WINE QUALITY PREDICTION

Submitted To - Mr. Mayur Dev Sewak
General Manager,
Operations Eisystems Services

&

Ms. Mallika Srivastava
Trainer, Data Science & Analytics
Domain Eisystems Services

Submitted By – ABHILASH BURAGOHAIN

Course – Data Science / Machine Learning

CONTENTS

ABSTRACT

The quality of a wine is important for the consumers as well as the wine industry. The traditional (expert) way of measuring wine quality is time-consuming. Nowadays, machine learning models are important tools to replace human tasks. In this case, there are several features to predict the wine quality but the entire features will not be relevant for better prediction. So, our thesis work is focusing on what wine features are important to get the promising result. For the purpose of classification model and evaluation of the relevant features, we used three algorithms namely support vector machine (SVM), naïve Bayes (NB), and artificial neural network (ANN). In this study, we used two wine quality datasets red wine and white wine. To evaluate the feature importance we used the Pearson coefficient correlation and performance measurement matrices such as accuracy, recall, precision, and f1 score for comparison of the machine learning algorithm. A grid search algorithm was applied to improve the model accuracy. Finally, we achieved the artificial neural network (ANN) algorithm has better prediction results than the Support Vector Machine (SVM) algorithm and the Naïve Bayes (NB) algorithm for both red wine and white wine datasets.

KEYWORDS

Quality of wine, Gradient descent classifier (GDC), Logistic Regression, Hadoop, Random Forest Classifier (RFC), Decision Tree Classifier (DT).

INTRODUCTION

The quality of the wine is a very important part for the consumers as well as the manufacturing industries. Industries are increasing their sales using product quality certification. Nowadays, all over the world wine is a regularly used beverage and the industries are using the certification of product quality to increases their value in the market. Previously, testing of product quality will be done at the end of the production, this is time taking process and it requires a lot of resources such as the need for various human experts for the assessment of product quality which makes this process very expensive. Every human has their own opinion about the test, so identifying the quality of the wine based on humans experts it is a challenging task.

There are several features to predict the wine quality but the entire features will not be relevant for better prediction.

Our goal for this project is to conduct an analysis on Wine Quality data from the University of California, Irvine Machine Learning Repository. The data are the results of a chemical analysis of wines grown in the same region in Italy, but derived from three different cultivars. The data has 13 attributes which are wine type, fixed acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality. We are interested to see if these attributes has an impact on the quality of wine. Therefore, our goal is to use machine learning to see if we can build an accurate model to predict the wine quality. Below we can see the features that were used to build this model.

PROJECT SUMMARY AND OBJECTIVES

My analysis will use Red Wine Quality Data Set, available on the UCI machine learning repository. I obtained the red wine samples from the north of Portugal model red wine quality based physiochemical tests. The dataset contains total 12 variables, which were recorded for 1599 observation. This data will allow us to create different regression models to determine how different independent variables help predict our dependent variable, quality. Knowing how each variable will impact the red wine quality will help producers, distributors and businesses in the red wine industry better assess their production, distribution, and pricing strategy.

Wine classification is a difficult task since taste is the least understood of human senses. A good wine quality prediction can be very useful in the certification phase, since currently the sensory analysis is performed by human tasters, being clearly a subjective approach. An automatic predictive system can be integrated into a decision support system, helping the speed and quality of the performance. Furthermore, a feature selection process can help to analyze the impact of the analytical tests. If it is concluded that several input variables are highly relevant to predict the wine quality, since in the production process some variables can be controlled, this information can be used to improve the wine quality. Classification models used here are:

1. Random Forest
2. Stochastic Gradient Descent
3. SVC
4. Logistic Regression

LANGUAGE USED

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For JIA, libraries used are speech recognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc.

Libraries Used:

1. **NumPy**: It is a general-purpose array processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is an open source software.
2. **Pandas**: It is a Python library used for working with data sets.It has functions for analyzing, cleaning, exploring, and manipulating data.The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. Pandas allows us to analyse big data and make conclusions based on statistical theories.Pandas can clean messy data sets, and make them readable and relevant.Relevant data is very important in data science.
3. **Matplotlib**: Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications. A Python matplotlib script is structured so that a few lines of code are all that is required in most instances to generate a visual data plot.
4. **Seaborn**: Seaborn is a Python data visualization library based on matplotlib.

