Wine Quality Prediction using Machine Learning Algorithms

A Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Engineering / Electronics & Communication Engineering

By

19K41A05B1	S. Shiva Keerthi
19K41A04H5	S. Ramya Teja
19K41A05B0	Ramsha Mehreen

Under the Guidance of
Dr. V. Venkataramana

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S.R.ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA) Dec-2021.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled "Wine Quality Prediction" is a record of bonafide work carried out by the student(s) S. Shiva Keerthi, S. Ramya Teja, Ramsha Mehreen bearing Roll No(s) 19K41A05B1, 19K41A04H5, 19K41A05B0 during the academic year 2020-21 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering/Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

.

Supervisor **Head of the Department**

External Examiner

ABSTRACT

Wine classification is a difficult task since taste is the lease understood of the 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 Logistic Regression, Decision Tree and Random Forest.

Table of Contents

1. 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.

The aim of this project is to predict the quality of wine whether it is good or bad with given a set of features as inputs. The dataset used is Wine Quality Classification Data set from Kaggle Website. Input variables are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulphur dioxide, total sulphur dioxide, density, sulphates, alcohol. And the output variable is quality (0-Bad and 1-Good).

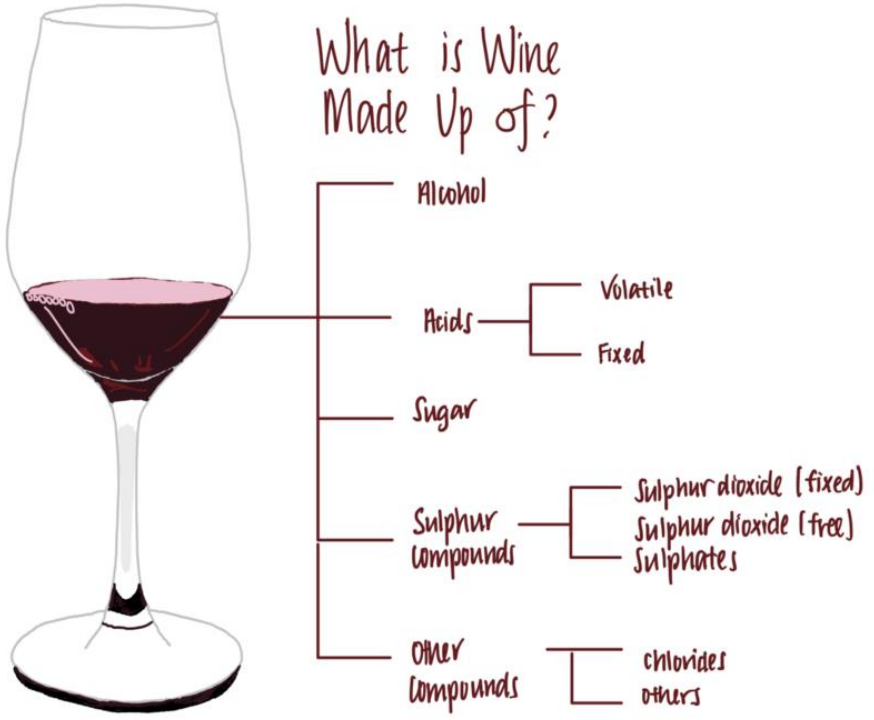


Figure 1: Visualization of the wine.

2. LITERATURE SURVEY

Machine learning is helpful for a variety of situations. The prediction of dependent variable values from independent variables is one of the uses of this methodology. Wine Quality and style are highly influenced by the qualitative and quantitative composition of aromatic compounds having various chemical structures and properties and their interaction within different wine matrices. The understanding of interactions between the wine matrix and volatile compounds and impact on the overall flavor as well as on typical or specific aromas is getting more and more important for the creation of certain wine styles. Machine learning methods can nonetheless be helpful in solving this challenge and in anticipating danger early. Some of the techniques used for such prediction problems are logistic regression, Decision Tree, and Random Forest algorithms.

3. DESIGN:

3.1 REQUIREMENT SPECIFICATION(S/W & H/W)

Hardware Requirements

- ✓ **System** : Processor Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, 1190 MHz, 4 Core(s), 8 Logical Processor(s)
- ✓ **RAM** : 8GB
- ✓ **Hard Disk** : 557GB
- ✓ **Input** : Keyboard and Mouse
- ✓ **Output** : PC

Software Requirements

- ✓ **OS** : Windows 10
- ✓ **Platform** : Google Colaboratory, Jupiter Notebook
- ✓ **Web Application** : Stream lit
- ✓ **Program Language** : Python

