used to plot a histogram.
2. As we can see, the most common vote is '6', when the lowest vote is '3', and the highest vote is '6'. In general, we may see that most of the parameters (except the "type" parameter, which is binary parameter) are normally distributed.



```
In [5]: data_cat = data.select_dtypes(include='object')
        data_cat.head()
```

Out[5]:

	type
0	white
1	white
2	white
3	white
4	white

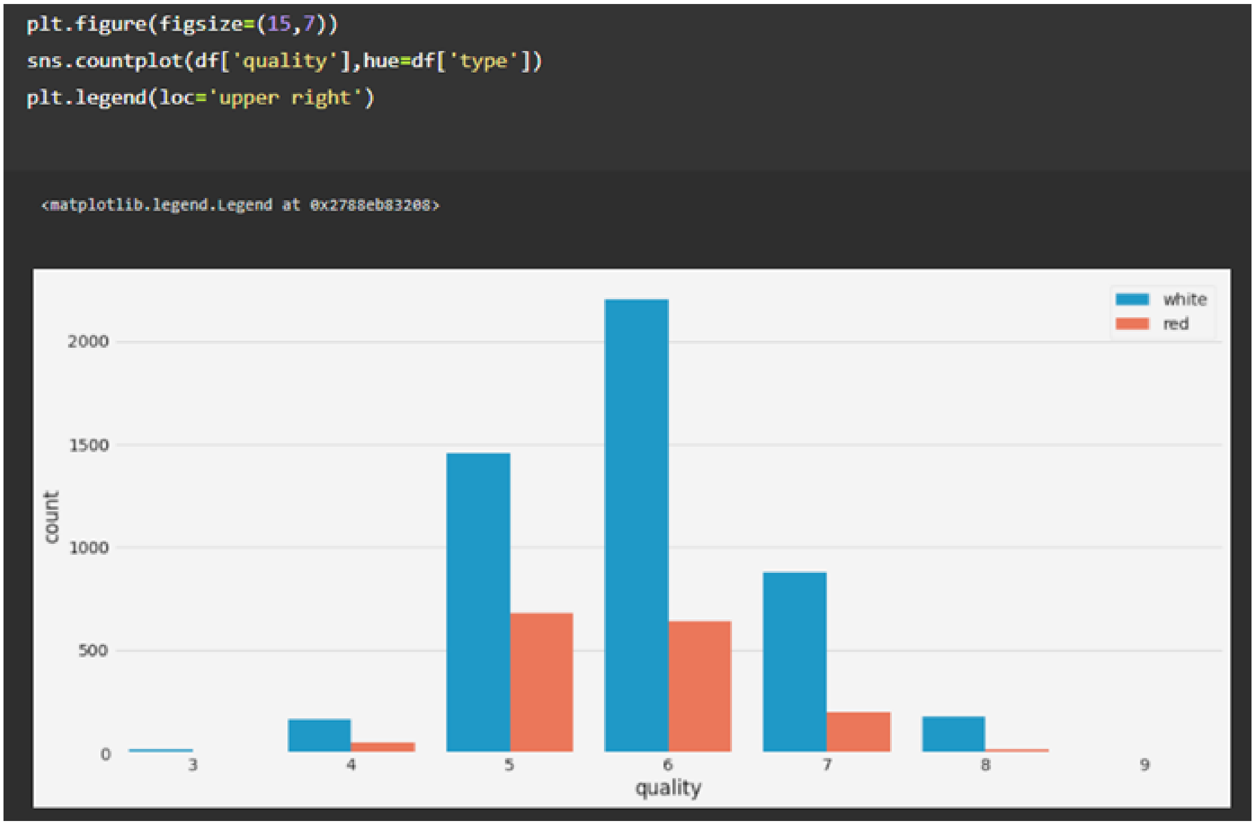
```
In [6]: plt.figure(figsize=(18,4))
        for i,j in enumerate(data_cat):
            plt.subplot(1,4,i+1)
            sns.countplot(data[j])
```

```
/Users/arvind/opt/anaconda3/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the followin
g variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing o
ther arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

1. Bi-variate Analysis

1. To find the relation between two features we use bivariate analysis.

1. Count plot is used here. As a 1^{st} parameter we are passing x value and as a 2^{nd} parameter we are passing hue value.
2. From this plot we can see the relationship between type and the quality of the data.