It provides a high-level interface for drawing attractive and informative statistical graphics.

5. **Sklearn.model selection**: The classes in the sklearn.feature_selection module can be used for feature selection/dimensionality reduction on sample sets, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets.

6. **Sklearn.ensemble**: The sklearn.ensemble module includes two averaging algorithms based on randomized decision trees: the RandomForest algorithm and the Extra-Trees method. Both algorithms are perturb-and-combine techniques [B1998] specifically designed for trees. This means a diverse set of classifiers is created by introducing randomness in the classifier construction. The prediction of the ensemble is given as the averaged prediction of the individual classifiers.

7. **Sklearn.metrics**: The sklearn.metrics module implements functions assessing prediction error for specific purposes. These metrics are detailed in sections on Classification metrics, Multilabel ranking metrics, Regression metrics and Clustering metrics.

SOFTWARE USED

ANACONDA NAVIGATOR

Anaconda comes with over 250 packages automatically installed and over 7500 additional packages can be installed from PyPi as well as the conda package and environment manager. It also includes a GUI, Anaconda Navigator as a graphical alternative to the command-line interface (CLI).

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

JUPYTER NOTEBOOK:

Jupyter Notebook (formerly IPython Notebook) is a web-based interactive computational environment for creating notebook documents. Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension

DIAGRAMS

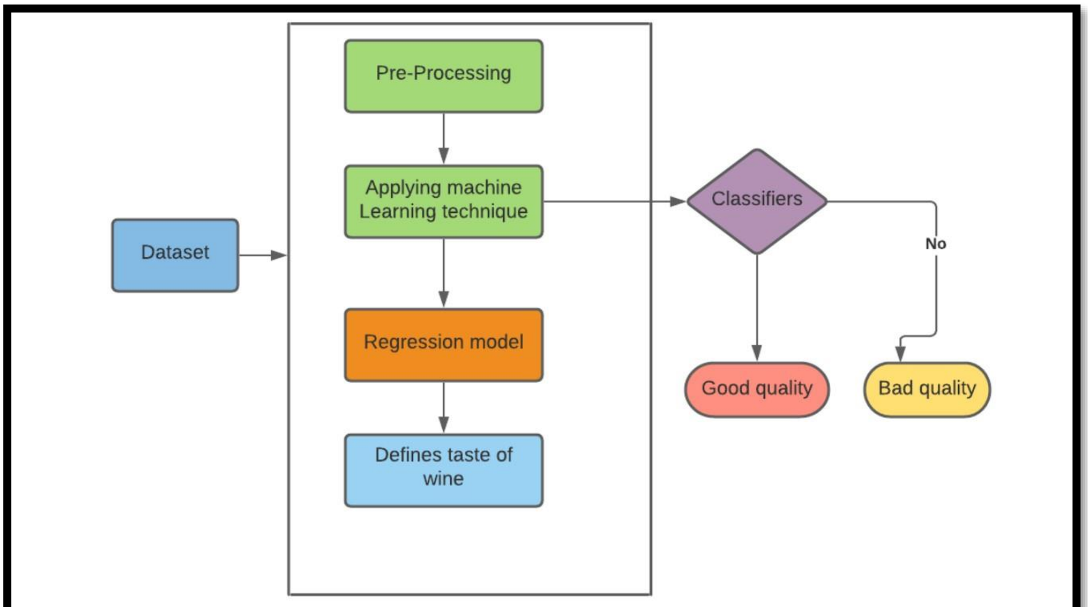
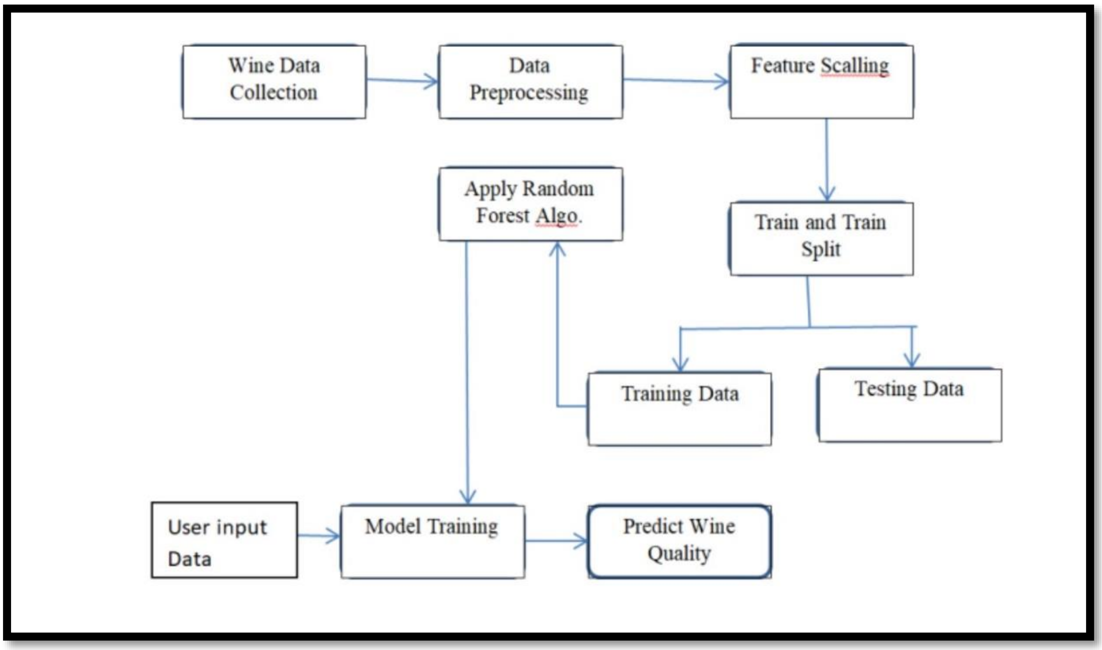


