RED AUTHENTICITÉ

Using Neural Networks to develop a Prediction Model for predicting the Quality of Red Wine

A PROJECT REPORT

Submitted By

HIMANSHU BERIWAL (Reg No: RA1811028010007)

RUSHIL RAI (Reg No: RA1811028010022)

Under the guidance of

Ms S.Priya

(Assistant Professor (Sr.G), Department of Computer Science & Engineering)

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

November 2020

ACKNOWLEDGEMENTS

We wish to thank **Dr. B. Amutha**, Head of Department, Department of Computer Science Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor **Dr. S.Ganesh Kumar**, Associate Professor, Department of Computer Science Engineering, SRM Institute of Science and Technology, for her great support at all the stages of project work.

Our inexpressible respect and thanks to our guide and also our Faculty Advisor, **Ms S.Priya**, Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing an opportunity to pursue the project under her mentorship. She provided us with utmost help and opportunities to explore. She has been a very passionate mentor and her passion for Computer Science and Problem Solving inspired us all through our journey.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

Himanshu Beriwal

Rushil Rai

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that the Industrial training report titled "RED AUTHENTICITÉ - Using Neural Networks to develop a Prediction Model for predicting the Quality of Red Wine" is the bonafide work of "Himanshu Beriwal [RA1811028010007]" & "Rushil Rai [RA1811028010022]" submitted for the course 18CSP103L Seminar – I. This report is a record of successful completion of the specified course evaluated based on literature reviews and the supervisor. No part of the Seminar Report has been submitted for any degree, diploma, title, or recognition before.

SIGNATURE	**SIGNATURE**
Ms. S Priya	**Dr. S.Ganesh Kumar**
Assistant Professor	**Academic Advisor/HOD**
Dept. of Computer Science	**Dept. of Computer Science**
& Engineering	**& Engineering**

ABSTRACT

Red Wine is one of the most popular and renowned Alcoholic beverages around the world.

With such demand, arises a major problem of Quality Assurance. Alongside assurance, many brands now use Quality of their products as a way of promoting them. This is why knowing the Quality of a product, especially something like Red Wine, which is a consumable needs a much more enhanced and refined way of measuring Quality.

Since, fermentation of any good quality wine takes at least a few weeks, assurance of quality becomes tougher. Physical and manual Quality testing has proven to be ineffective and insufficient in recent times.

Software solutions and Computer Science techniques like Neural Networks and Machine Learning have revolutionised the field of Quality assessment and assurance.

With this project we have tried and attempted exploration in this field. We have used Neural Networks to try and bring up an enhanced solution to Red Wine Quality Prediction.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Developing a Prediction Model using Machine Learning to determine the Quality of Red Wines distributed.

1.2 OBJECTIVES

Collecting and Maintaining a Large and Clean Dataset which is to be modelled to analyse quality of wine.

Define various parameters and fields according to which Quality shall be predicted, e.g. Fixed Acidity, Volatile Acidity, etc.

Use various Machine Learning Algorithms and Techniques which will help us rate the Quality of various different Wines.

Further classification and rectification to be done to achieve maximum results and accuracy.

1.3 APPROACHES

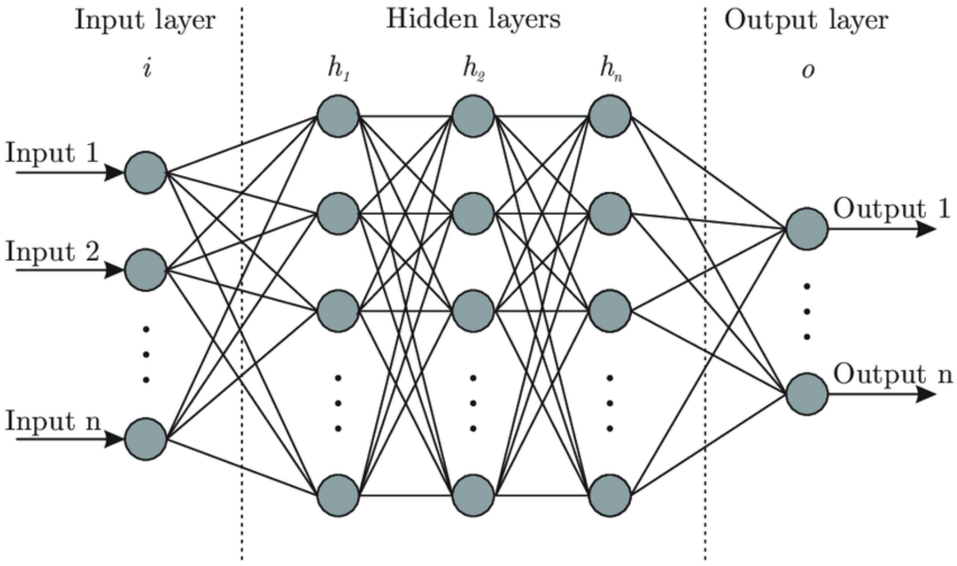
NEURAL NETWORKS :

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

A “neuron” in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

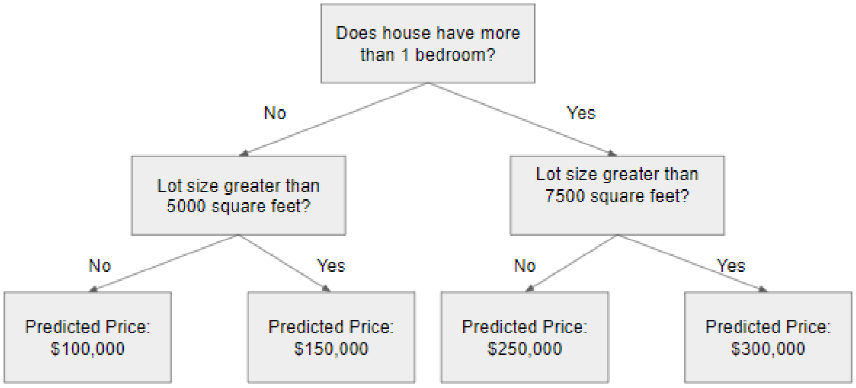


In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map.

For instance, the patterns may comprise a list of quantities for technical indicators about a security; potential outputs could be “buy,” “hold” or “sell.”

Hidden layers fine-tune the input weightings until the neural network’s margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis.

DECISION TREE :


Does house have more than 1 bedroom?
No
Yes
Lot size greater than 5000 square feet?
Lot size greater than 7500 square feet?
No
Yes
No
Yes
Predicted Price: $100,000
Predicted Price: $150,000
Predicted Price: $250,000
Predicted Price: $300,000

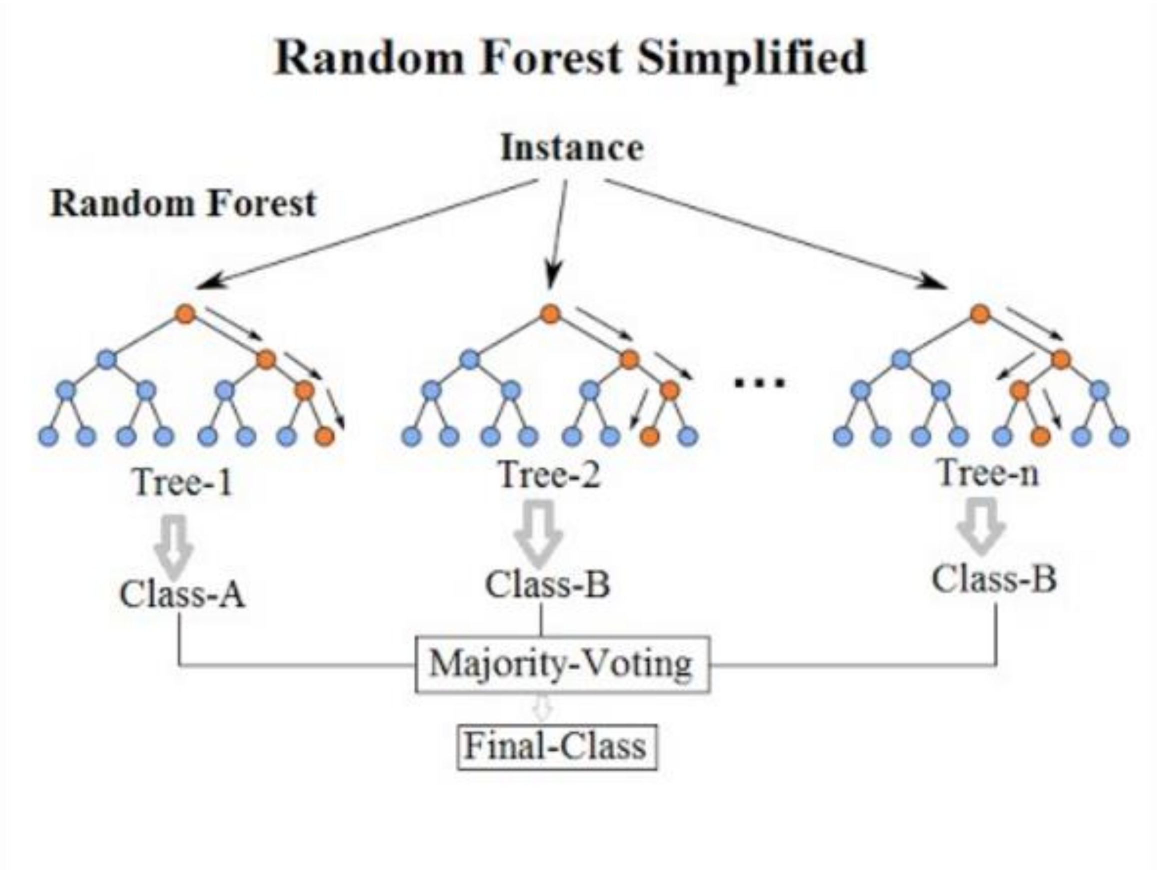
Decision trees, used in operational research, strategic planning and machine learning, are a popular model. Each square above is called a node, and the more nodes you have the (usually) more precise your decision tree will be. The leaves of the tree are called the last nodes of the decision tree, where a decision is made. Decision trees are intuitive and simple to build, but when it comes to accuracy, they fall short.

Advantages: Decision Tree is easy to understand and imagine, needs little planning of information, and can handle numerical and categorical information.

Disadvantages: Decision trees can produce complex trees that do not generalize well and decision trees can be unstable because a completely different tree may be produced due to small variations in the data.

RANDOM FOREST :

Random forests are an ensemble learning methodology focused on decision-making trees. Random forests include using bootstrapped datasets of the original data to create several decision trees and choosing a subset of variables randomly for each stage of the decision tree. The model then selects the mode of each decision tree for all the predictions.

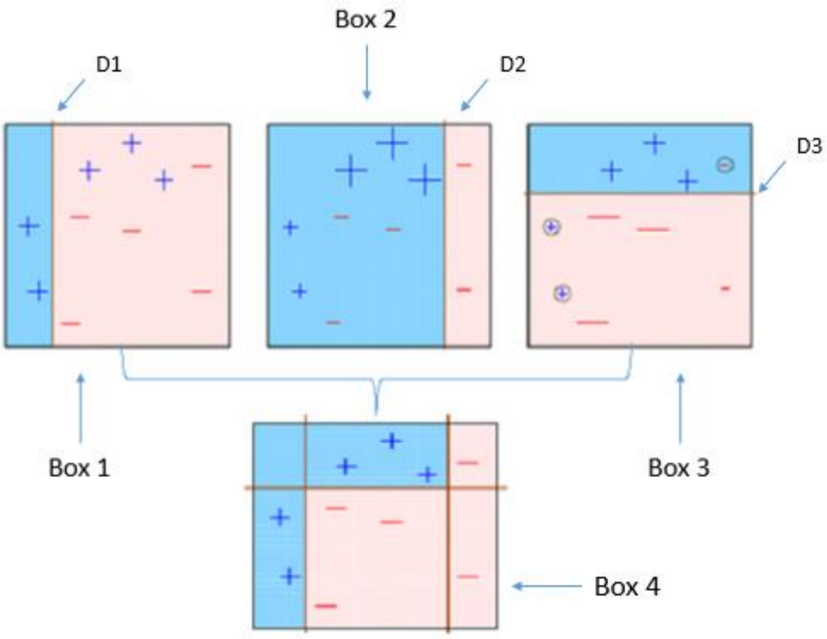


Advantages: In most situations, reductions in over-fitting and random forest classifiers are more specific than decision trees.