```
plt.figure(figsize=(15,7))
sns.countplot(df['quality'],hue=df['type'])
plt.legend(loc='upper right')
```

```
<matplotlib.legend.Legend at 0x2788eb83208>
```

1. A scatter plot is a means to represent data in a graphical format. A simple scatter plot makes use of the Coordinate axes to plot the points, based on their values. Scatter plots uses dots to represent individual pieces of data.

1. A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.

1. Let’s see the relationship between the dependent and independent variables of our dataset.
2. The visualization represents the relationship between Citric Acid and our target variable, Quality. We can see there’s not much variation in citric acid values over the quality.

1. Multivarlate Analysis

1. In simple words, multivariate analysis is to find the relation between multiple features. Here we have used a heatmap from the seaborn package.

1. Correlation is a statistical measure that expresses the extent to which two variables are linearly related. It's a common tool for describing simple relationships without making a statement about cause and effect.
2. To visualize the correlation heat map() function is used. From the below image we can easily find the highly correlated feature. Abs() method is used to convert the negative correlation to positive correlation.

1. From the above correlation plot for the given dataset for wine quality prediction, we can easily see which items are related strongly with each other and which items are related weekly with each other. For Example,
2. The strongly correlated items are :

   1.fixed acidity and citric acid. 2.free sulphur dioxide and total sulphor dioxide. 3.fixed acidity and density. 4. alcohol and quality.

   so, from above points there is a clear inference that alcohol is the most important characteristic to determine the quality of wine.

1. The weakly correlated items are :

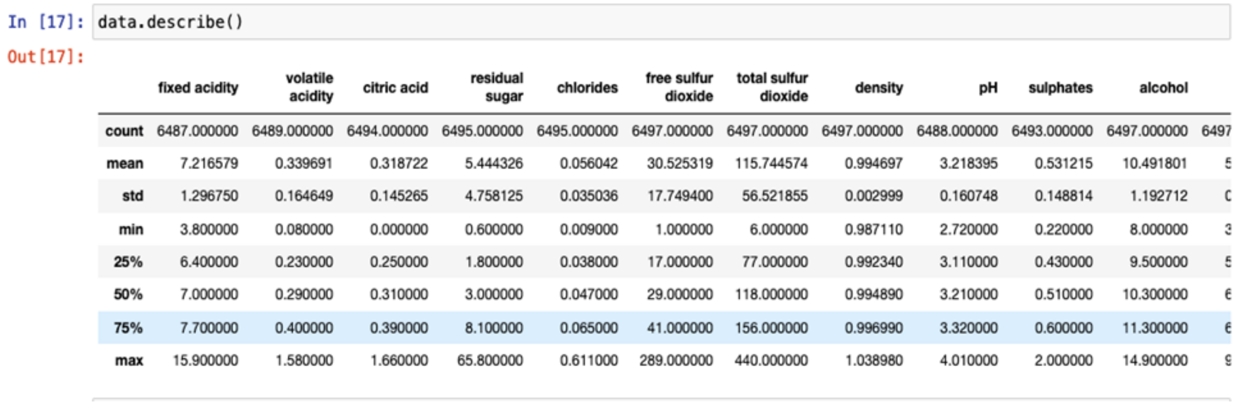
   1.citric acid and volatile acidity. 2.fixed acidity and ph. 3.density and alcohol.

   These are some relations which do not depend on each other at all.

1. Descriptive Analysis

1. Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of

continuous features.