Figure 1: Proposed Wine Quality Classification model

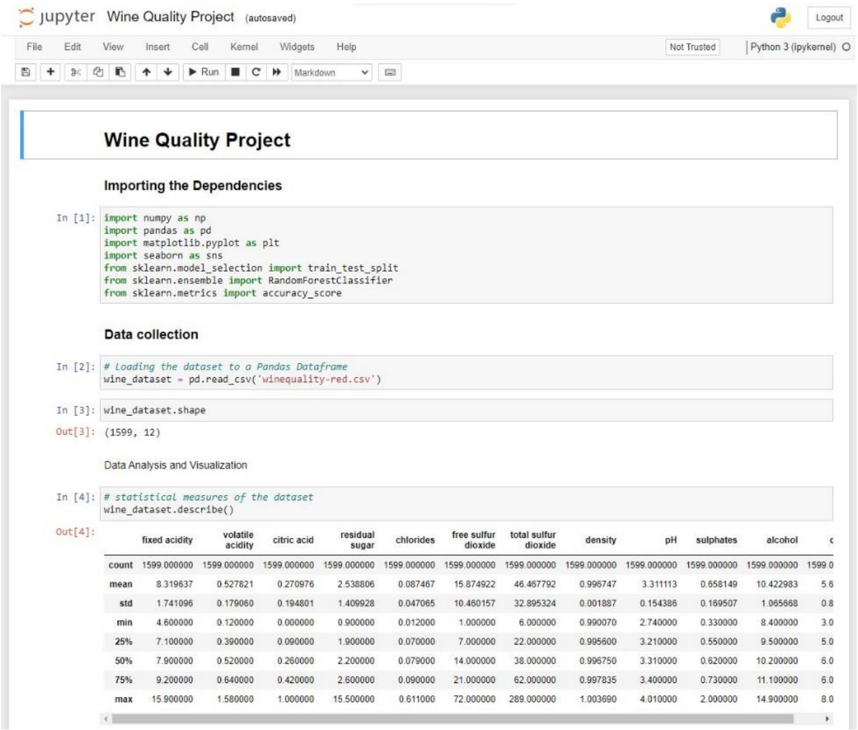
The overall flow structure of the proposed Wine quality classification scheme is depicted in figure 1. The wine quality dataset taken from Kaggle. Subsequently, the machine learning models where used which includes regression analysis and classification techniques to presume the better accuracy of the model. To classify given input data into good quality and bad quality, all data were fed into three different classifier.