Disadvantages: Poor prediction in real time, hard to implement and complicated algorithms.

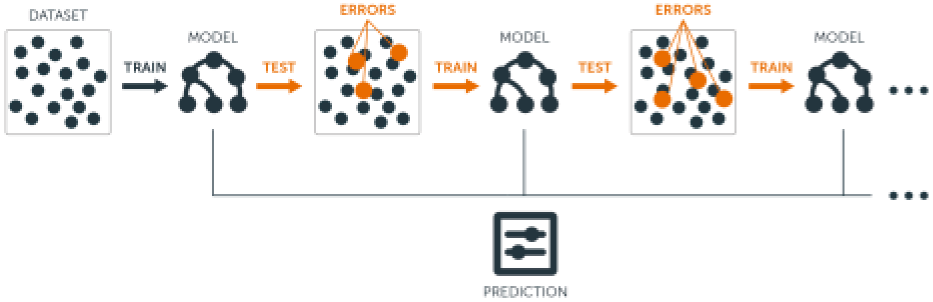
AdaBOOST :

AdaBoost is a method of ensemble learning (also known as "meta-learning") that was initially designed to improve binary classifier performance. In order to benefit from the failures of weak classifiers and transform them into strong ones, AdaBoost uses an iterative technique.



AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique that is used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights to incorrectly classified instances.

GRADIENT BOOSTING :



Gradient boosting is a machine learning technique that generates a prediction model in the form of an ensemble of weak prediction models, usually decision trees, for regression and classification issues. As other boosting techniques do it constructs the model in a phase-wise fashion and generalizes them by allowing an arbitrary differentiable loss function to be optimized.

1.4 WINE QUALITY

During the process of evaluating Wine Quality, there are a few attributes which are to be taken into consideration.

Input variables (based on physicochemical tests) :

1) Fixed acidity :

The total acidity is typically divided into two groups: volatile acids and fixed or nonvolatile acids. The following are among the fixed acids you will find in wines: tartaric, malic, citric, and succinic. In the data sets, this vector is expressed in g(tartaric acid)/dm3.

2) Volatile acidity :

Volatile acidity is essentially the mechanism of transforming wine into vinegar. The volatile acidity is represented in g(aceticacid)/dm3 in this data collection.

3) Citric acid :

One of the fixed acids you'll find in wines is citric acid. In the data set it's represented in g/dm3.

4) Residual sugar :

Usually, residual sugar refers to or is prevented by the sugar left after fermentation ends. In the data set it's represented in g/dm3.

5) Chlorides :

Chlorides may be a major contributor to wine saltiness. In the data, it is written in g(sodium chloride)/dm3.

6) Free sulphur dioxide :

it is said to be bound by the portion of the sulfur dioxide that is added to a wine and that is lost in it while the active part is said to be free. The winemaker is always going to try to acquire the maximum amount of free sulfur to attach. This variable is represented in the data for mg/dm3.

7) Total sulphur dioxide :

The number of the bound and free sulfur dioxide (SO2) is the total sulfur dioxide. It's represented here as mg/dm3. There are legal limits on the amount of sulfur in wines: red wines in the EU can only contain 160mg/L.

8) Density :

As an indicator of the conversion of sugar to alcohol, density is commonly used. It's represented here as g/cm3.

9) pH :

A numerical scale to specify the acidity or fundamentality of the wine is pH or the potential of hydrogen. The pH of most wines is between 2.9 and 3.9 and is therefore acidic.

10) Sulphates :

Just as gluten is for food, sulfates are for wine. They are expressed in g(potassium sulphate)/dm3 in this case.

11) Alcohol :

Wine is an alcoholic drink, and the percentage of alcohol will differ from wine to wine, as you know.It is common that wine is represented in % vol in datasets.

Output variable (based on sensory data):

12) Quality (score between 0 and 10) :

The wine quality was rated between 0 (very poor) and 10 (very good) by wine tasters. The median of at least three assessments made by these same wine experts is the eventual number.

Dataset:

The Dataset used during this Project was obtained from the UCI Machine Learning Repository.

The Dataset goes under the name Wine Data Set and is based on chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

CHAPTER 2

LITERATURE SURVEY

2.1 Selection of Important Features and Predicting Wine Quality using Machine Learning Techniques

Year of Publication - 2017

Dataset Used - Portuguese Wine Dataset

This Paper, presented by Yogesh Gupta, used Linear Regression, Support Vector Machine methods and Neural Network techniques to Predict Wine Quality.

Advantages - Performed over a very large dataset and uses 11 physico-chemical characteristics.

Disadvantages - Works better with fewer characteristics, becomes less accurate when those specific characteristics are not considered.

2.2 Prediction of Quality for Different Type of Wine based on Different Feature Sets Using Supervised Machine Learning Techniques

Year of Publication - 2019

Dataset Used - Portuguese Wine Dataset

This Paper, presented by Satyabrata Aich, Ahmed Absi, Kueh Lee Hui and Mangal Sain, used Genetic Algorithm and Simulated Annealing for Feature Selection and Support Vector Machines to Predict Quality of Wines.

Advantages - Various different models tried, Usage of combinatorial optimization method.

Disadvantages - SA and GA based feature sets don’t provide the highest accuracy.

2.3 Wine Quality Prediction using Machine Learning Algorithms

Year of Publication - 2016

Dataset Used - Red Wine Quality Dataset from Kaggle

This Paper, presented by Devika Pawar, Aakansha Mahajan and Sachin Bhoithe, used Random Forest Decision Trees, Support Vector Machines and Stochastic Gradient Descent techniques to Predict Quality of Wines.

Advantages - Data Splitting was Performed and Data was highly processed.

Disadvantages - Accuracy lower than other implementations.

2.4 Quality Prediction of Red Wine based on Different Feature Sets Using Machine Learning Techniques

Year of Publication - 2020

Dataset Used - Red Wine Dataset extracted from Portuguese Wine Dataset

This Paper, presented by Nikita Sharma, used Logistic Regression, Support Vector Machines, Random Forest and Decision Tree techniques to Predict Quality of Wines.

Advantages - Most number of Models tried and Upto 100% accuracy achieved in certain cases.

Disadvantages - Very high Delta of Accuracy, Model can underperform upto 48%.

2.5 Using Data Mining for Wine Quality Assessment

Year of Publication - 2009

Dataset Used - Portuguese Wine Dataset

This Paper, presented by Paulo Cortez, Telmo Matos and Jose Luis Reis, used Logistic Regression, Support Vector Machine and Multilayer Perceptron Neural Network techniques for Wine Quality Assessment.

Advantages - Utilization of Neural Networks, HIghly apt SVM and HIgh Accuracy.

Disadvantages - Slightly old as compared to other papers, hence misses out on few later advancements in techniques.

2.6 Classification of Cape Gooseberry Fruit According to its Level of Ripeness Using Machine Learning Techniques and Different Color Spaces

Year of Publication - 2019

Dataset Used - A sample of gooseberry fruits was collected from a plantation located in the village of El Faro, Celendin Province, Cajamarca, Peru. The sample consisted of 925 Cape gooseberry fruits with different levels of ripeness.

This Paper, presented by Wilson Castro, Jimy Oblitas, Miguel De-La-Torre, Carlos Cotrina ; Karen Bazán and Himer Avila-George , used Twelve classification models were developed by combining four machine learning techniques, (such as artificial neural networks (ANN), support vector machines (SVMs), decision trees(DT), and K-nearest neighbor algorithms(KNN))with three color spaces (RGB, HSV, and L*a*b*) for classification of Cape Gooseberry according to level of ripeness.

Advantage - The principal component analysis combination of color spaces and machine learning techniques improved the performance of the models.

Disadvantage - Increased complexity of models. Not all combinations have the same accuracy.

2.7 Review on Prediction of Heart Disease Using Data Mining Technique with Wireless Sensor Network

Year of Publication - 2015

Dataset Used - Medical attributes were obtained from the Cleveland heart disease databases. With the help of the dataset the pattern significant to the cardiac prediction was extracted.

This Paper, presented by Prachi Jambhulkar and Vaidehi Baporikar, used Naive bayes (Naïve bayes is a classification technique based on probability theory to find out most likely significant possible classifications), J48 decision tree and bagging techniques for Prediction of Heart Diseases.

Advantage - This system can be used for providing enhanced healthcare services to cardiac patients. Thus, the early diagnosis of heart disease detection may reduce the chances of death in cardiac patients.

Disadvantage - Making the model after physical testing is not feasible.

2.8 Classification of Paddy Rice Using a Stacked Generalization Approach and the Spectral Mixture Method Based on MODIS Time Series

Year of Publication - 2020

Dataset Used - The Dongting Lake area and Poyang Lake area.The Dongting Lake area is located in the middle reach of the Yangtze River, southern China.Poyang Lake is the largest freshwater lake in China, which is also located in the middle reach of the Yangtze River and has similar climatic conditions with Dongting Lake.

This Paper, presented by Meng Zhang, Huaiqing Zhang, Xinyu Li, Yang Liu, Yaotong Cai and Hui Lin, used support vector machine, random forest, k-nearest neighbor (kNN), extreme gradient boosting (XGB), and decision tree techniques for Classification of Paddy Rice.

Advantage - The proposed method achieved high classification accuracies in paddy rice mapping at large scales.

Disadvantage - A number of factors could affect the accuracy of paddy rice mapping when using the proposed method- temporal resolution of the MOD13Q1 dataset ,the residual cloud contamination in the 16-day MODIS time series EVI and the selection of machine learning classifiers in the stacking algorithm.

2.9 Evaluation of Italian wine by the electronic tongue: recognition, quantitative analysis and correlation with human sensory perception

Year of Publication - 2003

Dataset Used - Measurements with the electronic tongue were performed in three types of Italian wine- Castelli Romani denomination, Barbera d'Asti and Gutturnio wine.

This Paper, presented by A. Legin, A. Rudnitskaya, L. Lvova, Yu. Vlasov, C. Di Natale and A. D'Amico, used an electronic tongue based on a sensor array comprising 23 potentiometric cross-sensitive chemical sensors and pattern recognition and multivariate calibration data processing tools that determine taste and flavour of wine and, hence, the system was capable of predicting human sensory scores with average precision.

Advantage - The system can be put forward as an untraditional but yet promising instrument for multi component quantitative analysis of the wine and also for qualitative judgements about the identity of the wines, features of their flavour and hence quality of the wine.

Disadvantage - The main disadvantages of E-Tongue sensors are that they are easily affected by environmental conditions i.e. temperature and humidity, which may cause sensor drift, and the adsorption of solution components that influence the membrane potential.

2.10 A Machine Learning Approach for Lamb Meat Quality Assessment Using FTIR Spectra

Year of Publication - 2020

Dataset Used - Real world spectral dataset of fat samples from suckling lambs.Lambs came from the flocks of three farms. The whole dataset has 134 instances: 66 from lambs being fed with a MR, while the other 68 are reared on ewe milk EM (from up to three days after birth to slaughter).