3.2 Flow chart

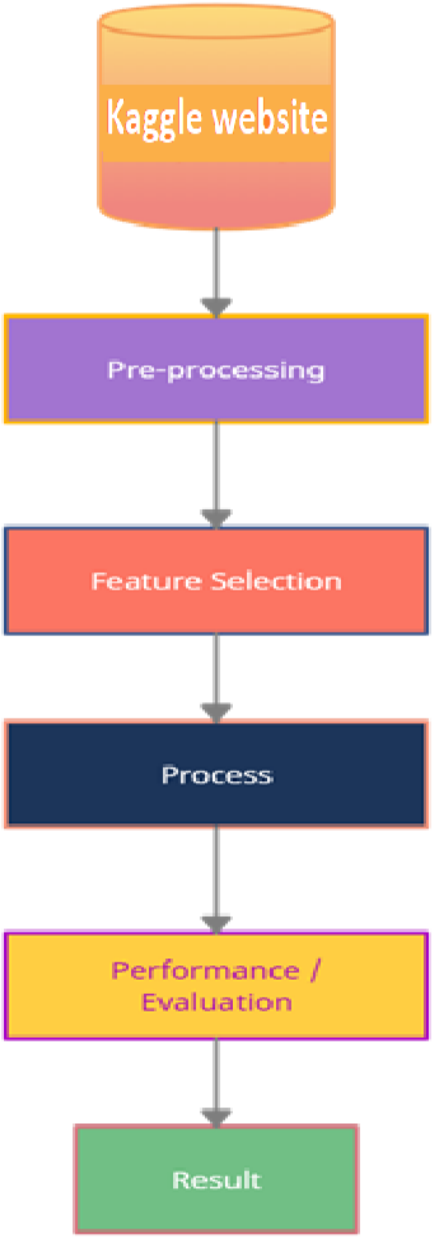


Figure 2: flow chart

This is an analysis of the wine prediction problem using the different machine learning models. It predicts whether the wine produced is a good quality wine or not. For Analysis such model we use different machine learning tools like pandas, matplotlib, sci-kit-learn, etc.

4. DATASET:

We collected the data from the Kaggle website. We use only 10 attributes to predict whether wine is good or bad.

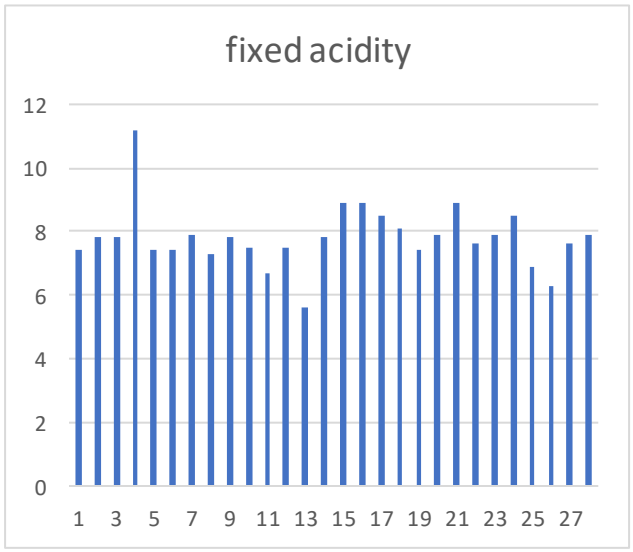
Input features:

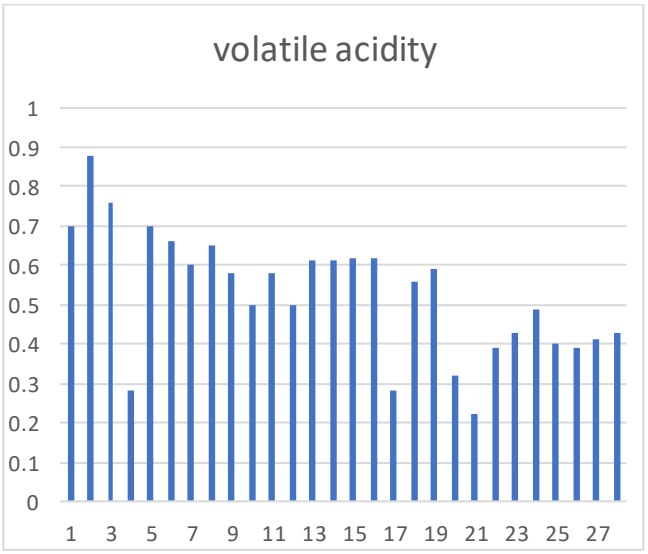
- Fixed acidity : are non-volatile acids that do not evaporate readily
    - Value : Numeric input
- Volatile acidity : are high acetic acid in wine which leads to an unpleasant vinegar taste
    - Value : Numeric input
- Citric acid : acts as a preservative to increase acidity (small quantities add freshness and flavor to wines)
    - Value : Numeric input
- Residual sugar : is the amount of sugar remaining after fermentation stops. The key is to have a perfect balance between — sweetness and sourness (wines > 45g/ltrs are sweet).
    - Value : Numeric input
- Chlorides : the amount of salt in the wine
    - Value : Numeric input
- Free sulfur dioxide : it prevents microbial growth and the oxidation of wine
    - Value : Numeric input
- Total sulfur dioxide : is the amount of free + bound forms of SO2
    - Value : Numeric input
- Density : sweeter wines have a higher density
    - Value : Numeric input
- Sulfates : a wine additive that contributes to SO2 levels and acts as an antimicrobial and antioxidant
    - Value : Numeric input
- Alcohol : the amount of alcohol in wine
    - Value : Numeric input

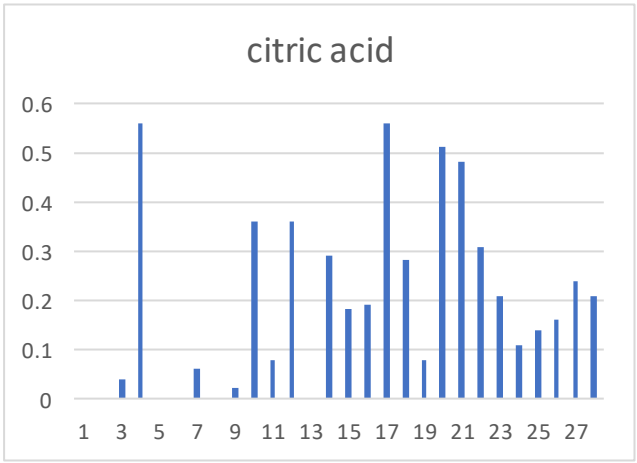
Output feature:

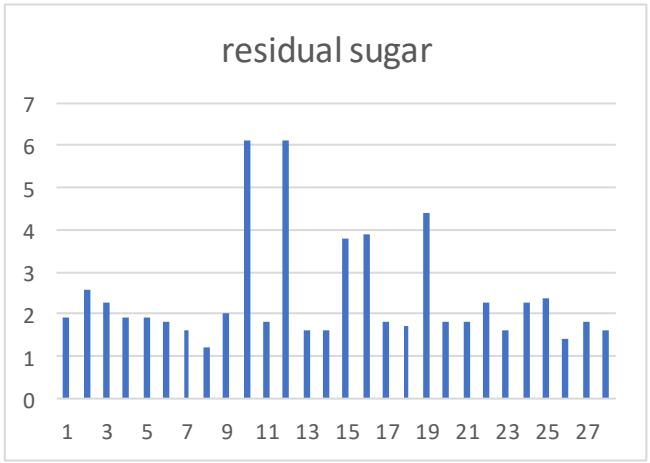
- Quality
    - Value : Good/Bad (Binary output)

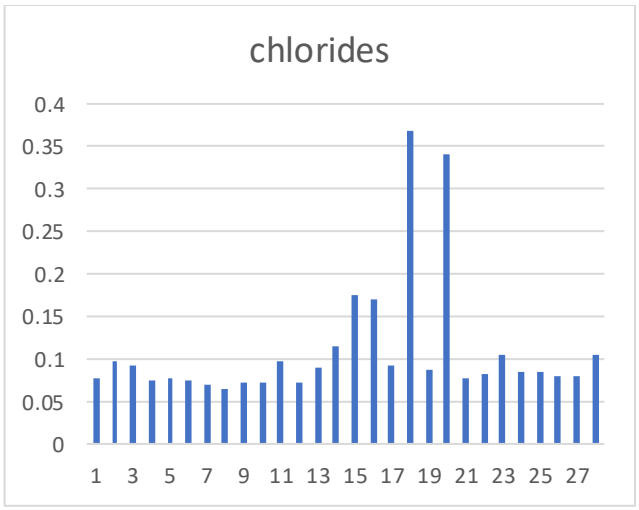
In total, we have 10 input features and 1 output feature i.e., 11 features for Wine Quality Prediction.