DATA FLOW DIAGRAM

SCREENSHOTS OF THE CODE

IMG 1



Wine Quality Project

Importing the Dependencies

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.model_selection import train_test_split
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.metrics import accuracy_score
```

Data collection

```
In [2]: # Loading the dataset to a Pandas Dataframe
        wine_dataset = pd.read_csv('winequality-red.csv')
```

```
In [3]: wine_dataset.shape

Out[3]: (1599, 12)
```

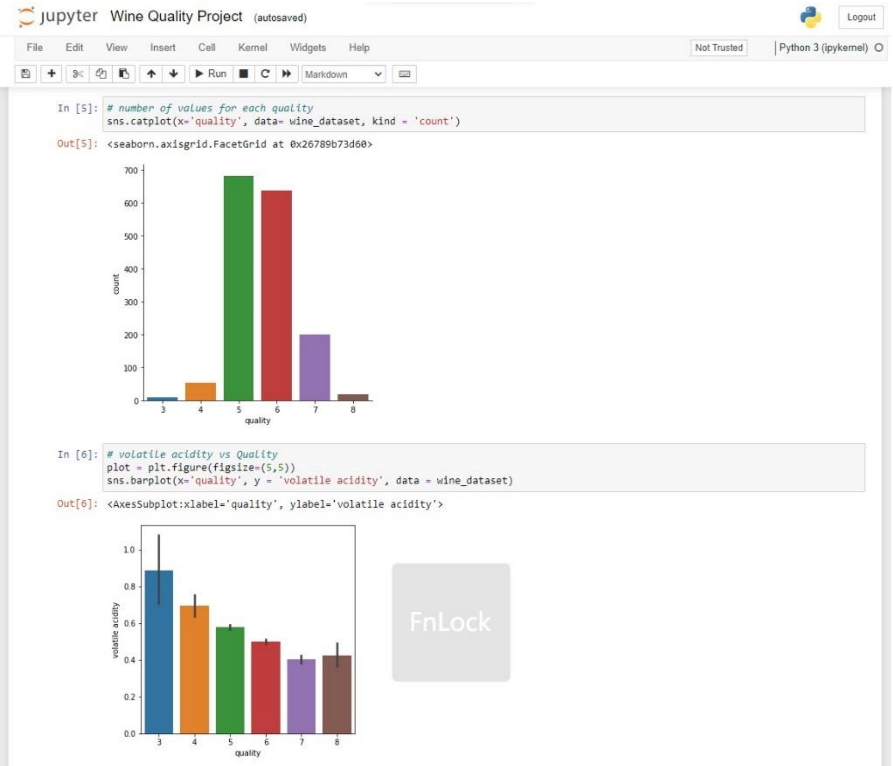
Data Analysis and Visualization