This Paper, presented by Rocío Alaiz-Rodríguez and Andrew C. Parnell, uses Neural network classifiers as well as different dimensionality reduction techniques. Six feature selection techniques -χ 2 , Information Gain, Gain Ratio, Relief and two embedded techniques based on the decision rule 1R and SVM (Support Vector Machine) are also assessed for Lamb Meat Quality Assessment.

Advantage - The classification model that was developed of fat samples according to the rearing system based on the FTIR spectra provided several advantages to the existing analytical techniques mainly its speed, cost and versatility. It also helped in providing a deeper insight to veterinarian experts about the wavelengths that provide more information for the discrimination of fatty tissues in suckling lamb meat.

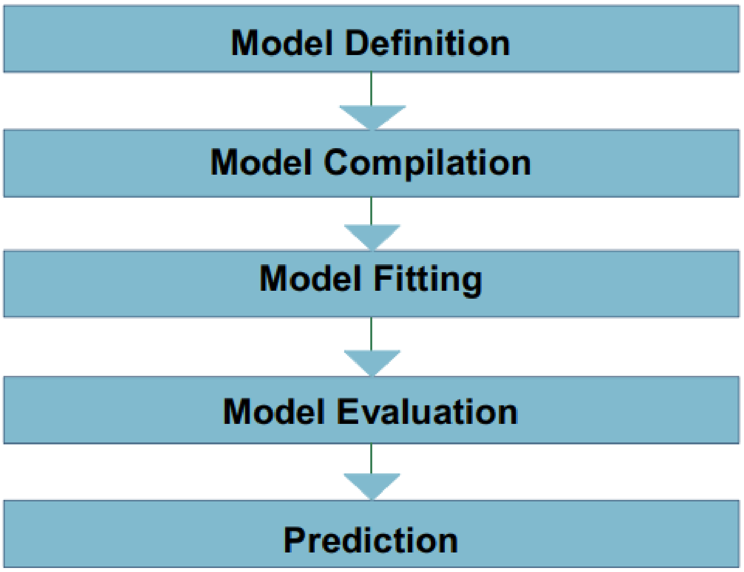
Disadvantage - The feature selectors tend to be unstable for this real world application, likely due to the combination of high dimensionality and relatively few samples and the FTIR spectra comprises a large number of irrelevant and redundant information.

CHAPTER 3

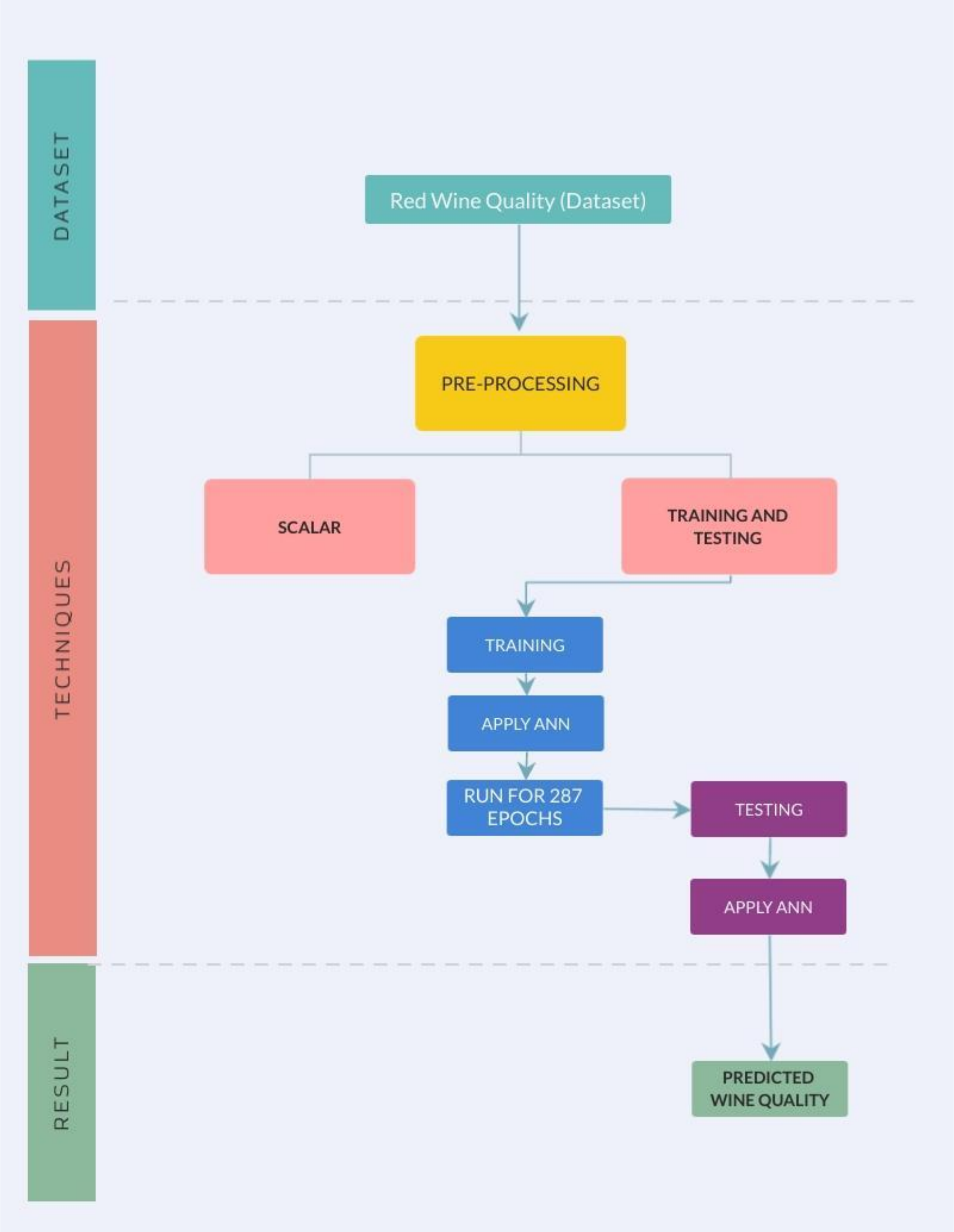
SYSTEM DESIGN

3.1 METHODOLOGY

For predicting the red wine quality implementing a small neural network would work rather than a very large deep neural network. A simple five-step model is developed and the prediction is analyzed based on different epochs and batch sizes.



3.2 SYSTEM ARCHITECTURE



3.3 DESCRIPTION

DATASET :

Machine learning data analysis uses algorithms to continuously improve itself over time, but quality data is necessary for these models to operate efficiently.

Different types of datasets are used to fulfill various roles in the system. The four primary data types used in machine learning is as follows:

1. Numerical Data :

   Numerical data, or quantitative data, is any form of measurable data such as height and weight, etc. This data is just raw numbers.

2. Categorical Data :

   Categorical data is sorted by defining characteristics like gender, social class,etc.

3. Time Series Data :

   Time series data consists of data points that are indexed at specific points in time.

4. Text Data :

   Text data is simply words, sentences, or paragraphs.

Datasets are divided into two cases generally :

1. **Training:**

   These datasets are fed in the machine learning algorithm to ensure that the model is interpreting this data accurately.

2. **Testing:**

   These datasets are used for testing purposes once the algorithm is trained and the validation sets into the system.

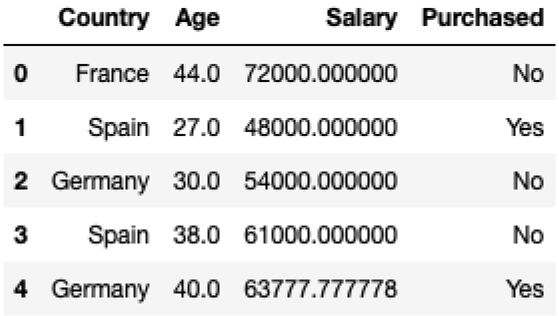
The Dataset used during this Project was obtained from the UCI Machine Learning Repository.

PRE -PROCESSING :

In Machine Learning, data preprocessing is an integral step as the quality of data and the valuable knowledge that can be obtained from it directly affects our model's ability to learn; thus before feeding it into our model, it is extremely necessary to preprocess the data.

STANDARDIZATION:

An integral preprocessing where transformation of values takes place such that the mean values of data is 0 and standard deviation is 1.



	Country	Age	Salary	Purchased
0	France	44.0	72000.000000	No
1	Spain	27.0	48000.000000	Yes
2	Germany	30.0	54000.000000	No
3	Spain	38.0	61000.000000	No
4	Germany	40.0	63777.777778	Yes

For example here we have 2 numerical values: Salary and age and both are to differ that is salary is always higher than age therefore more weightage will be given to salary but age is also considered an integral factor. In order to avoid this we perform standardization and for this a function called StandardScaler is used.

EPOCH :

epoch is a concept used in machine learning and represents the number of passes completed by the machine learning algorithm for the entire training dataset. Datasets (especially when the amount of data is very large) are typically grouped into batches.

TRAINING AND TESTING :

Training Data is a type of labeled data set used to train models of artificial intelligence or algorithms of machine learning to learn from these data sets and improve accuracy while predating the results.

Testing data is very different from training data, since it is a sort of sample of data used to verify model functioning for an impartial assessment of a final model fit on the training dataset.

ARTIFICIAL NEURAL NETWORKS (ANN) :

Artificial Neural networks are typically organized in layers. Layers are being made up of many interconnected ‘nodes’ which contain an ‘activation function’. A neural network may contain the following 3 layers:

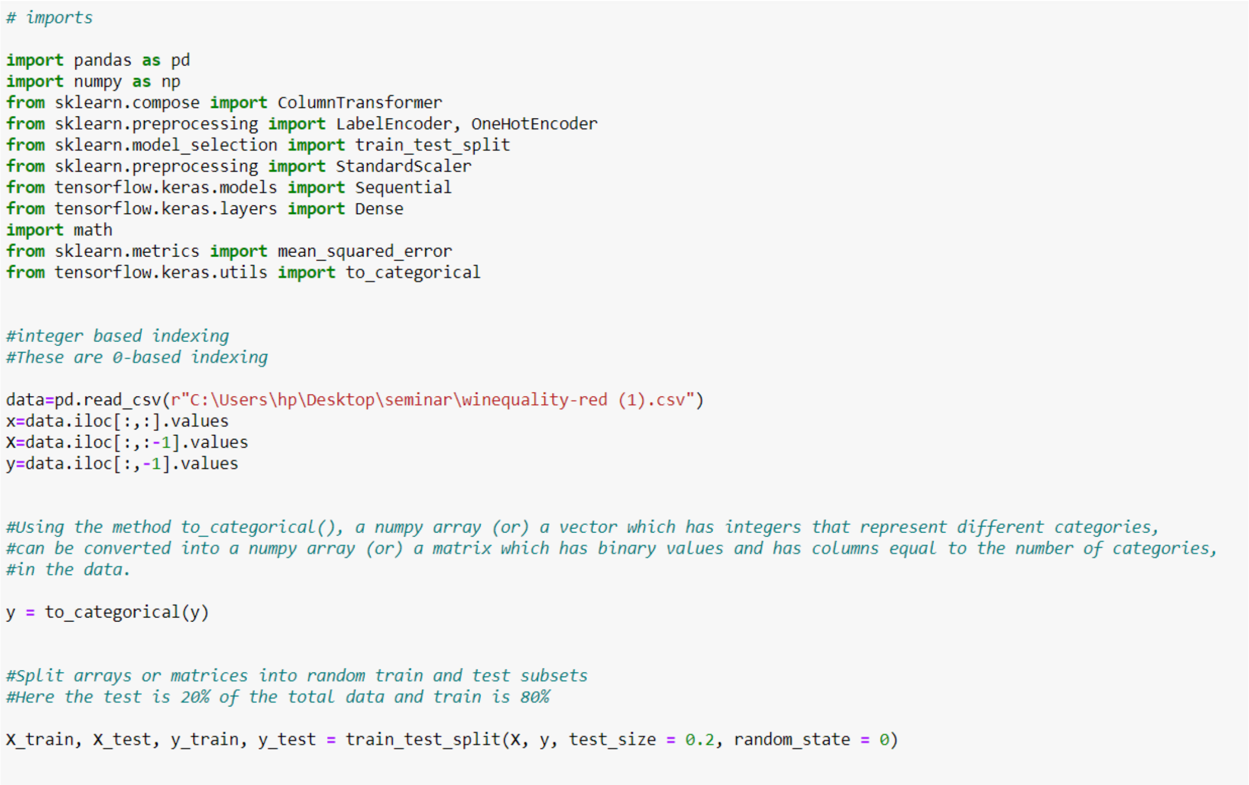
1. Input Layer
2. Hidden Layer
3. Output Layer