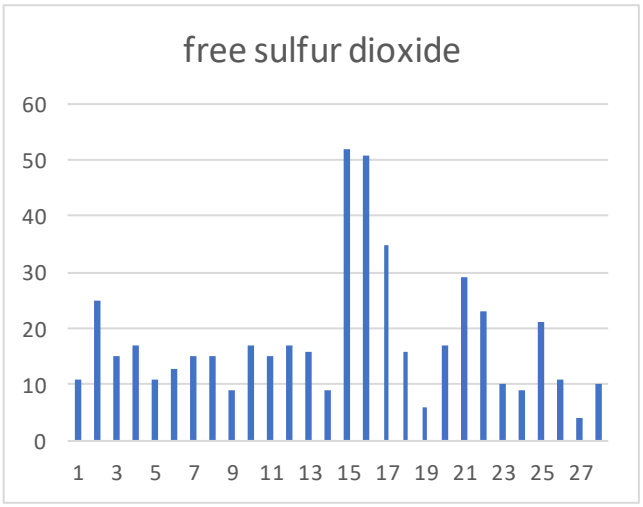


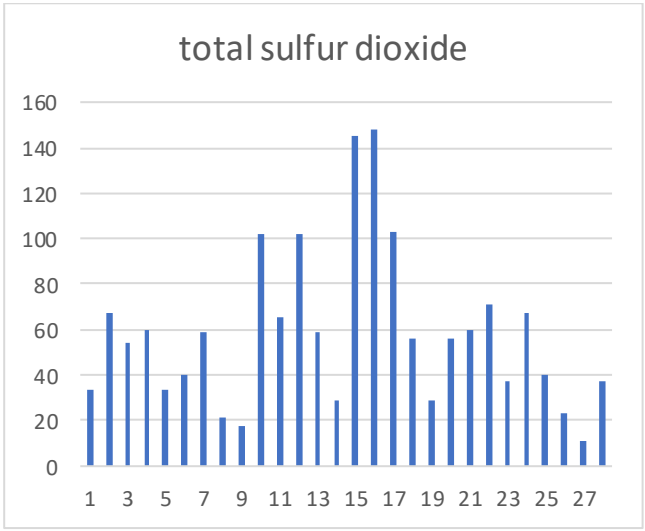


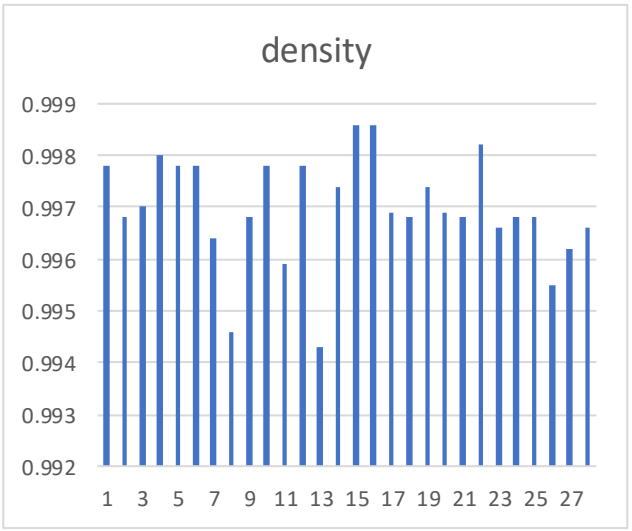


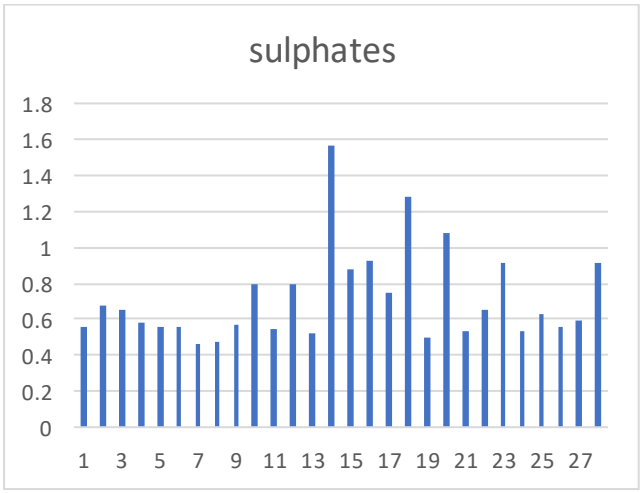



chlorides
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27


free sulfur dioxide
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27


total sulfur dioxide
160
140
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27


density
0.999
0.998
0.997
0.996
0.995
0.994
0.993
0.992
1 3 5 7 9 11 13 15 17 19 21 23 25 27


sulphates
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27

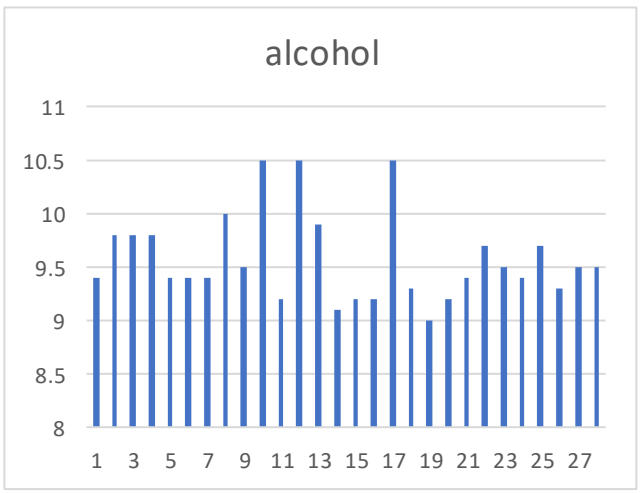

alcohol
11
10.5
10
9.5
9
8.5
8
1 3 5 7 9 11 13 15 17 19 21 23 25 27

Figure 3: Visualizing attributes of the dataset

5. DATA PREPROCESSING:

The wine prediction dataset is taken from Kaggle website it contains a large collection of datasets that have been used for the machine learning community. The wine dataset contains 1599 instances. And dataset have 10 input variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, sulfates, alcohol, and 1 output variable: quality (0-bad and 1-good). Below Table 1 description of the attributes.

Fixed acidity	Numeric [range=0 to 15.9 to 1.58;mean=8.319]
Volatile acidity	Numeric [range=0.12 to 1.58;mean=0.527]
Citric acid	Numeric [range= 0 to 1;mean=0.270]
Residual sugar	Numeric [range= 0.9 to 15.5;mean=2.538]
Chlorides	Numeric [range= 0.012 to 0.611;mean=0.087]
Free sulfur dioxide	Numeric [range= 1 to 72;mean=15.874]
Total sulfur dioxide	Numeric [range= 6 to 289;mean=46.467]

Density	Numeric [range= 0.99 to 1.003;mean=0.996]
Sulfates	Numeric [range= 0.33 to 2;mean=0.658]
Alcohol	Numeric [range= 8.4 to 14.9;mean=10.422]
Quality	Numeric [range= 0 or 1;mean=0.534]

Table 1: Kaggle Dataset range and datatype.

Data info:

```
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   sulphates             1599 non-null   float64
 9   alcohol               1599 non-null   float64
 10  quality               1599 non-null   int64
dtypes: float64(10), int64(1)
```

Missing values treatment:

The real world's dataset often has many missing values which can be treated by using certain methods. But in our dataset, there are no missing values, because we collected the data from Kaggle website directly. To treat the missing values, we generally use the following strategies:

- Remove the entire row (If missing values are less in number)
- Replace the missing value with either mean or median
- Replace the missing value with most frequent value in the column (This is generally used only for large dataset)

```
print(data.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
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Figure 4: Printing missing values from dataset

Outliers detection and treatment:

Outliers are data points that don't fit the pattern of rest of the numbers. They are the extremely high or extremely low values in the data set. A simple way to find an outlier is to examine the numbers in the data set. We can also detect outliers by Z-score method, its formula is given by**:**

- **Z = (x- μ)/ σ**
    - Where, x is each value in dataset,
    - μ is mean
    - σ is standard deviation

Normalization:

Normalization is a technique for organizing data in a database. Data normalization is the process of rescaling one or more attributes to the range of 0 to 1. This means that the largest value for each attribute is 1 and the smallest value is 0. It is important that a database is normalized to ensure only related data is stored in each table and to avoid biasing towards huge values. When we normalize the data while feeding it to the model, we also have to de-normalize it. This process can be done using the formulas below:

- $x_{nor} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$
- $y_i = y_{nor}(y_{max} - y_{min}) + y_{min}$

Data split:

To train any machine learning model irrespective what type of dataset is being used, we have to split the dataset into training and testing data. The reason to split the data is to give the machine learning model an effective mapping of input to outputs and to evaluate the model performance. We pass the training data to train our machine learning model and then test the model on testing data. In our model, we used 70:30 data split. That is the data is split 70% for training and 30% for testing. We can do the data split using train_test_split module in python.

Correlation Matrix

A correlation matrix is simply a table that displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a scatterplot. coefficients for different variables.

How it is calculated?