```
In [4]: # statistical measures of the dataset
        wine_dataset.describe()
```

Out[4]:

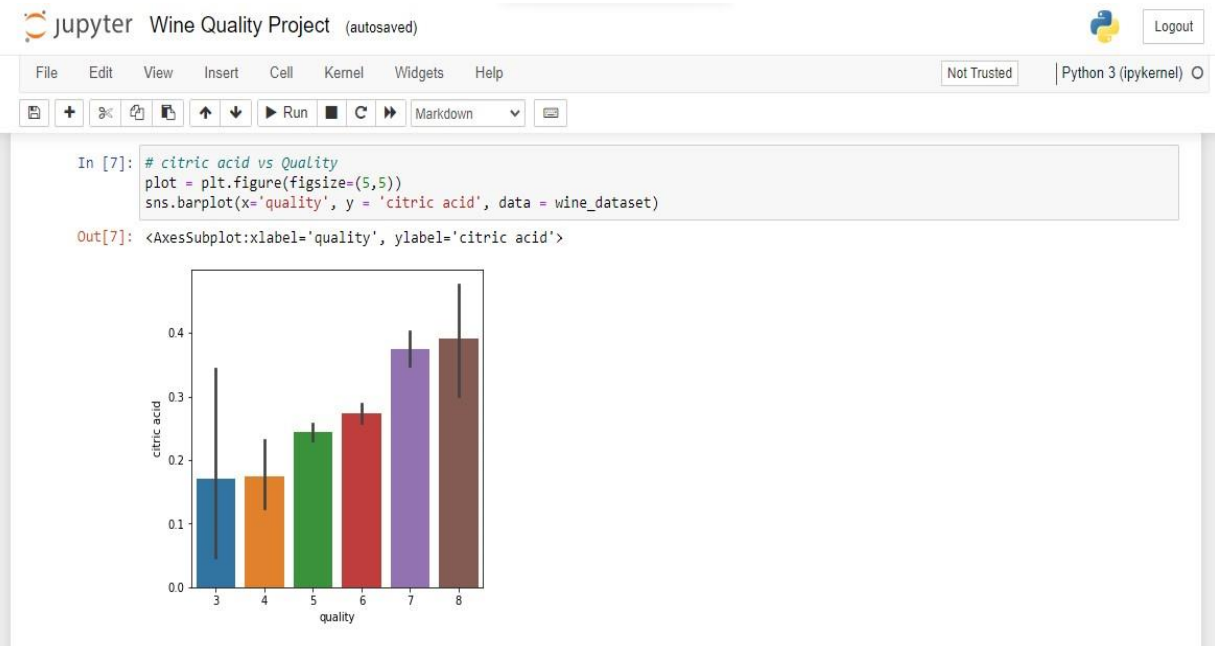
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	c
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.0
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.6
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.8
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.0
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.0
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.0
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.0
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.0

IMG 2



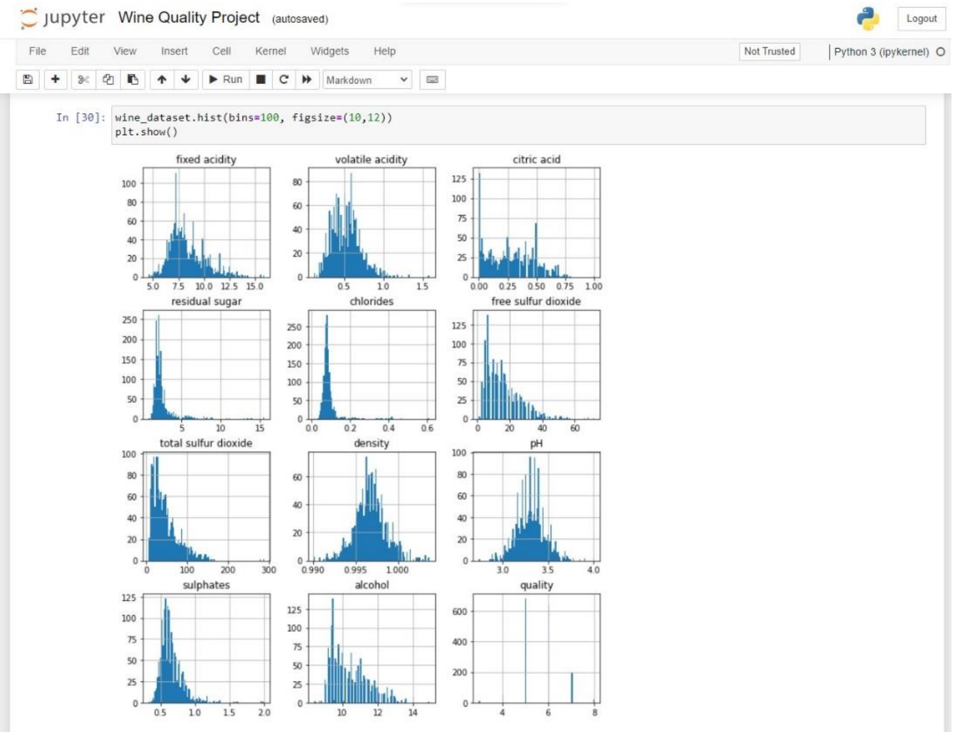
```
In [5]: # number of values for each quality
        sns.catplot(x='quality', data= wine_dataset, kind = 'count')