```
In [17]: data.describe()
Out[17]:
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	
count	6487.000000	6489.000000	6494.000000	6495.000000	6495.000000	6497.000000	6497.000000	6497.000000	6488.000000	6493.000000	6497.000000	6497
mean	7.216579	0.339691	0.318722	5.444326	0.056042	30.525319	115.744574	0.994697	3.218395	0.531215	10.491801	5
std	1.296750	0.164649	0.145265	4.758125	0.035036	17.749400	56.521855	0.002999	0.160748	0.148814	1.192712	C
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000	8.000000	3
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000000	0.992340	3.110000	0.430000	9.500000	5
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000000	0.994890	3.210000	0.510000	10.300000	6
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000000	0.996990	3.320000	0.600000	11.300000	6
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000000	1.038980	4.010000	2.000000	14.900000	9

4.3 Data Preprocessing

As we have understood how the data is. Lets pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much of randomness so we need to clean the dataset properly in order to fetch good results. This activity includes Removing unwanted columns, Handling missing values, Converting the target variable into binary class variable, Handling categorical data, Splitting dataset into training and test set and feature scaling.

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

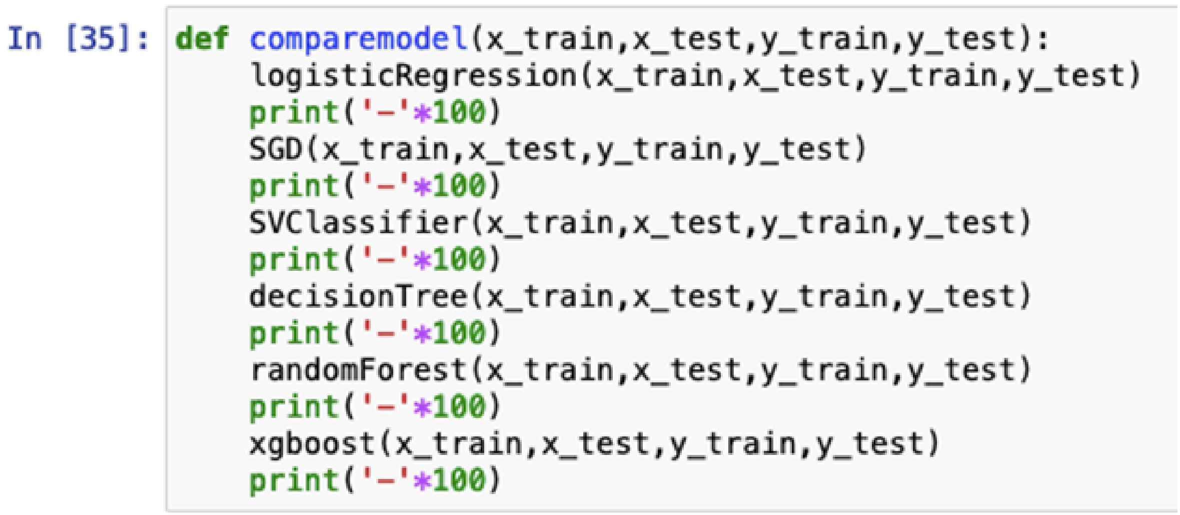
4.4 Model Building

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying two regression algorithms. The best model is saved based on its performance.The models used are:

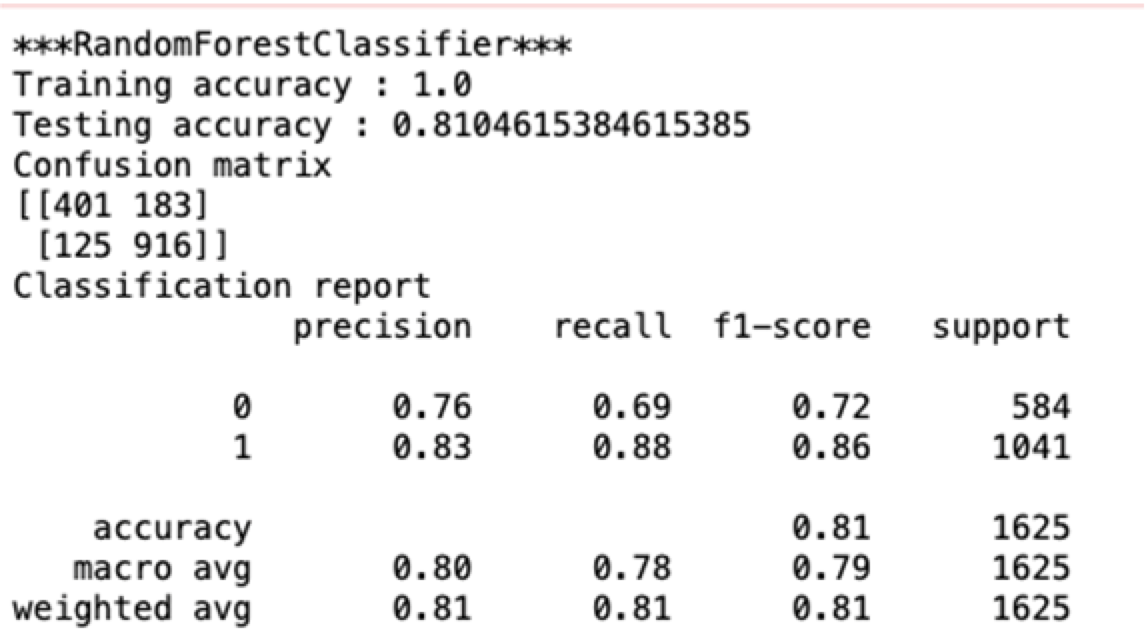
1. Linear Regression

2. Stochastic Gradient Descent Classifier Model
3. Support Vector Classifier Model
4. Decision Tree Model
5. Random Forest Regressor Model
6. XGBoost Model

For comparing the above four models compareModel function is defined.
After calling the function, the results of models are displayed as output. From the six models, random forest is performing well. From the below image, we can see the train accuracy of the model is 81% accuracy. So, here random forest is selected as the best performing algorithm and evaluated with cross validation. Additionally, we can tune the model with hyper parameter tuning techniques.

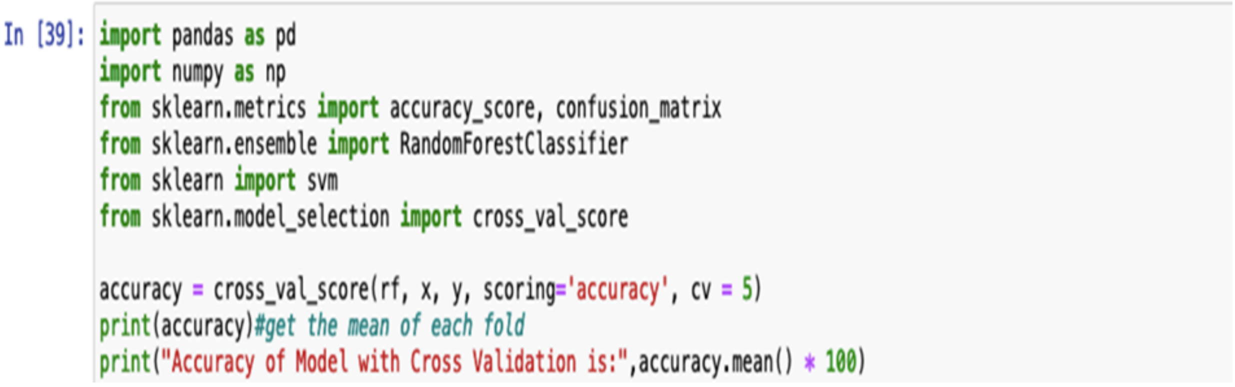


```
In [35]: def comparemodel(x_train,x_test,y_train,y_test):
             logisticRegression(x_train,x_test,y_train,y_test)
             print('-'*100)
             SGD(x_train,x_test,y_train,y_test)
             print('-'*100)
             SVClassifier(x_train,x_test,y_train,y_test)
             print('-'*100)
             decisionTree(x_train,x_test,y_train,y_test)
             print('-'*100)
             randomForest(x_train,x_test,y_train,y_test)
             print('-'*100)
             xgboost(x_train,x_test,y_train,y_test)
             print('-'*100)
```