A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable (X_i) in the table is correlated with each of the other values in the table (X_j)... The diagonal of the table is always a set of ones because the correlation between a variable and itself is always 1. Let's perform the Correlation matrix to understand the relation between the dependent variable and the independent variable and within the independent variable.

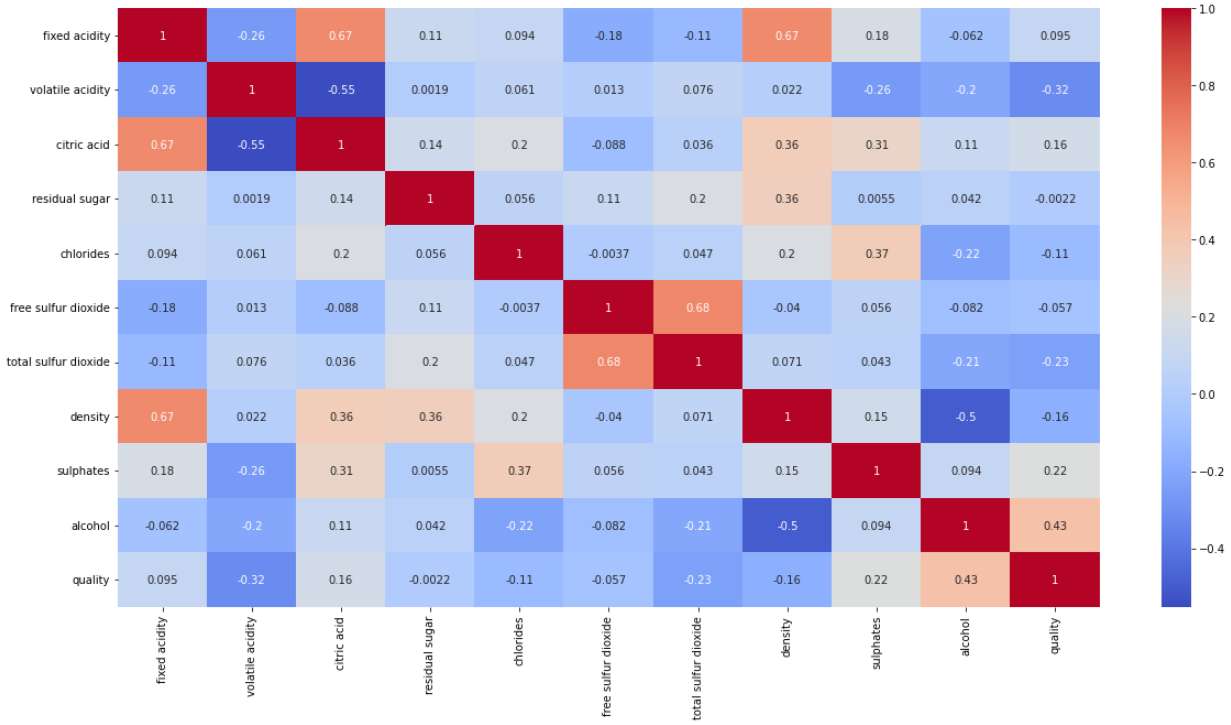


Figure 5: Correlation Matrix

6. METHODOLOGY:

This section talks about the algorithms used for the project. We used three different algorithms like Logistic Regression, Decision Tree and Random forest.

Logistic Regression

Logistic regression is a supervised algorithm for study classification. The likelihood of a destination variable was predicted. The nature of the target or dependent variable is dichotomous, meaning that only two possible classes are available. Here, we use binary logistic regression which uses the sigmoid function to convert real numbers into binary and to calculate accuracies we use binary cross entropy loss function.

Steps for Logistic Regression:

Step 1: Read the data

Step 2: Data pre-processing (Normalization, missing values treatment, outliers)

Step 3: Initialize the model parameters, number of iterations and eta

Step 4: Gradient calculation (For all iterations and data samples)

Step 5: Read the model parameters

Step 6: Print confusion matrix and accuracy for training and testing data

Step 7: Model performance evaluation and deployment

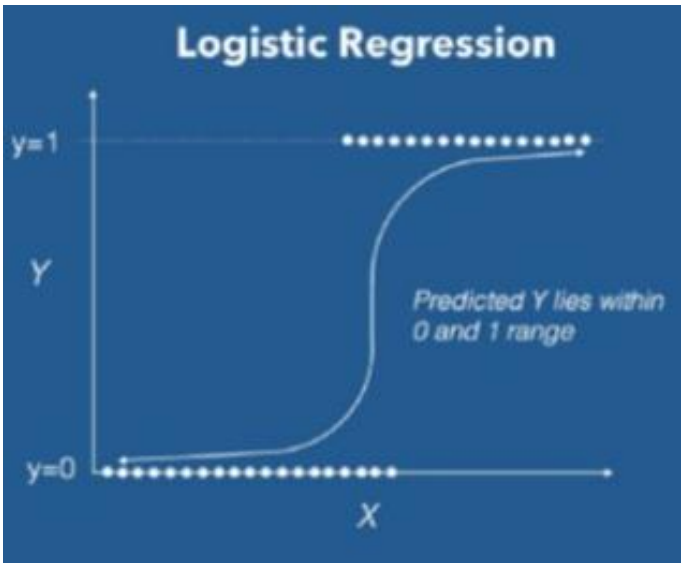


Figure 6: Logistic Regression

Decision tree

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label**.** Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated for the sub tree rooted at the new node. We can perform decision tree using entropy and gini index.

Steps for decision tree using entropy:

Step 1: Determine the Root of the Tree.

Step 2: Calculate Entropy for The Classes.

Step 3: Calculate Entropy After Split for Each Attribute.

Step 4: Calculate Information Gain for each split.

Step 5: Perform the Split.

Step 6: Perform Further Splits until leaf nodes

Step 7: Complete the Decision Tree.

The decision trees of our model using entropy and gini index are shown below:

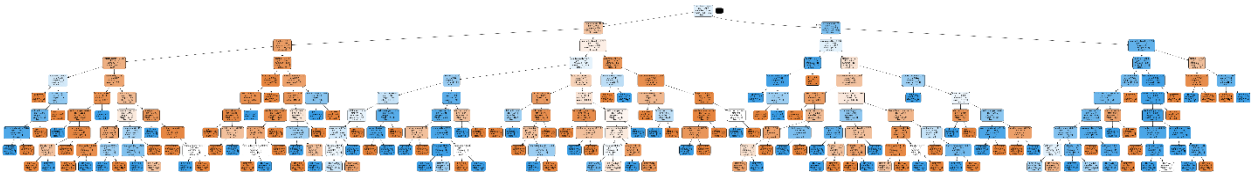


Figure 7: Decision Tree

Random forest:

A random forest is a supervised machine learning algorithm that is constructed from decision tree algorithms. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. Random forest establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the over fitting of datasets and increases precision.