Out[5]: <seaborn.axisgrid.FacetGrid at 0x26789b73d60>
```

```
In [6]: # volatile acidity vs Quality
        plot = plt.figure(figsize=(5,5))
        sns.barplot(x='quality', y = 'volatile acidity', data = wine_dataset)

Out[6]: <AxesSubplot:xlabel='quality', ylabel='volatile acidity'>
```



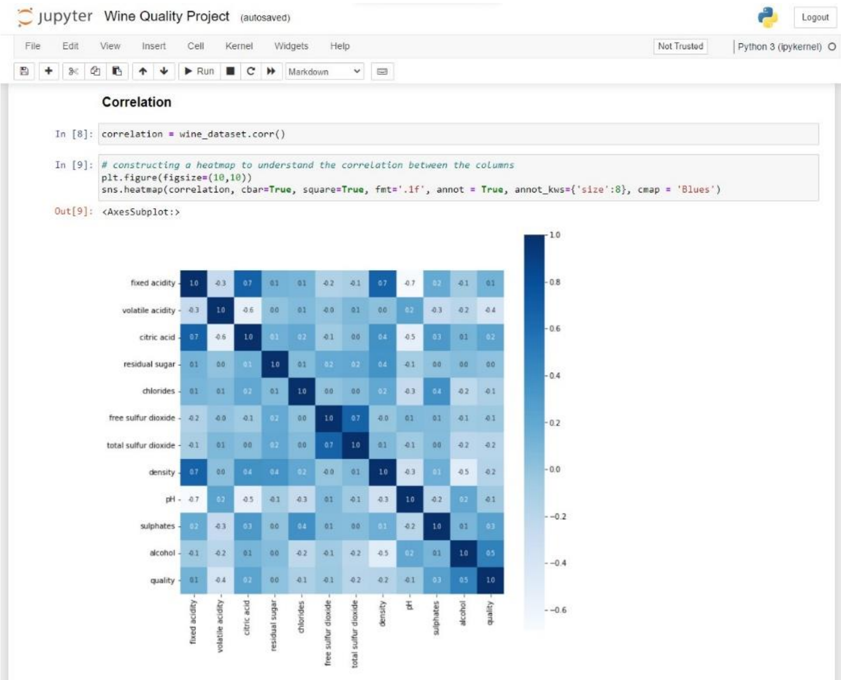
```
In [7]: # citric acid vs Quality
        plot = plt.figure(figsize=(5,5))
        sns.barplot(x='quality', y = 'citric acid', data = wine_dataset)

Out[7]: <AxesSubplot:xlabel='quality', ylabel='citric acid'>
```

IMG 3



```
In [30]: wine_dataset.hist(bins=100, figsize=(10,12))
         plt.show()
```

IMG 4



Correlation

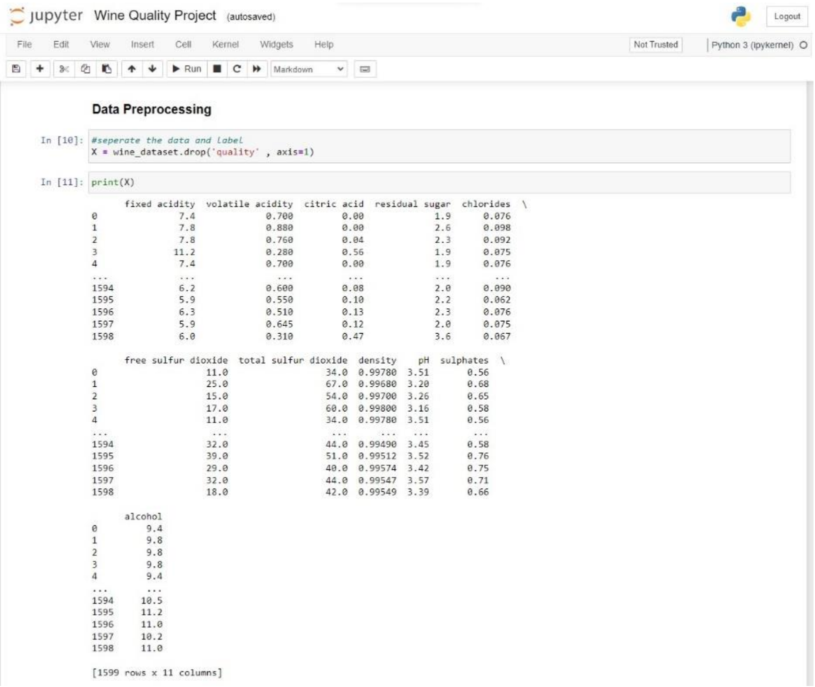
```
In [8]: correlation = wine_dataset.corr()
```