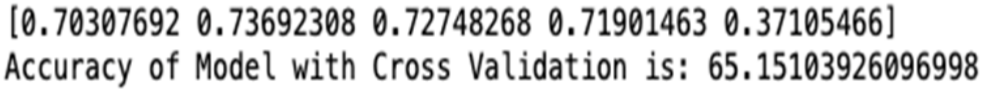


```
***RandomForestClassifier***
Training accuracy : 1.0
Testing accuracy : 0.8104615384615385
Confusion matrix
[[401 183]
 [125 916]]
Classification report
              precision    recall  f1-score   support

           0       0.76      0.69      0.72       584
           1       0.83      0.88      0.86      1041

    accuracy                           0.81      1625
   macro avg       0.80      0.78      0.79      1625
weighted avg       0.81      0.81      0.81      1625
```

Evaluating performance of the model and saving the model :
From sklearn, cross_val_score is used to evaluate the score of the model. On the parameters, we have given rf (model name), x, y, cv (as 5 folds). Our model is performing well. So, we are saving the model by pickle.dump().



```
In [39]: import pandas as pd
         import numpy as np
         from sklearn.metrics import accuracy_score, confusion_matrix
         from sklearn.ensemble import RandomForestClassifier
         from sklearn import svm
         from sklearn.model_selection import cross_val_score

         accuracy = cross_val_score(rf, x, y, scoring='accuracy', cv = 5)
         print(accuracy)#get the mean of each fold
         print("Accuracy of Model with Cross Validation is:",accuracy.mean() * 100)
```



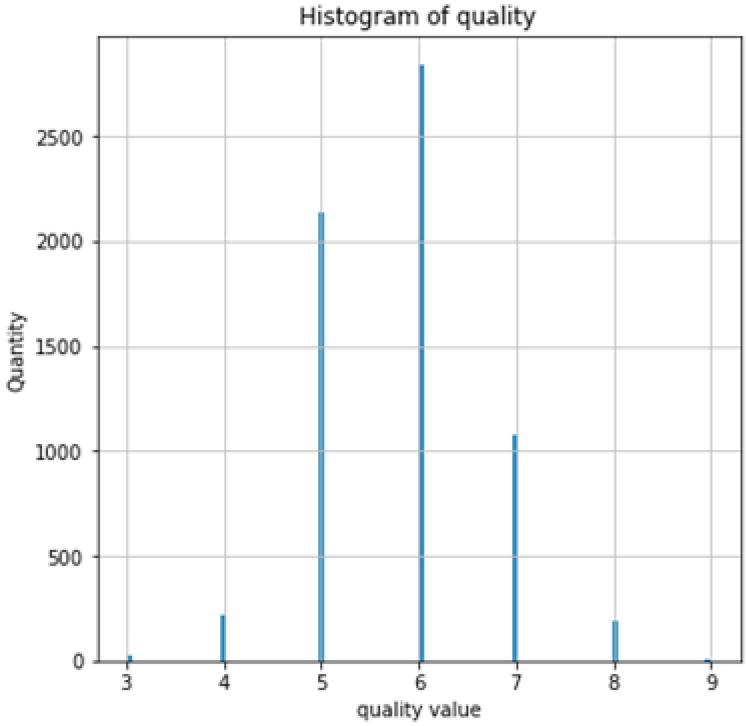
```
[0.70307692 0.73692308 0.72748268 0.71901463 0.37105466]
Accuracy of Model with Cross Validation is: 65.15103926096998
```