CHAPTER 4

CODING

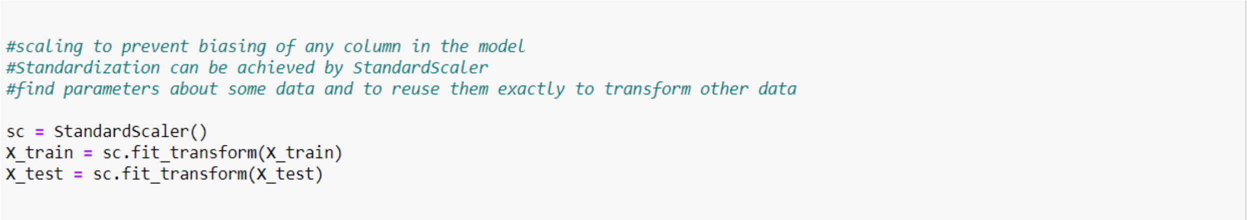
NEURAL NETWORKS :

PRE -PROCESSING :



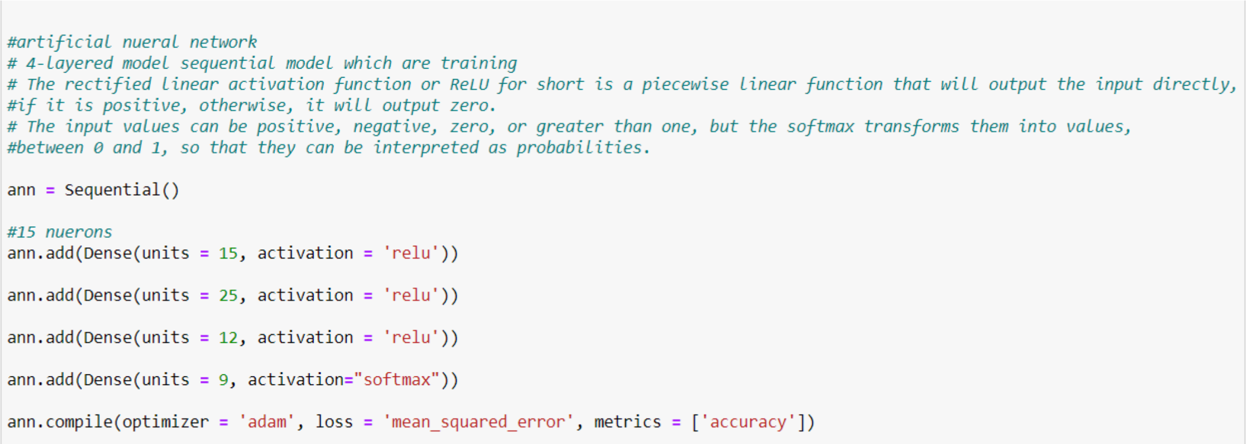
```
# imports

import pandas as pd
import numpy as np
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import math
from sklearn.metrics import mean_squared_error
from tensorflow.keras.utils import to_categorical


#integer based indexing
#These are 0-based indexing

data=pd.read_csv(r"C:\Users\hp\Desktop\seminar\winequality-red (1).csv")
x=data.iloc[:,:].values
X=data.iloc[:,:-1].values
y=data.iloc[:,-1].values


#Using the method to_categorical(), a numpy array (or) a vector which has integers that represent different categories,
#can be converted into a numpy array (or) a matrix which has binary values and has columns equal to the number of categories,
#in the data.

y = to_categorical(y)


#Split arrays or matrices into random train and test subsets
#Here the test is 20% of the total data and train is 80%

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

STANDARDIZATION :



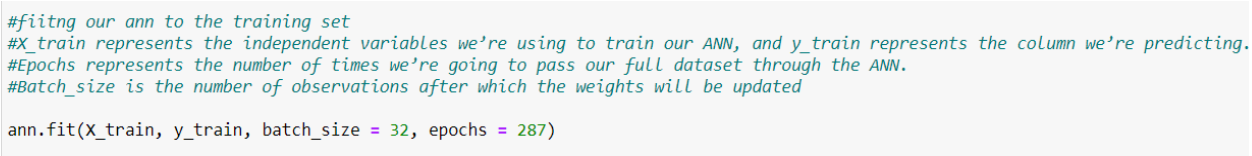
```
#scaling to prevent biasing of any column in the model
#Standardization can be achieved by StandardScaler
#find parameters about some data and to reuse them exactly to transform other data

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

MODEL :



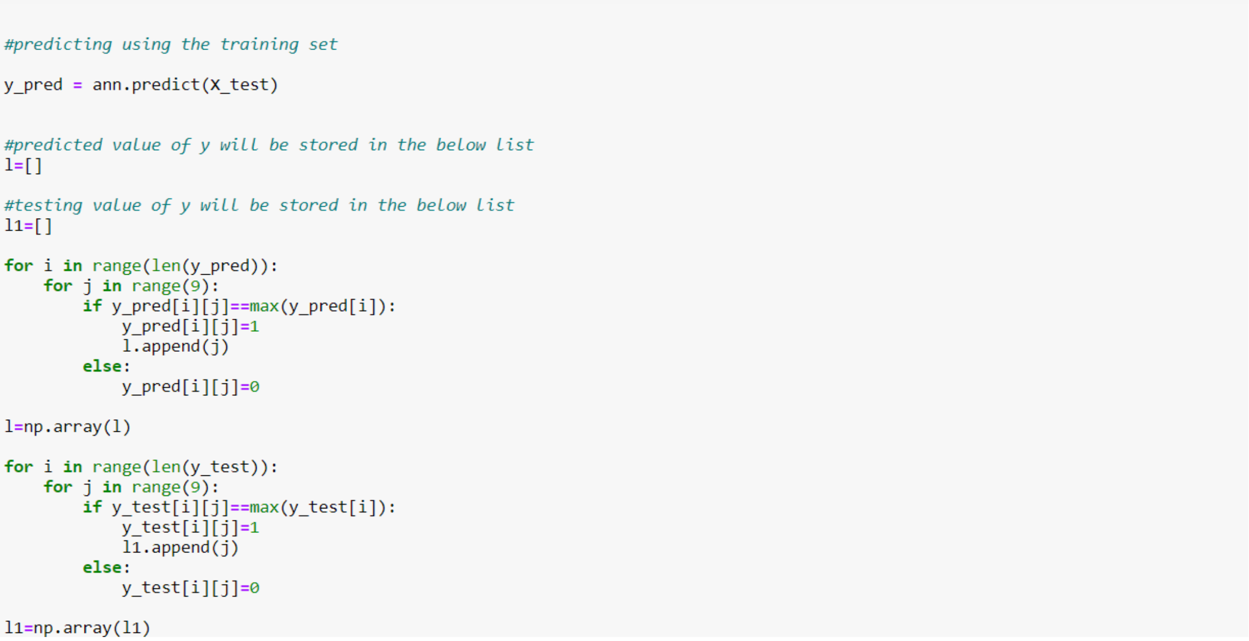
```
#artificial nueral network
# 4-layered model sequential model which are training
# The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly,
#if it is positive, otherwise, it will output zero.
# The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values,
#between 0 and 1, so that they can be interpreted as probabilities.

ann = Sequential()

#15 nuerons
ann.add(Dense(units = 15, activation = 'relu'))

ann.add(Dense(units = 25, activation = 'relu'))

ann.add(Dense(units = 12, activation = 'relu'))

ann.add(Dense(units = 9, activation="softmax"))

ann.compile(optimizer = 'adam', loss = 'mean_squared_error', metrics = ['accuracy'])
```

TRAINING :



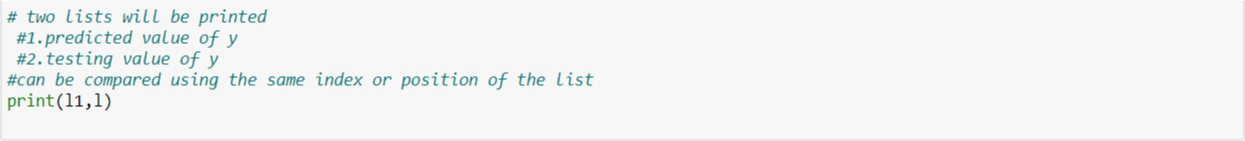
```
#fiitng our ann to the training set
#X_train represents the independent variables we're using to train our ANN, and y_train represents the column we're predicting.
#Epochs represents the number of times we're going to pass our full dataset through the ANN.
#Batch_size is the number of observations after which the weights will be updated

ann.fit(X_train, y_train, batch_size = 32, epochs = 287)
```

TESTING :

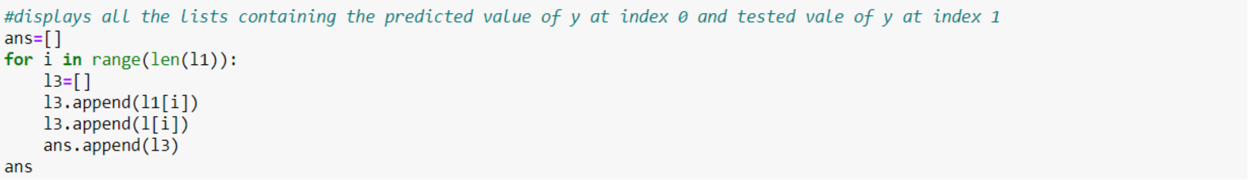


```
#predicting using the training set

y_pred = ann.predict(X_test)


#predicted value of y will be stored in the below list
l=[]

#testing value of y will be stored in the below list
l1=[]

for i in range(len(y_pred)):
    for j in range(9):
        if y_pred[i][j]==max(y_pred[i]):
            y_pred[i][j]=1
            l.append(j)
        else:
            y_pred[i][j]=0

l=np.array(l)

for i in range(len(y_test)):
    for j in range(9):
        if y_test[i][j]==max(y_test[i]):
            y_test[i][j]=1
            l1.append(j)
        else:
            y_test[i][j]=0

l1=np.array(l1)
```

PRINTING TEST AND RESULT VALUES IN LIST :