```
In [9]: # constructing a heatmap to understand the correlation between the columns
        plt.figure(figsize=(10,10))
        sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot = True, annot_kws={'size':8}, cmap = 'Blues')
```

```
Out[9]: <AxesSubplot:>
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.0	-0.3	0.7	0.1	0.1	-0.2	-0.1	0.7	-0.7	0.2	-0.1	0.1
volatile acidity	-0.3	1.0	-0.6	0.0	0.1	-0.0	0.1	0.0	0.2	-0.3	-0.2	-0.4
citric acid	0.7	-0.6	1.0	0.1	0.2	-0.1	0.0	0.4	-0.5	0.3	0.1	0.2
residual sugar	0.1	0.0	0.1	1.0	0.1	0.2	0.2	0.4	-0.1	0.0	0.0	0.0
chlorides	0.1	0.1	0.2	0.1	1.0	0.0	0.0	0.2	-0.3	0.4	-0.2	-0.1
free sulfur dioxide	-0.2	-0.0	-0.1	0.2	0.0	1.0	0.7	-0.0	0.1	0.1	-0.1	-0.1
total sulfur dioxide	-0.1	0.1	0.0	0.2	0.0	0.7	1.0	0.1	-0.1	0.0	-0.2	-0.2
density	0.7	0.0	0.4	0.4	0.2	-0.0	0.1	1.0	-0.3	0.1	-0.5	-0.2
pH	-0.7	0.2	-0.5	-0.1	-0.3	0.1	-0.1	-0.3	1.0	-0.2	0.2	-0.1
sulphates	0.2	-0.3	0.3	0.0	0.4	0.1	0.0	0.1	-0.2	1.0	0.1	0.3
alcohol	-0.1	-0.2	0.1	0.0	-0.2	-0.1	-0.2	-0.5	0.2	0.1	1.0	0.5
quality	0.1	-0.4	0.2	0.0	-0.1	-0.1	-0.2	-0.2	-0.1	0.3	0.5	1.0

IMG 5



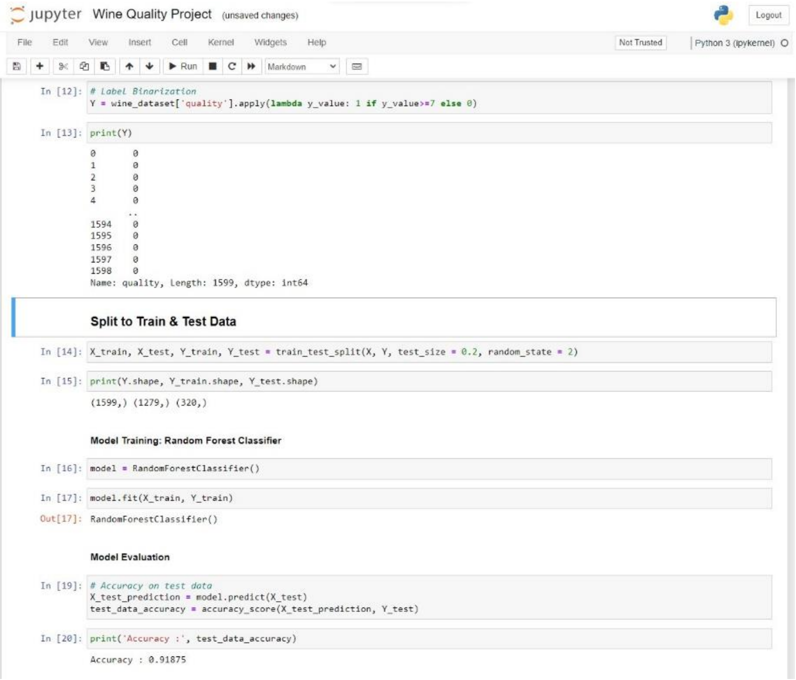
Data Preprocessing

```
In [10]: #seperate the data and label
         X = wine_dataset.drop('quality' , axis=1)
```

```
In [11]: print(X)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56
1                    25.0                  67.0  0.99680  3.20       0.68
2                    15.0                  54.0  0.99700  3.26       0.65
3                    17.0                  60.0  0.99800  3.16       0.58
4                    11.0                  34.0  0.99780  3.51       0.56
...                   ...                   ...      ...   ...        ...
1594                 32.0                  44.0  0.99490  3.45       0.58
1595                 39.0                  51.0  0.99512  3.52       0.76
1596                 29.0                  40.0  0.99574  3.42       0.75
1597                 32.0                  44.0  0.99547  3.57       0.71
1598                 18.0                  42.0  0.99549  3.39       0.66

      alcohol
0         9.4
1         9.8
2         9.8
3         9.8
4         9.4
...       ...
1594     10.5
1595     11.2
1596     11.0
1597     10.2
1598     11.0

[1599 rows x 11 columns]
```