4.5 Application Building

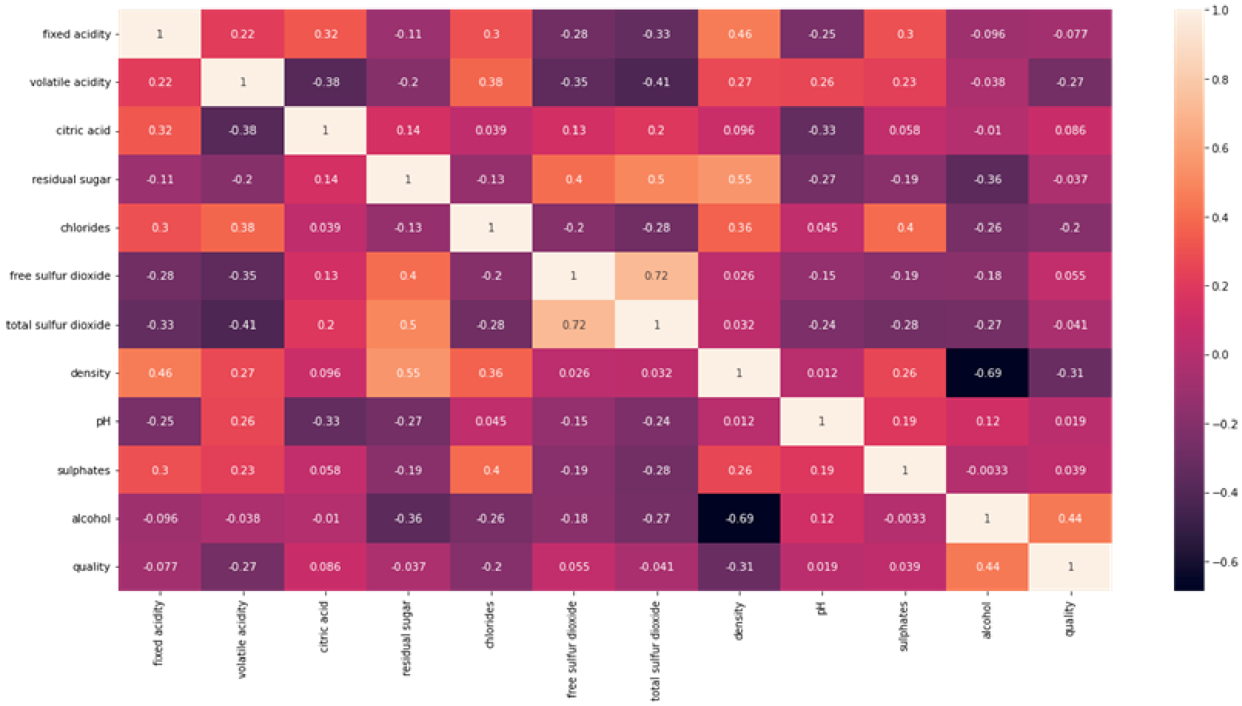
This section will build a web application integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.
This section has the following tasks