Steps of random forest:

Step 1: Selection of random samples from the dataset.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output

Step 4: Final output is considered on majority voting or averaging

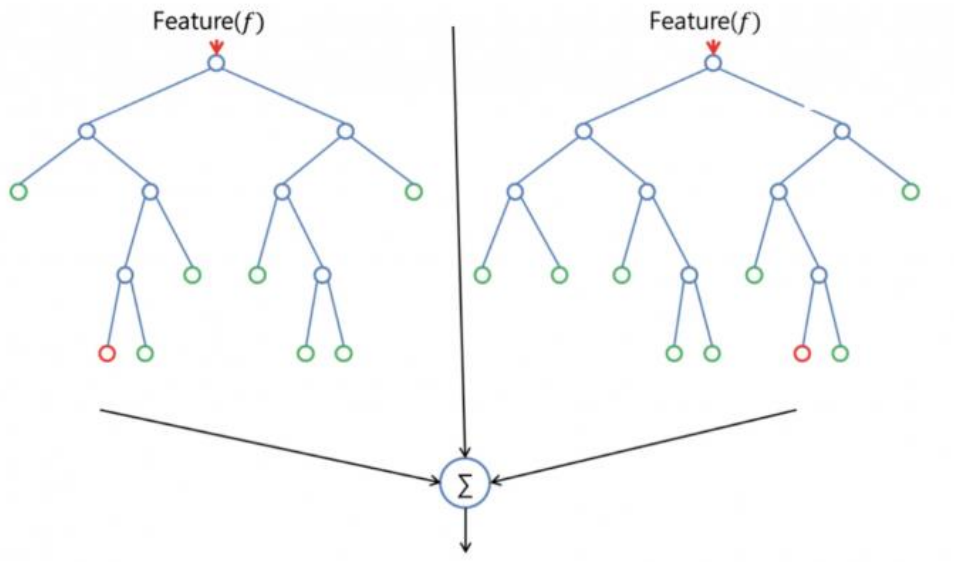


Figure 8: Random forest classifier

7. RESULTS:

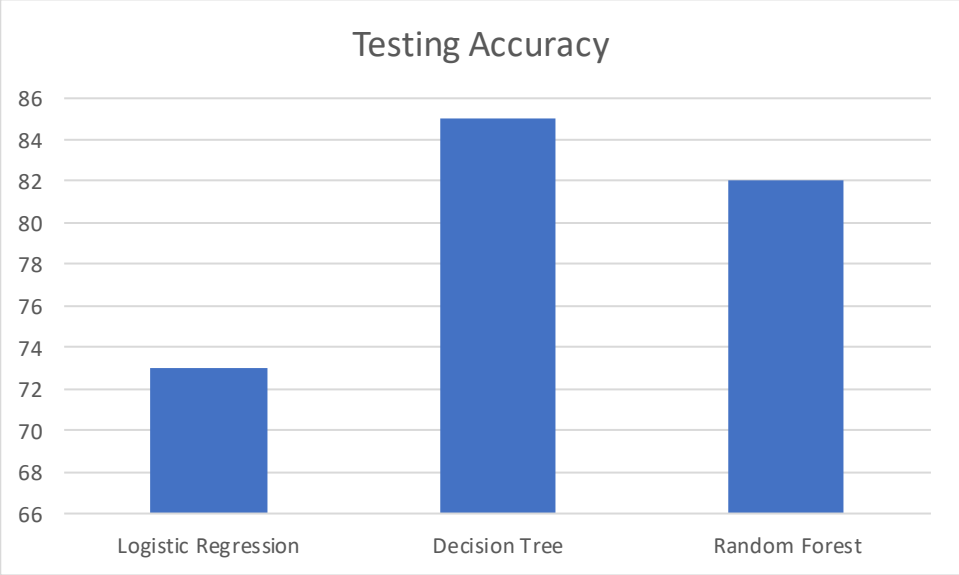


Figure 9: Graph depicting the accuracies of all models

Algorithm	**Training error**	**Testing error**
Logistic Regression	74%	73%
Decision tree	99%	85%
Random Forest	94%	82%

Table 1: Accuracy of all models

Cost	**1 - 4**	**5**	**6**	**7**	**None**
Entropy	0.699	0.789	0.815	0.832	1.0
Gini	0.699	0.78	0.729	0.727	1.0

Table 2: Decision tree training accuracy table

Cost	**1 - 4**	**5**	**6**	**7**	**None**
Entropy	0.699	0.718	0.78	0.852	0.76
Gini	0.699	0.725	0.725	0.747	0.76

Table 3: Decision tree testing accuracy table

Cost	1	2	3	4	None
Entropy	0.897	0.892	0.957	0.955	1.0
Gini	0.879	0.895	0.952	0.950	1.0

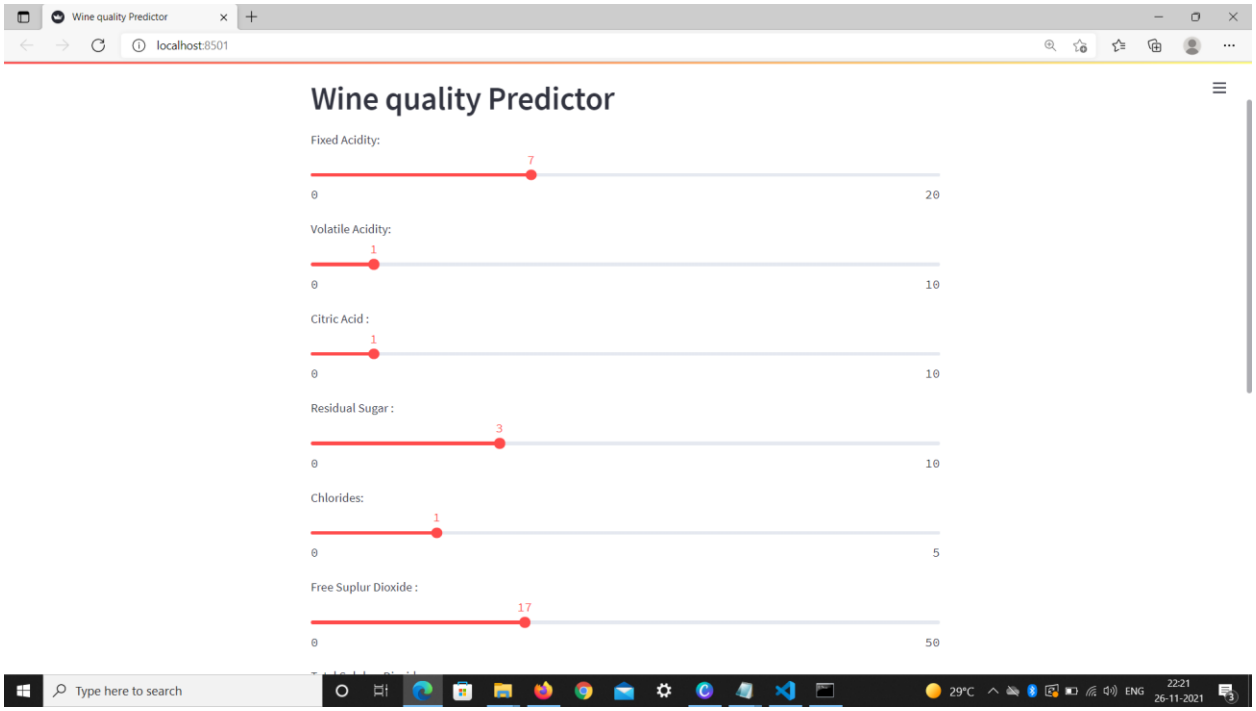
Table 4: Random forest training accuracy table

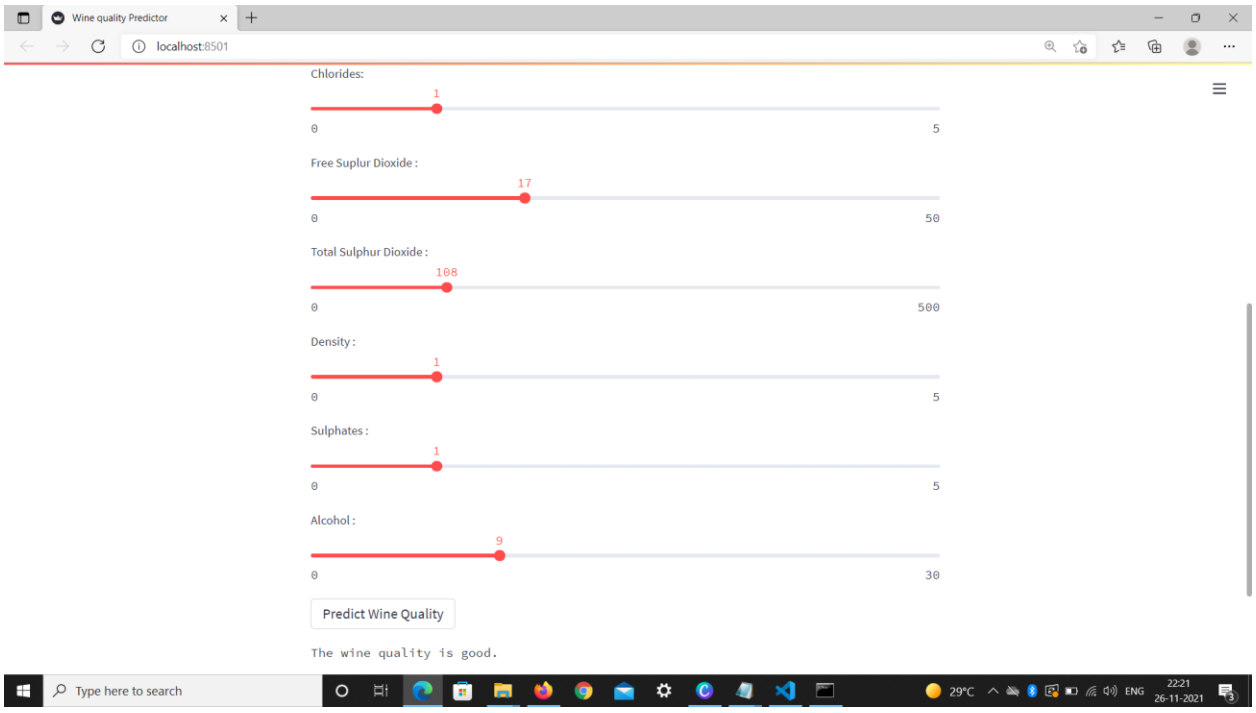
Cost	1	2	3	4	None
Entropy	0.731	0.691	0.781	0.735	0.82
Gini	0.718	0.714	0.768	0.766	0.81

Table 5: Random forest testing accuracy table

From the above tables, we can see that the highest accuracy is for Decision tree with 85%. Hence, we use it to deploy our model. The other models are also showing good accuracies of 82% and 73% for Random forest and Logistic regression models respectively.

8. **WEB APPLICATION:** Using streamlit software





9. CONCLUSION:

Based on the above accuracy tables, we can conclude that for this project, decision tree has given us the highest accuracy of 85%. We have developed the web application using this decision tree model. There is a further scope of improvement with accuracy of the model, we can try several other models to improve the accuracy.

10. REFERENCES:

[1] Yunhui Zeng1 , Yingxia Liu1 , Lubin Wu1 , Hanjiang Dong1. "Evaluation and Analysis Model of Wine Quality Based on Mathematical Model ISSN 2330-2038 E-ISSN 2330-2046,Jinan University, Zhuhai,China.

[2] Paulo Cortez1, Juliana Teixeira1, Ant´onio Cerdeira2."Using Data Mining for Wine Quality Assessment".

[3] Yesim Er*1 , Ayten Atasoy1. "The Classification of White Wine and Red Wine According to Their Physicochemical 2147-67992147-6799,3rd September 2016