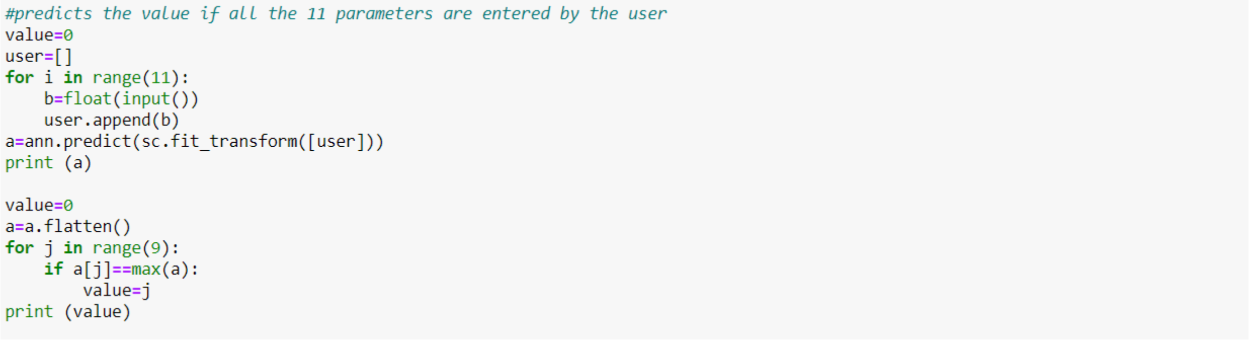


```
# two lists will be printed
 #1.predicted value of y
 #2.testing value of y
#can be compared using the same index or position of the list
print(l1,l)
```



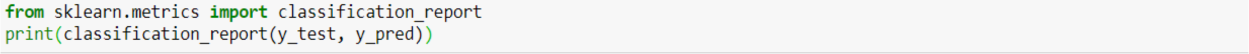
```
#displays all the lists containing the predicted value of y at index 0 and tested vale of y at index 1
ans=[]
for i in range(len(l1)):
    l3=[]
    l3.append(l1[i])
    l3.append(l[i])
    ans.append(l3)
ans
```

TAKING USER INPUT AND PRINTING THE PREDICTED VALUE :



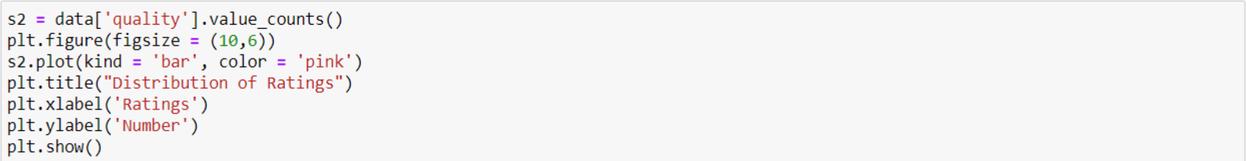
```
#predicts the value if all the 11 parameters are entered by the user
value=0
user=[]
for i in range(11):
    b=float(input())
    user.append(b)
a=ann.predict(sc.fit_transform([user]))
print (a)

value=0
a=a.flatten()
for j in range(9):
    if a[j]==max(a):
        value=j
print (value)
```

CLASSIFICATION REPORT :



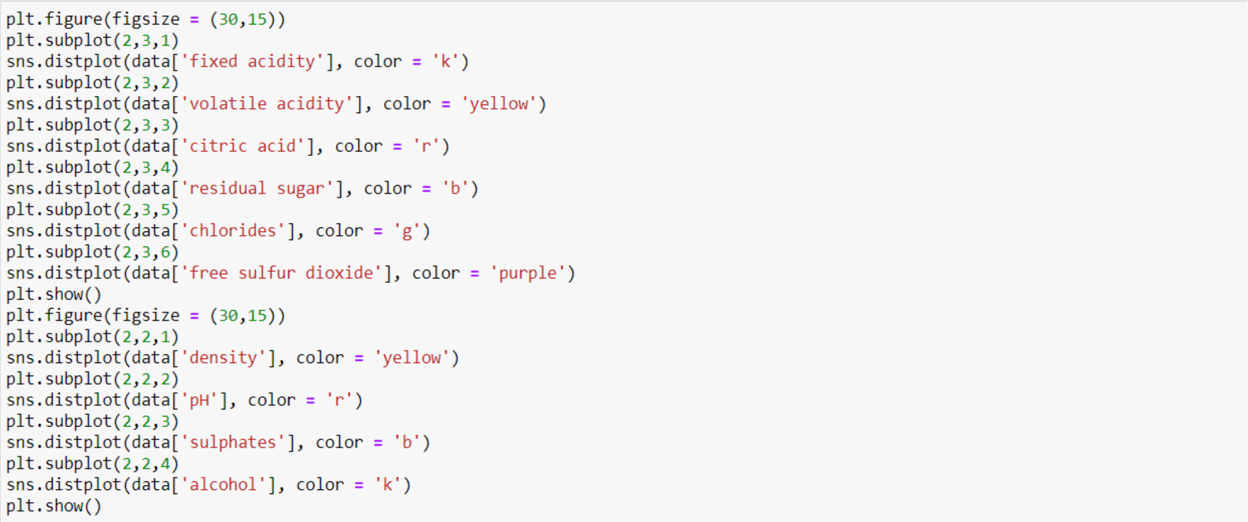
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

VALUE COUNT :



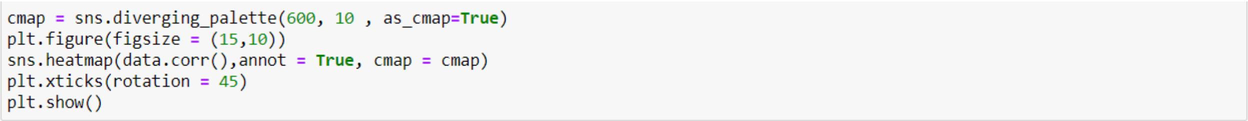
```
s2 = data['quality'].value_counts()
plt.figure(figsize = (10,6))
s2.plot(kind = 'bar', color = 'pink')
plt.title("Distribution of Ratings")
plt.xlabel('Ratings')
plt.ylabel('Number')
plt.show()
```

VISUALISATION :



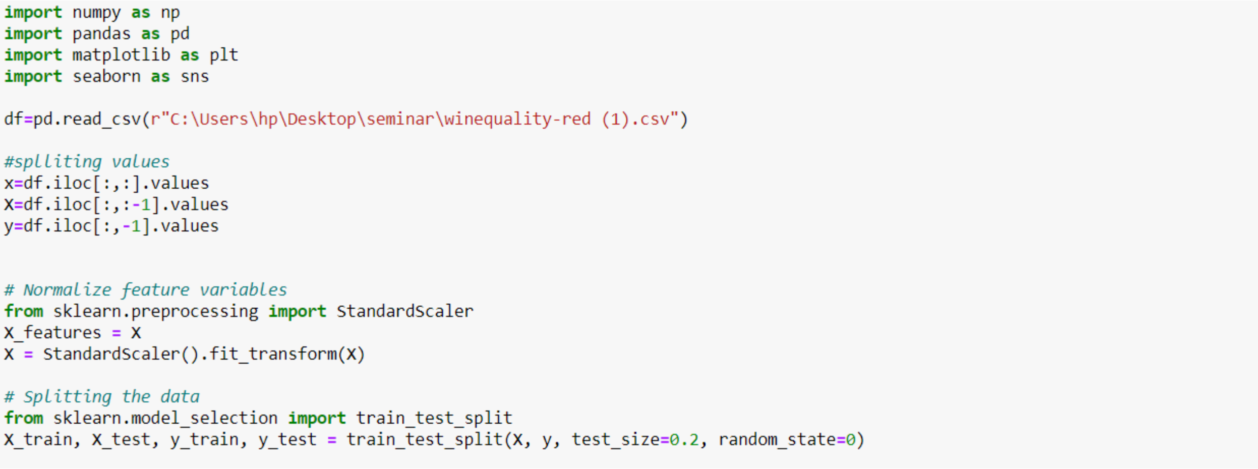
```
plt.figure(figsize = (30,15))
plt.subplot(2,3,1)
sns.distplot(data['fixed acidity'], color = 'k')
plt.subplot(2,3,2)
sns.distplot(data['volatile acidity'], color = 'yellow')
plt.subplot(2,3,3)
sns.distplot(data['citric acid'], color = 'r')
plt.subplot(2,3,4)
sns.distplot(data['residual sugar'], color = 'b')
plt.subplot(2,3,5)
sns.distplot(data['chlorides'], color = 'g')
plt.subplot(2,3,6)
sns.distplot(data['free sulfur dioxide'], color = 'purple')
plt.show()
plt.figure(figsize = (30,15))
plt.subplot(2,2,1)
sns.distplot(data['density'], color = 'yellow')
plt.subplot(2,2,2)
sns.distplot(data['pH'], color = 'r')
plt.subplot(2,2,3)
sns.distplot(data['sulphates'], color = 'b')
plt.subplot(2,2,4)
sns.distplot(data['alcohol'], color = 'k')
plt.show()
```

CORRELATION :



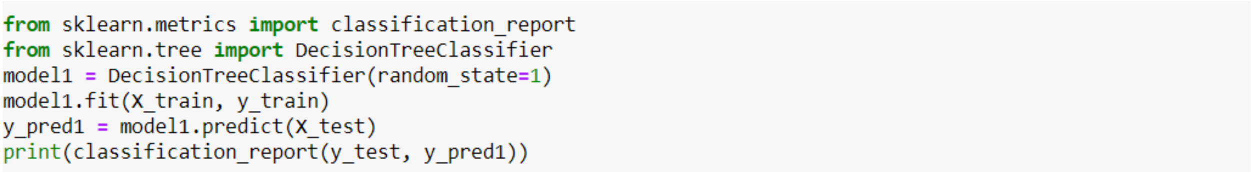
```
cmap = sns.diverging_palette(600, 10 , as_cmap=True)
plt.figure(figsize = (15,10))
sns.heatmap(data.corr(),annot = True, cmap = cmap)
plt.xticks(rotation = 45)
plt.show()
```

COMMON CODE (FOR DECISION TREE, AdaBOOST, GRADIENT BOOST) :



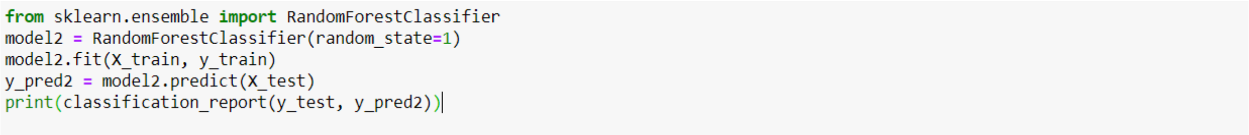
```
import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns

df=pd.read_csv(r"C:\Users\hp\Desktop\seminar\winequality-red (1).csv")

#splliting values
x=df.iloc[:,:].values
X=df.iloc[:,:-1].values
y=df.iloc[:,-1].values


# Normalize feature variables
from sklearn.preprocessing import StandardScaler
X_features = X
X = StandardScaler().fit_transform(X)

# Splitting the data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

DECISION TREE :



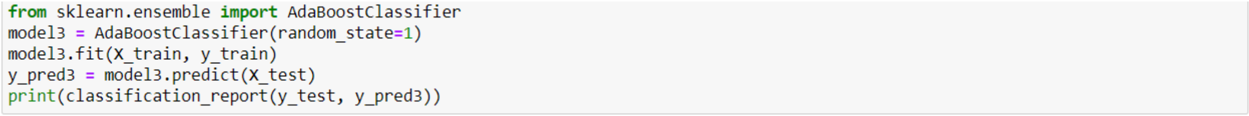
```
from sklearn.metrics import classification_report
from sklearn.tree import DecisionTreeClassifier
model1 = DecisionTreeClassifier(random_state=1)
model1.fit(X_train, y_train)
y_pred1 = model1.predict(X_test)
print(classification_report(y_test, y_pred1))
```

RANDOM FOREST :



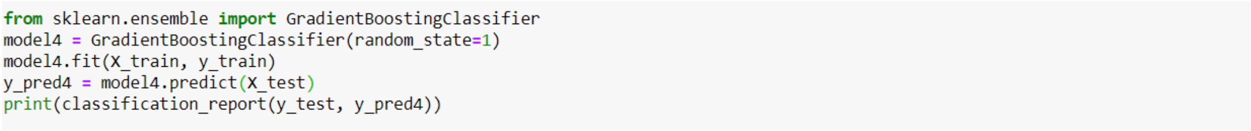
```
from sklearn.ensemble import RandomForestClassifier
model2 = RandomForestClassifier(random_state=1)
model2.fit(X_train, y_train)
y_pred2 = model2.predict(X_test)
print(classification_report(y_test, y_pred2))
```

AdaBOOST :