IMG 6



```
In [12]: # Label Binarization
         Y = wine_dataset['quality'].apply(lambda y_value: 1 if y_value>=7 else 0)

In [13]: print(Y)

         0       0
         1       0
         2       0
         3       0
         4       0
                ..
         1594    0
         1595    0
         1596    0
         1597    0
         1598    0
         Name: quality, Length: 1599, dtype: int64
```

Split to Train & Test Data

```
In [14]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 2)

In [15]: print(Y.shape, Y_train.shape, Y_test.shape)

         (1599,) (1279,) (320,)
```

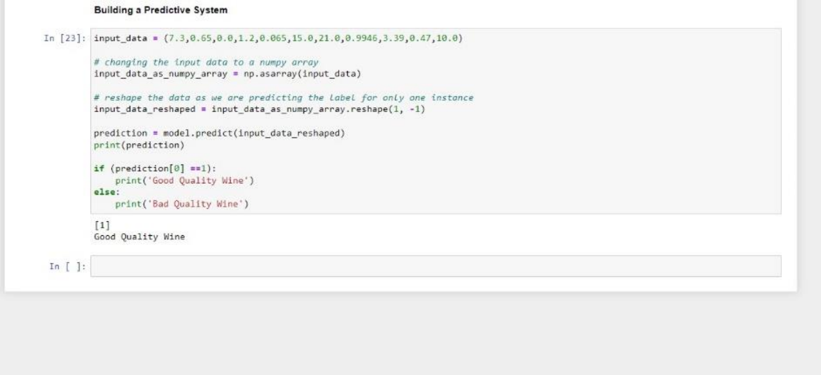
Model Training: Random Forest Classifier

```
In [16]: model = RandomForestClassifier()

In [17]: model.fit(X_train, Y_train)

Out[17]: RandomForestClassifier()
```

Model Evaluation

```
In [19]: # Accuracy on test data
         X_test_prediction = model.predict(X_test)
         test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [20]: print('Accuracy :', test_data_accuracy)

         Accuracy : 0.91875
```

IMG 7



Building a Predictive System

```
In [23]: input_data = (7.3,0.65,0.0,1.2,0.065,15.0,21.0,0.9946,3.39,0.47,10.0)

         # changing the input data to a numpy array
         input_data_as_numpy_array = np.asarray(input_data)

         # reshape the data as we are predicting the label for only one instance
         input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)

         prediction = model.predict(input_data_reshaped)
         print(prediction)

         if (prediction[0] ==1):
             print('Good Quality Wine')
         else:
             print('Bad Quality Wine')

         [1]
         Good Quality Wine

In [ ]:
```

IMG 8

REFERENCE

- Websites referred

♣ www.kaggle.com

♣ www.pythonprogramming.net

♣ https://scikit-learn.org/

♣ www.geekforgeeks.com

♣ www.google.co.in

- Books referred

♣ Python Programming - Kiran Gurbani

♣ Learning Python - Mark Lutz

- YouTube Channels referred

♣ codewithharry

♣ edureka!

<u>THANK YOU</u>