[4] Ebeler S. (1999) "Flavor Chemistry — Thirty Years of Progress: chapter Linking flavour chemistry to sensory analysis of wine". Kluwer Academic Publishers, 409–422.

[5] Legin, Rudnitskaya, Luvova, Vlasov, Natale and D'Amico. (2003) "Evaluation of Italian wine by the electronic tongue: recognition, quantitative analysis and correlation with human sensory perception". Analytica Chimica Acta 484 (1): 33–34.

[6] Sun, Danzer and Thiel. (1997) "Cl`assification of wine samples by means of artificial neural networks and discrimination analytical methods". Fresenius Journal of Analytical Chemistry 359 (2) 143–149.

[7] Vlassides, Ferrier and Block. (2001) "Using historical data for bioprocess optimization: modeling wine characteristics using artificial neural networks and archived process information". Biotechnology and Bioengineering 73 (1) 55-68.

[8] Moreno, Gonzalez-Weller, Gutierrez, Marino, Camean, Gonzalez and Hardisson. (2007) "Differentiation of two Canary DO red wines according to their metal content from inductively coupled plasma optical emission spectrometry and graphite furnace atomic absorption spectrometry by using Probabilistic Neural Networks". Talanta 72 263–268.

[9] Yu, Lin, Xu, Ying, Li and Pan. (2008) "Prediction of Enological Parameters and Discrimination of Rice Wine Age Using Least-Squares Support Vector Machines and Near Infrared Spectroscopy". Agricultural and Food Chemistry 56 307–313.

[10] Beltran, Duarte-Mermoud, Soto Vicencio, Salah and Bustos. (2008) “Chilean Wine Classification Using Volatile Organic Compounds Data Obtained With a Fast GC Analyzer”. IEEE Transactions on Instrumentation and Measurement 57 2421-2436.

[11] Cortez, Cerdeira, Almeida, Matos and Reis. (2009) “Modeling wine preferences by data mining from physicochemical properties”. Decision Support Systems 47 547-553.

[12] Jambhulkar and Baporikar. (2015) “Review on Prediction of Heart Disease Using Data Mining Technique with Wireless Sensor Network”. International Journal of Computer Science and Applications 8 (1) 55-59.

[13] Zaveri, and Joshi. (2017) “Comparative Study of Data Analysis Techniques in the domain of medicative care for Disease Predication”. International Journal of Advanced Research in Computer Science 8 (3) 564-566.

[14] Portuguese Wine - Vinho Verde. Comissa˜o de Viticultura da Regia˜o dos Vinhos Verdes (CVRVV), http://www.vinhoverde.pt, July 2008