```
from sklearn.ensemble import AdaBoostClassifier
model3 = AdaBoostClassifier(random_state=1)
model3.fit(X_train, y_train)
y_pred3 = model3.predict(X_test)
print(classification_report(y_test, y_pred3))
```

GRADIENT BOOSTING :



```
from sklearn.ensemble import GradientBoostingClassifier
model4 = GradientBoostingClassifier(random_state=1)
model4.fit(X_train, y_train)
y_pred4 = model4.predict(X_test)
print(classification_report(y_test, y_pred4))
```

CHAPTER 5

RESULT

MODEL SELECTION :

In Machine Learning, there is no free lunch. Therefore, choosing the algorithm to use depends on several variables, from the type of issue at hand to the type of output that you are looking for.

The aim of model selection is to find the best generalisation properties of the network architecture, that is, that which minimises the error on the data set instances selected.

Five approaches were taken to solve the red-wine quality problem :

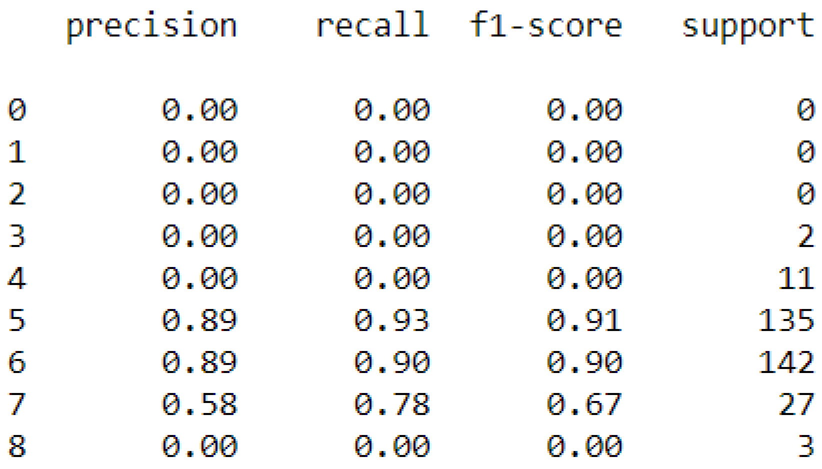
1. Neural Networks
2. Decision Tree
3. Random Forest
4. AdaBoost
5. Gradient Boost

Some classification techniques used are as follows:

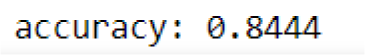
1. **Precision** is a measure of classifier's exactness. The higher the precision the more accurate the classifier.
2. **Recall** is a measure of a classifier's completeness. The higher the recall, the more cases the classifier covers.
3. The **F1 Score or F-score** is a weighted average of precision and recall.

The Classification Report and the Accuracy of the approaches are as follows :

NEURAL NETWORKS :

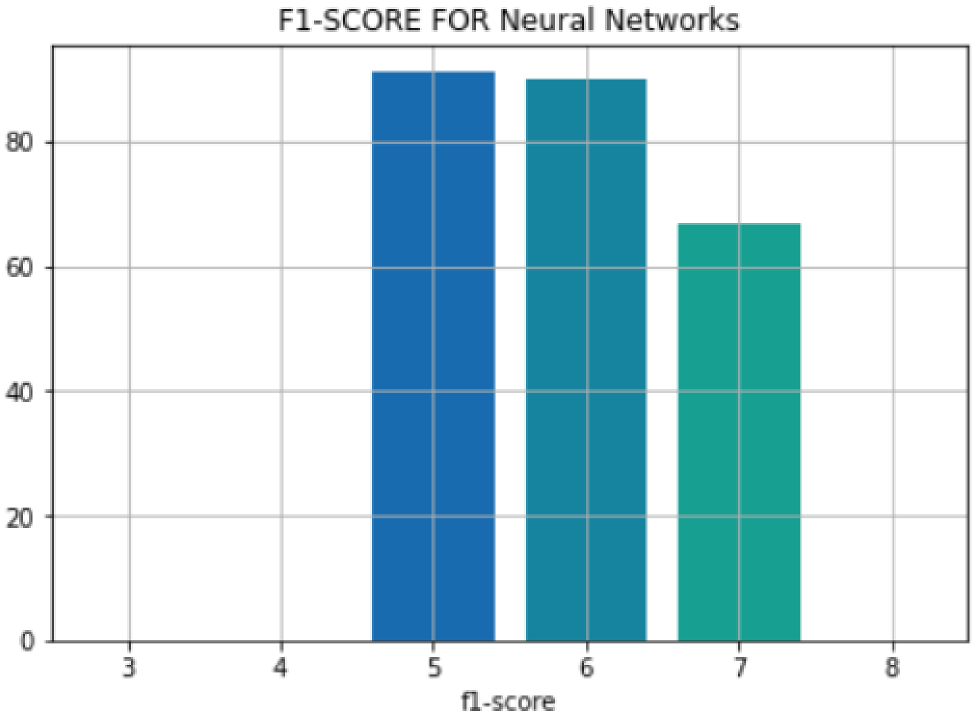


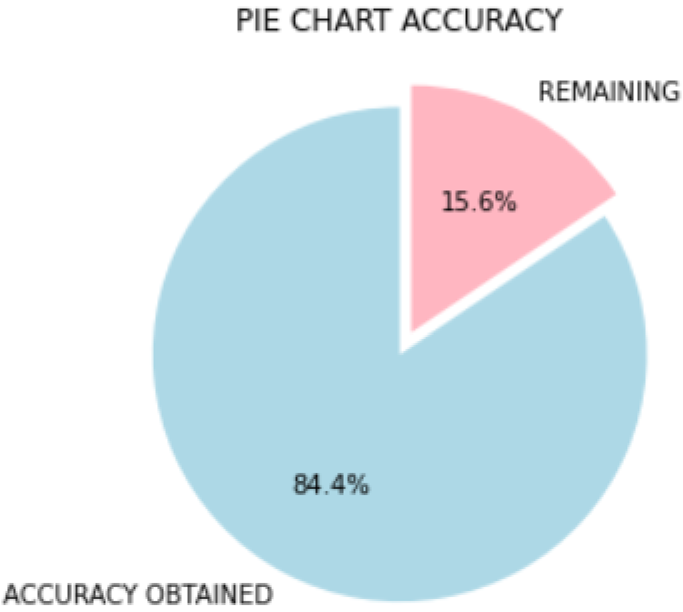
```
         precision    recall  f1-score   support

0           0.00      0.00      0.00         0
1           0.00      0.00      0.00         0
2           0.00      0.00      0.00         0
3           0.00      0.00      0.00         2
4           0.00      0.00      0.00        11
5           0.89      0.93      0.91       135
6           0.89      0.90      0.90       142
7           0.58      0.78      0.67        27
8           0.00      0.00      0.00         3
```



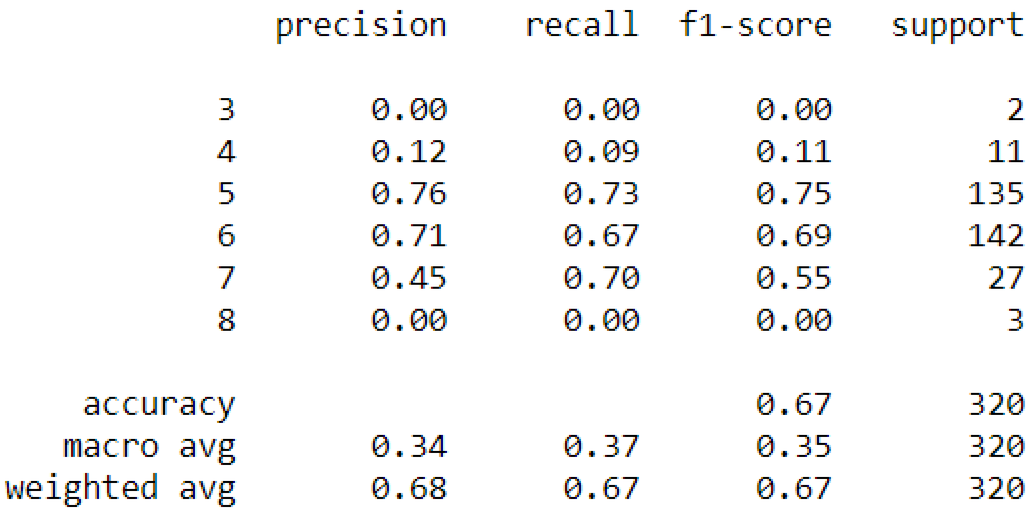
```
accuracy: 0.8444
```

VISUALIZATION :



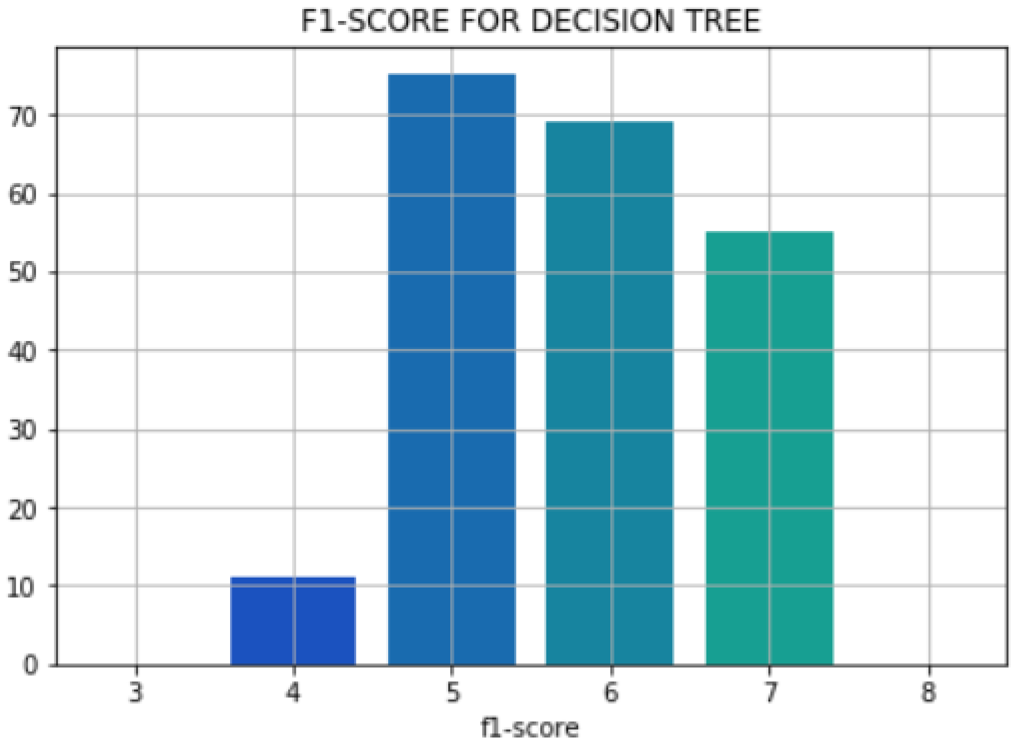


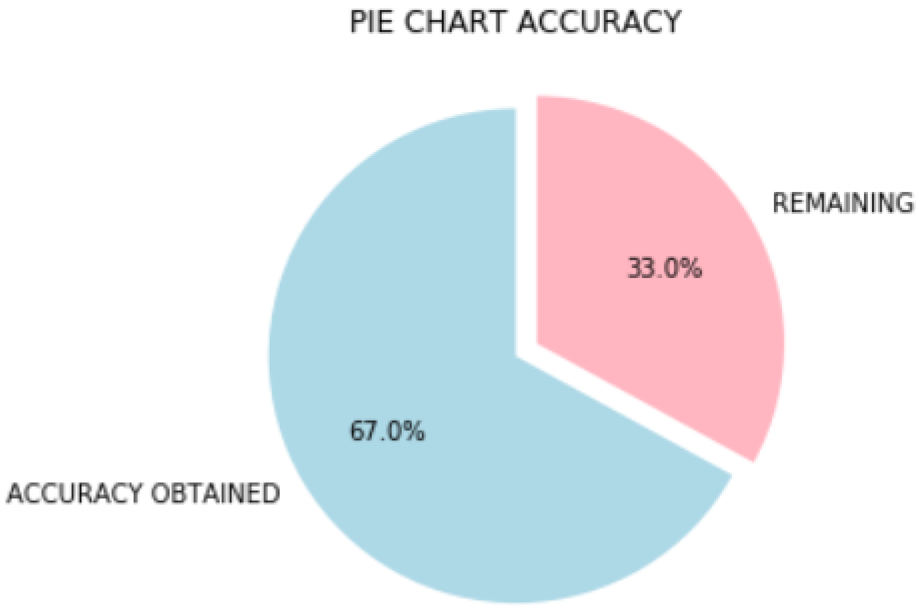
DECISION TREE :



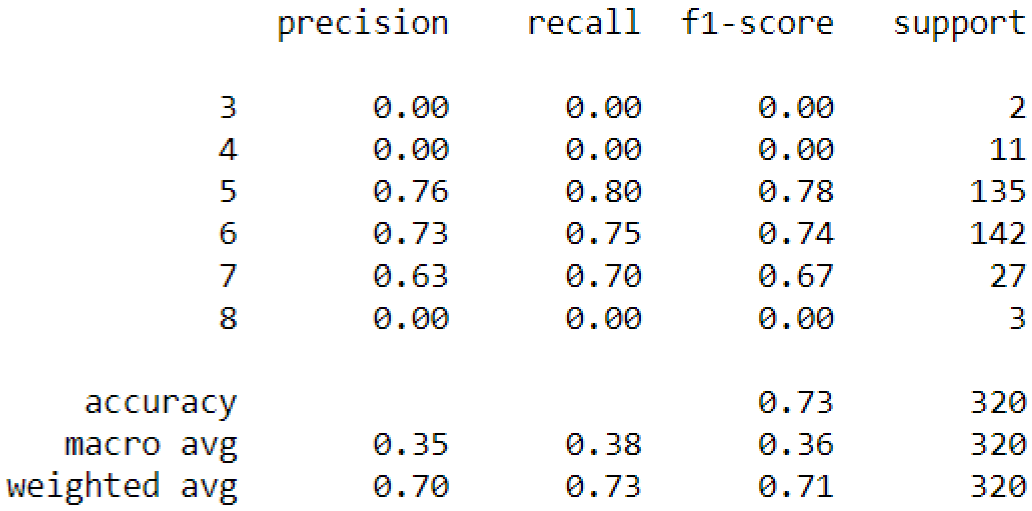
```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.12      0.09      0.11        11
           5       0.76      0.73      0.75       135
           6       0.71      0.67      0.69       142
           7       0.45      0.70      0.55        27
           8       0.00      0.00      0.00         3

