Project Based Learning:

Predicting Wine Quality using Wine Quality Dataset

Parth Nautiyal
B. Tech (CSE (ML and AI)),
Graphic Era Deemed to be University
parthnautiyal1016@gmail.com

Abstract / Motivation

Focus of this project is using machine learning to build a model which can predict the quality of the red wine data provided. Motivation for building this model are twofold. First, to apply and test my knowledge of what I have learned in the machine learning class. Secondly, to build a model using Random Forest algorithm which can surpass the accuracy when the same model is applied using a decision tree algorithm.

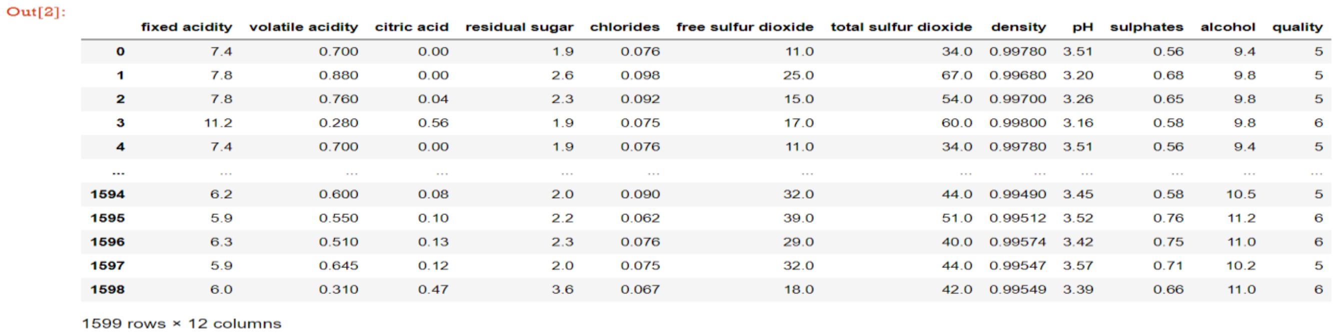
1 Introduction

In the recent years there has been an increase in the production of wine and with huge production comes a huge dataset. And, with the advent of new technologies and machine learning algorithms, we can decrease the rigorous task of humans predicting the wine quality with a huge list of recorded pH values, densities, acidity, etc. Here, I am intending to use Machine Learning to build a classifier which can take in features like fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free Sulphur dioxide, total Sulphur dioxide, density, pH, sulphates, alcohol and produce as output a prediction of the quality of red wine. Furthermore, the ability to produce an accurate prediction of the quality of the red wine has significant potential in the wine business.

Approach:

Data Set:

The following dataset represents red wine quality. With features as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free Sulphur dioxide, total Sulphur dioxide, density, pH, sulphates, alcohol and quality column representing quality of the red wine.
Reference of the dataset:
https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/download



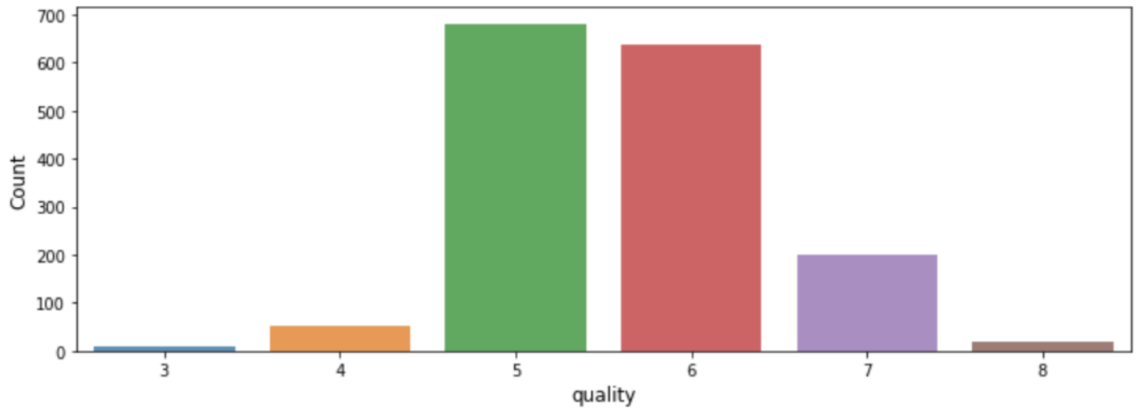
Out[2]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