1. Building HTML Pages
2. Building server-side script

5 FLOWCHART

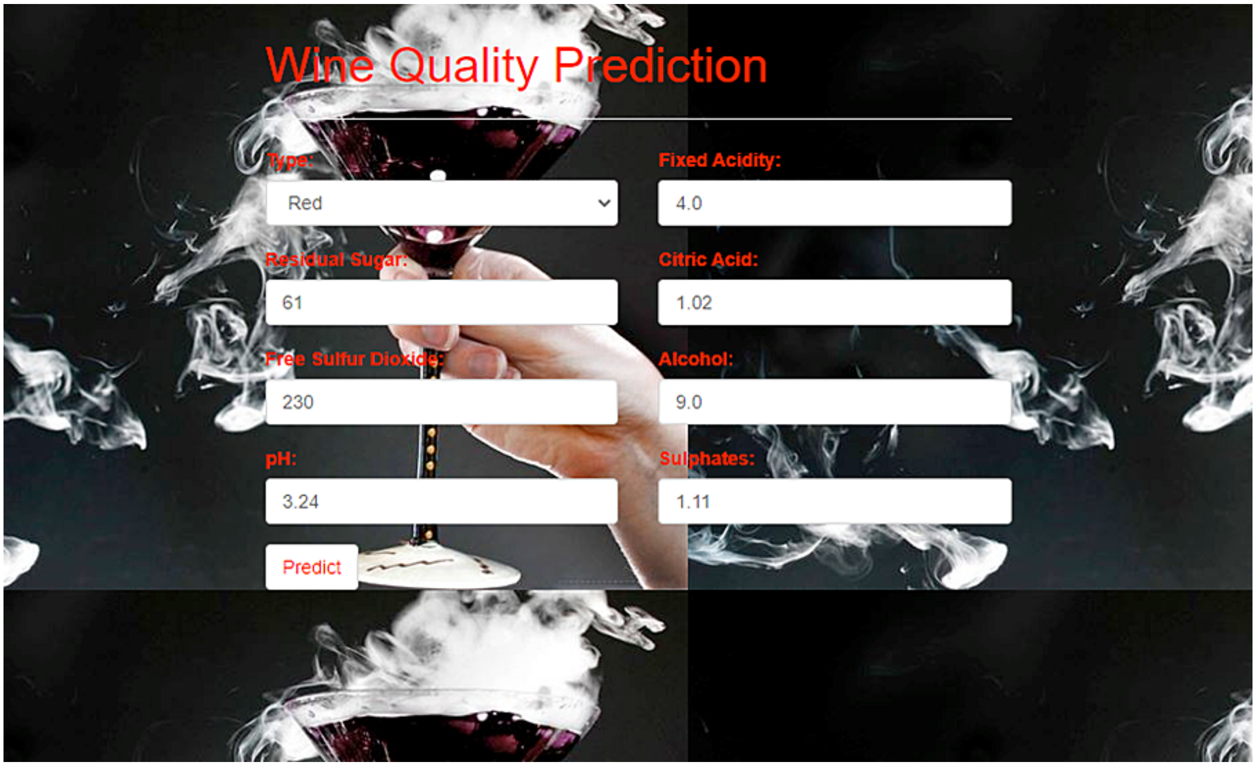


HeatMap



6 RESULT

Input:



Wine Quality Prediction

Type: Red

Fixed Acidity: 4.0

Residual Sugar: 61

Citric Acid: 1.02

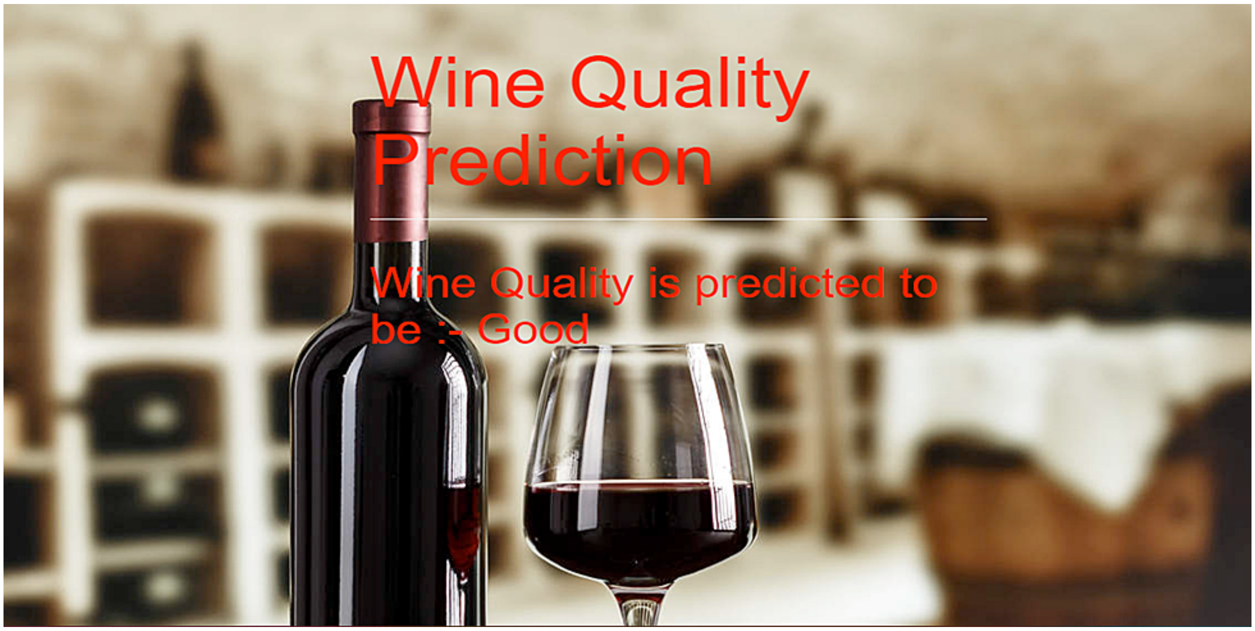
Free Sulfur Dioxide: 230

Alcohol: 9.0

pH: 3.24

Sulphates: 1.11

Final Output:



Wine Quality Prediction

Wine Quality is predicted to be :- Good

7 ADVANTAGES & DISADVANTAGES

Advantages of this project are :

1. You can predict the quality of wine.

1. You’ll be able to understand the problem to classify if it is a regression or a classification kind of problem.

2. You will be able to know how to pre-process/clean the data using different data pre-processing techniques.

3. You will be able to analyze or get insights into data through visualization.

4. Applying different algorithms according to the dataset and based on visualization.

5. You will be able to know how to find the accuracy of the model.

6. You will be able to know how to build a web application using the Flask framework.

Disadvantages of this project are :

1. Time Consuming
2. More Resources and Space
3. Algorithm Selection

8 APPLICATIONS

1. First, you can predict the quality of wine.
2. By predicting the wine quality you can use correct amount of chemical ingredients.
3. By using the correct amount of chemical ingredients you can make good wine.
4. The main application of building this machine learning model is to make high quality of wine by predicting the data.

9 CONCLUSION

The current study provides the prediction of wine quality using machine learning.By the end of this project you will be able to understand the problem to classify it is a regression or classification kind of problem.You will be able to preprocess the the data using different data-preprocessing techniques.You will be able to analyse the data through visualization.Applying different algorithms according to the dataset and based on visualization.You will be able to know how to find the accuracy of the model. You will be able to know how to build a web application using Flask framework.

10 FUTURE SCOPE

1. In future, I want to learn and use Machine Learning through various platforms like Google Co-lab, IBM Machine Learning, Azure Machine Learning, Tensor Flow etc...
2. I also want to learn and use Cloud Computing through various platform like

Amazon Web Service, Google Cloud Platform, Microsoft Azure etc...

3. I also want to learn and use Django python web framework that enables rapid development of secure and maintainable websites.

11 BIBLIOGRAPY

1. guided learning module-applied data science https://smartinternz.com/Student/guided_project_info/8872#

1. D. H. Wolpert. The supervised learning no-free-lunch theorems. In Soft Computing and Industry, pages 25–42. Springer, 2012.

2. A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings of the twenty-first international conference on Machine learning, page 78. ACM, 2017.

3. R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai, volume 14, pages 1137–1145, 2019.