    accuracy                           0.67       320
   macro avg       0.34      0.37      0.35       320
weighted avg       0.68      0.67      0.67       320
```

VISUALIZATION :



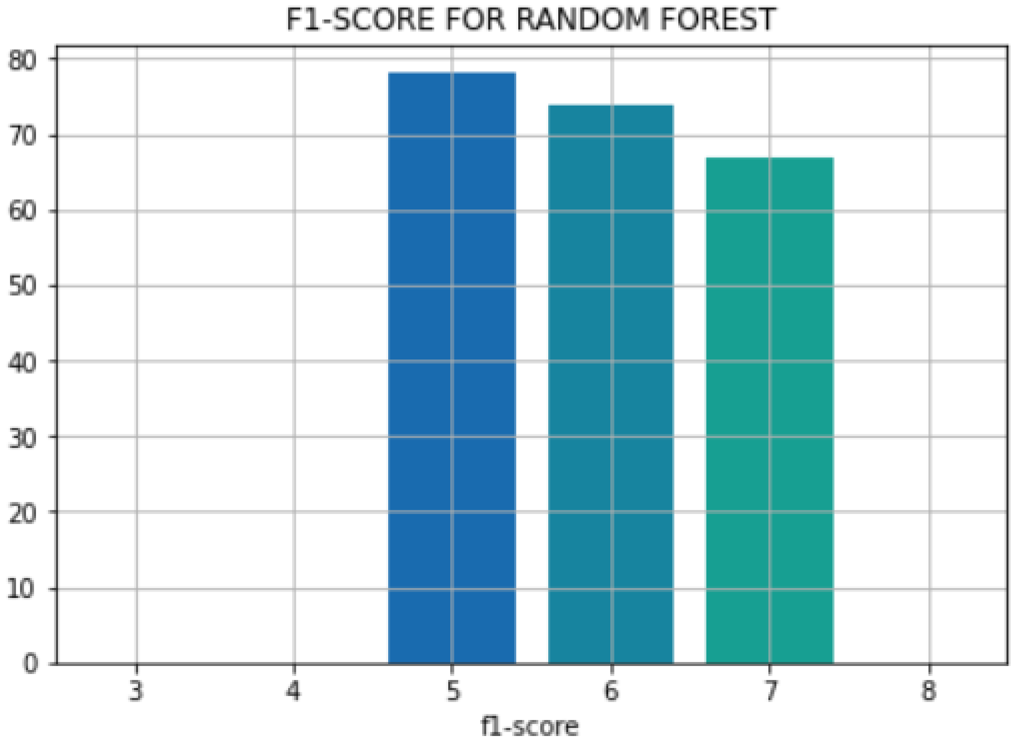


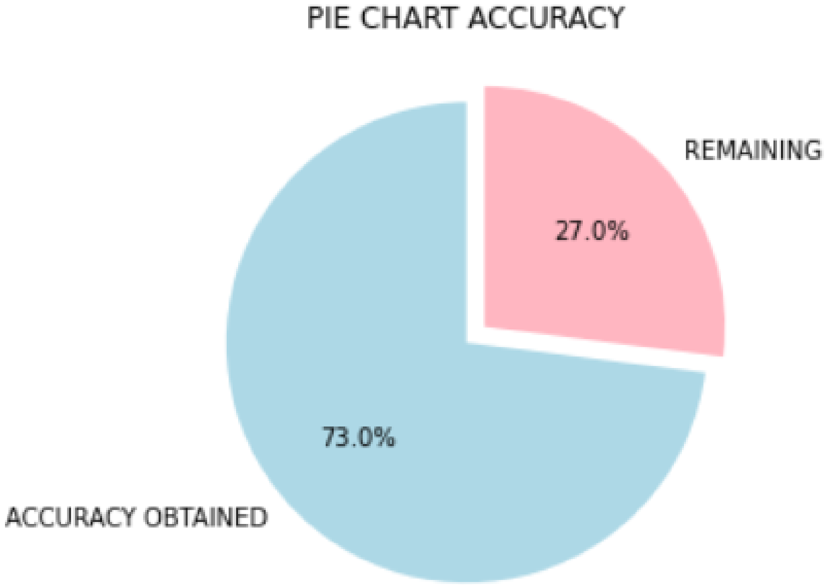
RANDOM FOREST :



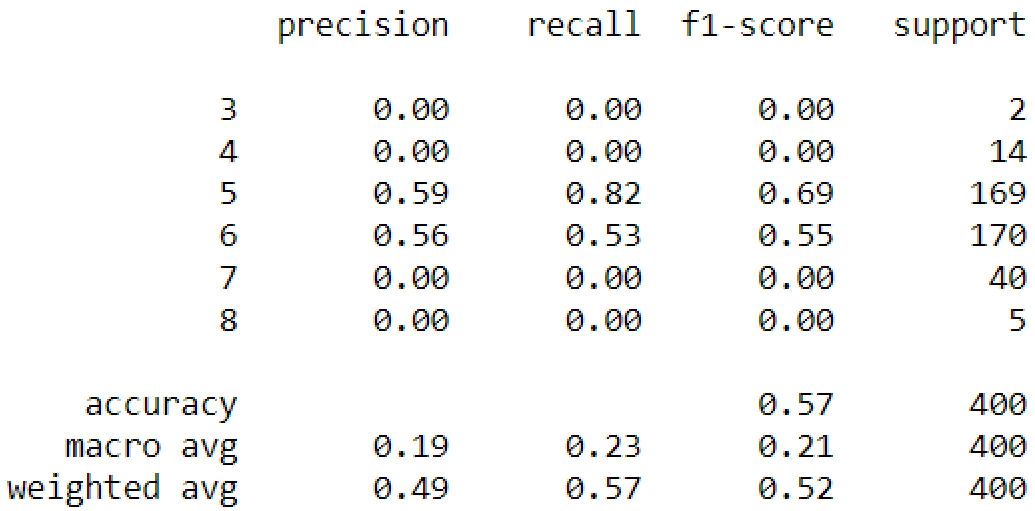
```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        11
           5       0.76      0.80      0.78       135
           6       0.73      0.75      0.74       142
           7       0.63      0.70      0.67        27
           8       0.00      0.00      0.00         3