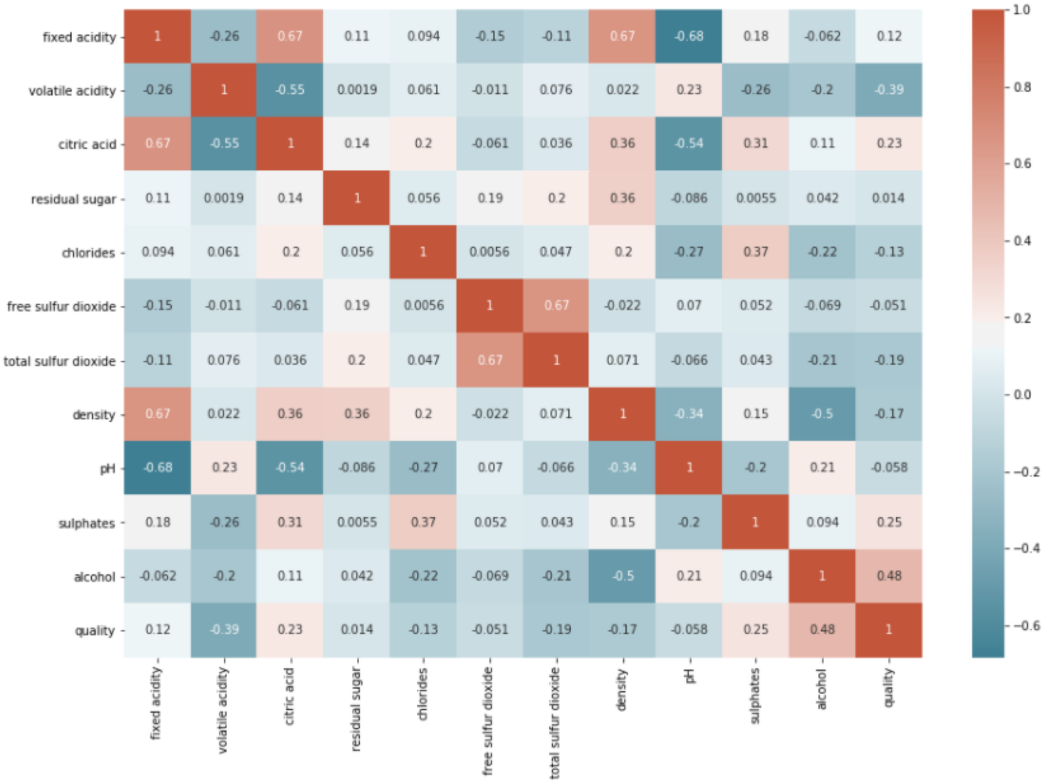
1599 rows × 12 columns

Data Visualization:

1. Plotted the bar graph of the quality feature vs count. Using seaborn library module.



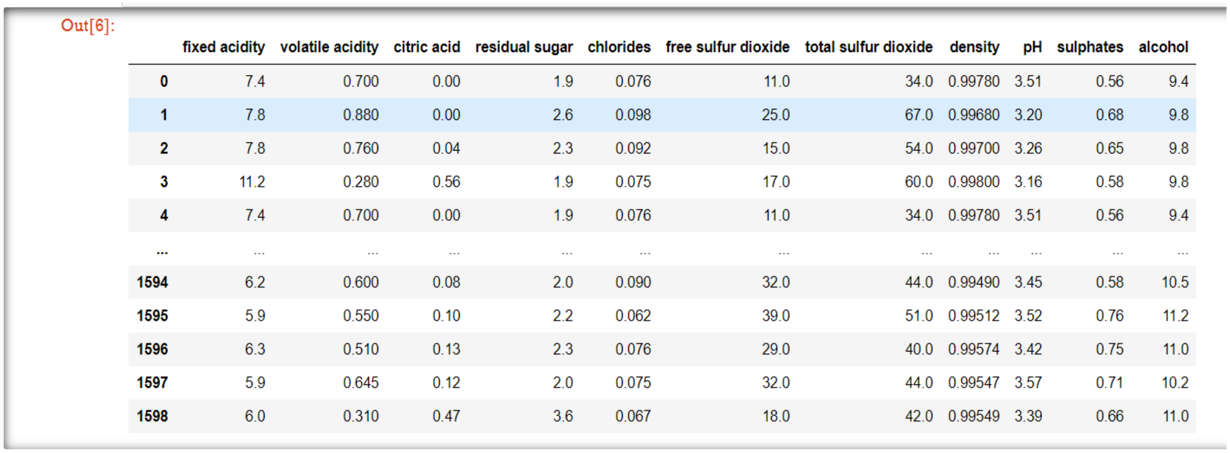
2. *Plotting correlation matrix:*



***Pre-Processing Data*:**

1. Create Classification version of target variable, by converting quality values to binary i.e., 0 or 1.

2. Separated feature variables and target variable



Out[6]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0

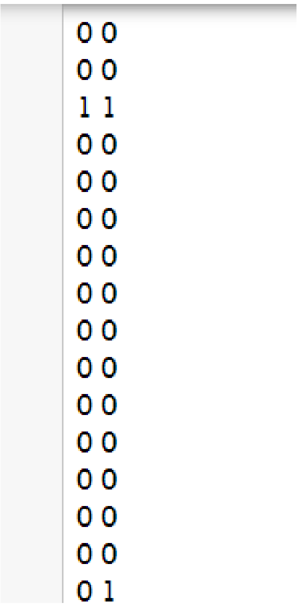
3. Using sklearn library, for Standard Scaling of the data. So that the difference between values is reduced.

4. Splitting the data into training and testing, 75% and 25% respectively.

Algorithm: Random Forest Classifier.

Definition: A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

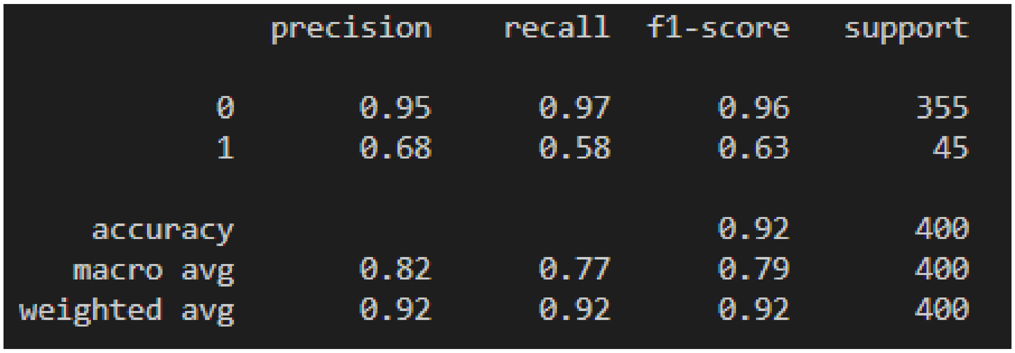
After implementing random forest algorithm, we have the following prediction results of first 16 rows:



```
0 0
0 0
1 1
0 0
0 0
0 0
0 0
0 0
0 0
0 0
0 0
0 0
0 0
0 0
0 0
0 1
```

Accuracy of Model:

Now, in order to check the accuracy of our model we use module of sklearn.metrics and print the classification report:



```
              precision    recall  f1-score   support

           0       0.95      0.97      0.96       355
           1       0.68      0.58      0.63        45

    accuracy                           0.92       400
   macro avg       0.82      0.77      0.79       400
weighted avg       0.92      0.92      0.92       400
```

Our model gives an Accuracy score of 0.92 which is 92%, which is considered as a good prediction.