    accuracy                           0.73       320
   macro avg       0.35      0.38      0.36       320
weighted avg       0.70      0.73      0.71       320
```

VISUALIZATION :



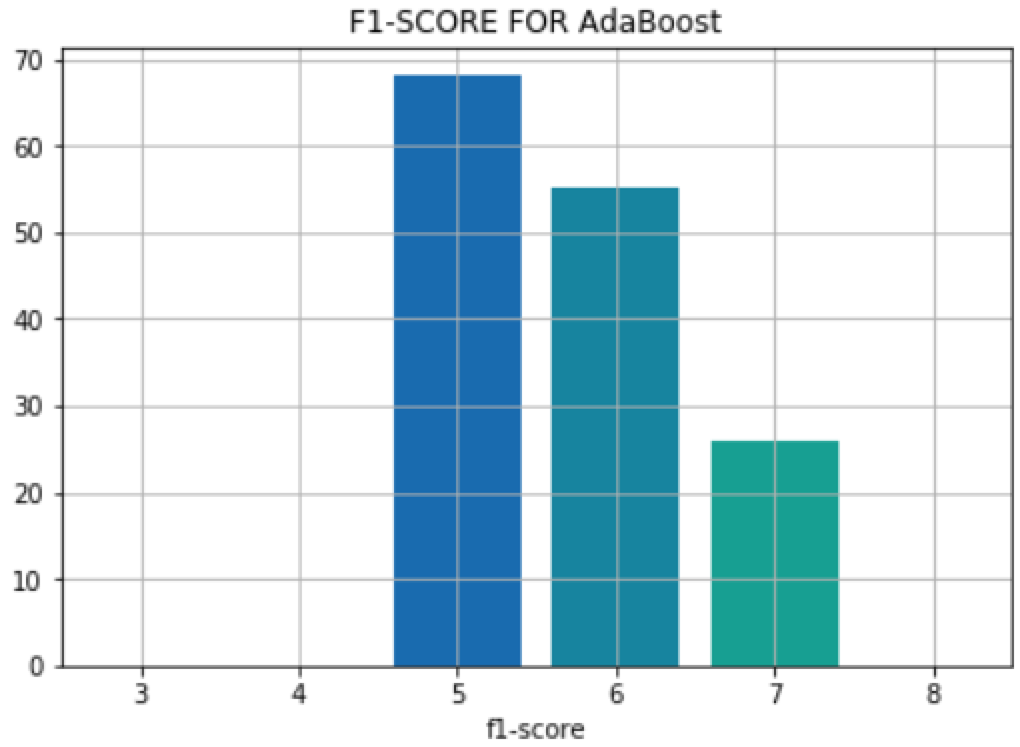


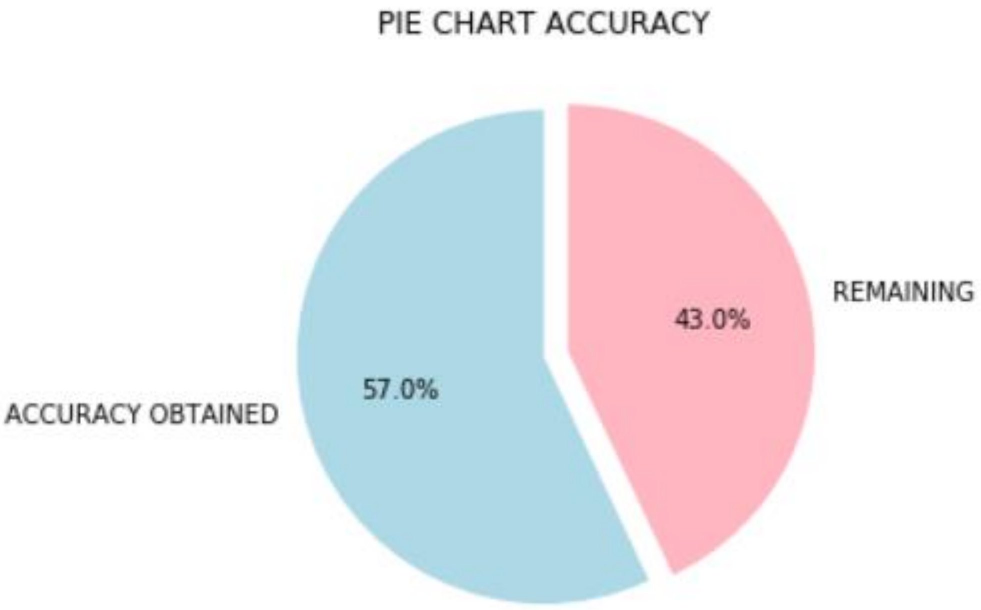
AdaBOOST :



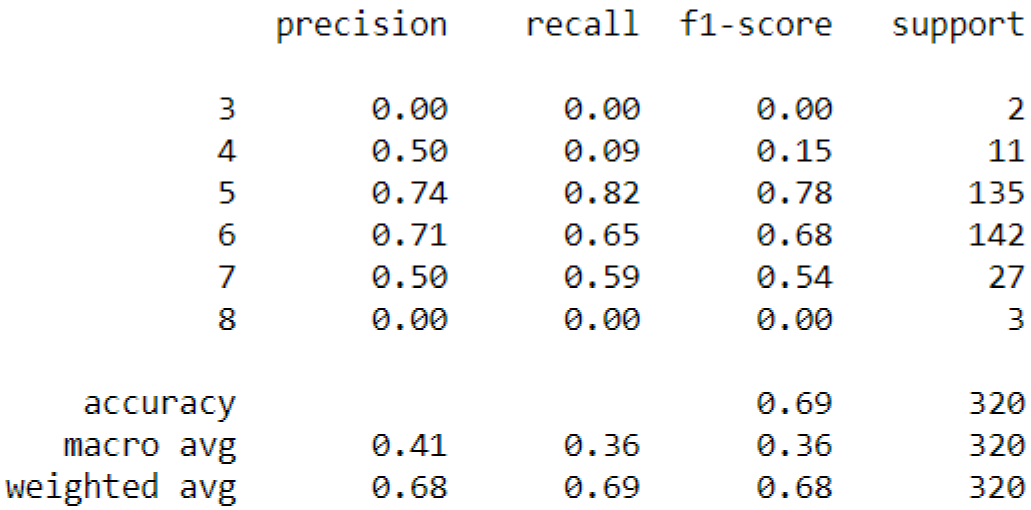
```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        14
           5       0.59      0.82      0.69       169
           6       0.56      0.53      0.55       170
           7       0.00      0.00      0.00        40
           8       0.00      0.00      0.00         5

    accuracy                           0.57       400
   macro avg       0.19      0.23      0.21       400
weighted avg       0.49      0.57      0.52       400
```

VISUALIZATION :



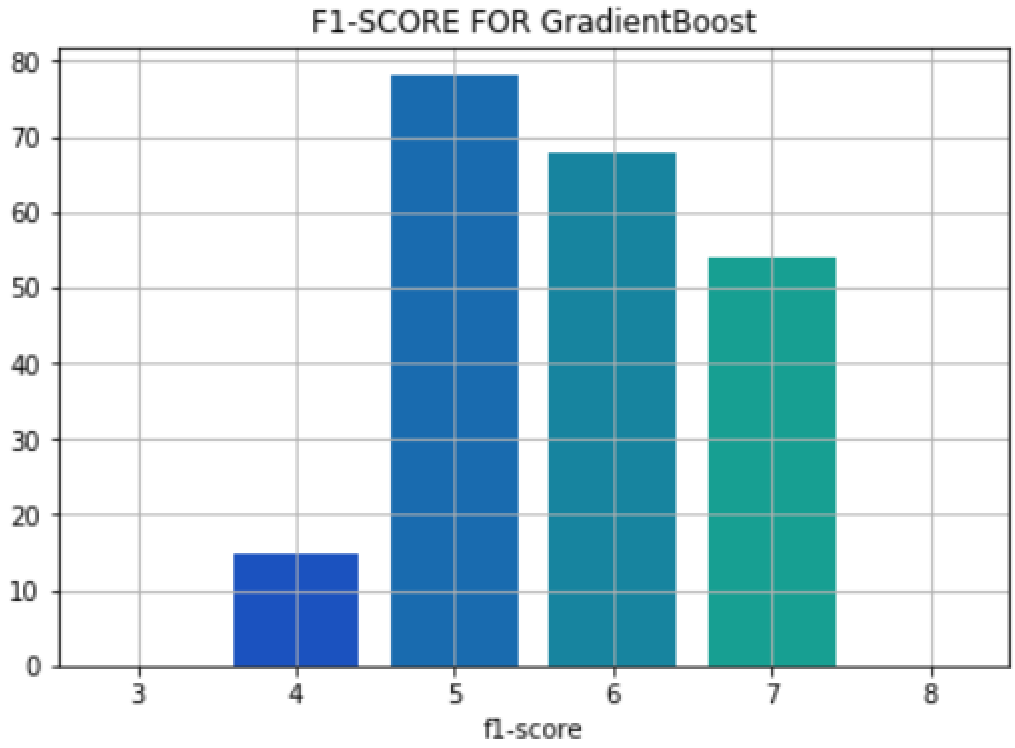


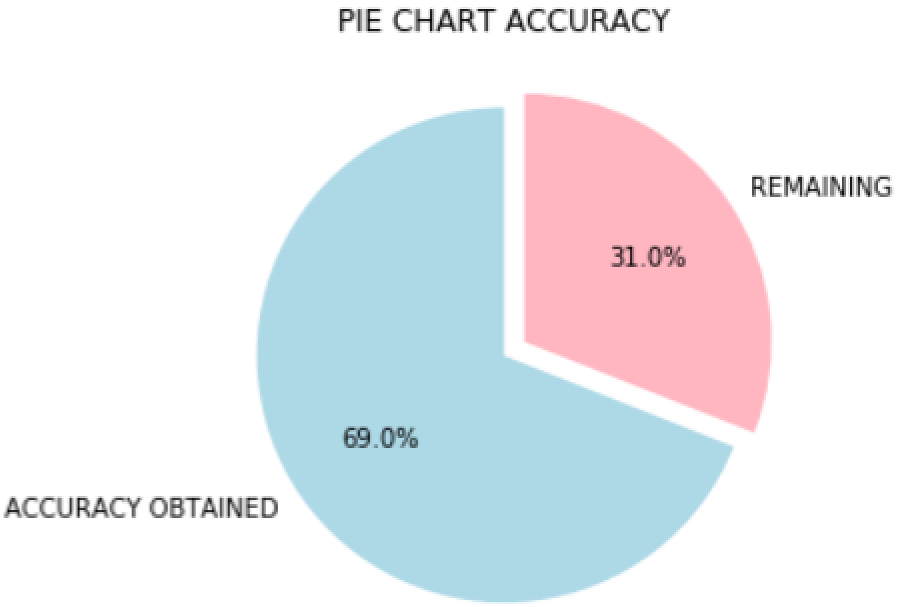
GRADIENT BOOST :



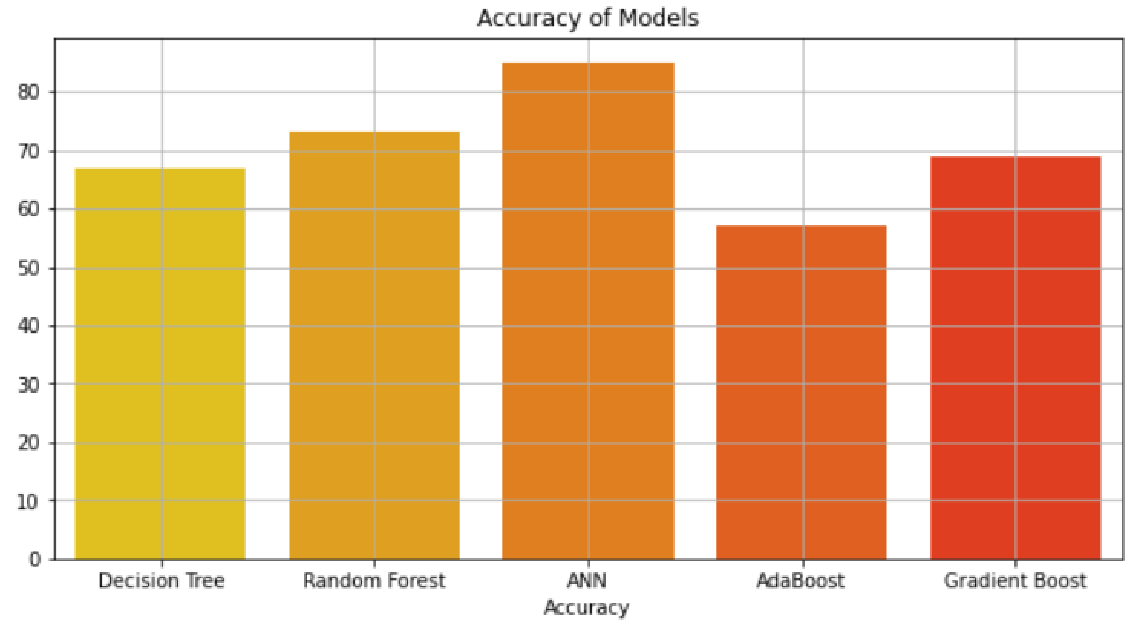
```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.50      0.09      0.15        11
           5       0.74      0.82      0.78       135
           6       0.71      0.65      0.68       142
           7       0.50      0.59      0.54        27
           8       0.00      0.00      0.00         3

    accuracy                           0.69       320
   macro avg       0.41      0.36      0.36       320
weighted avg       0.68      0.69      0.68       320
```

VISUALIZATION :





COMPARING ACCURACY OF ALL FIVE APPROACHES:



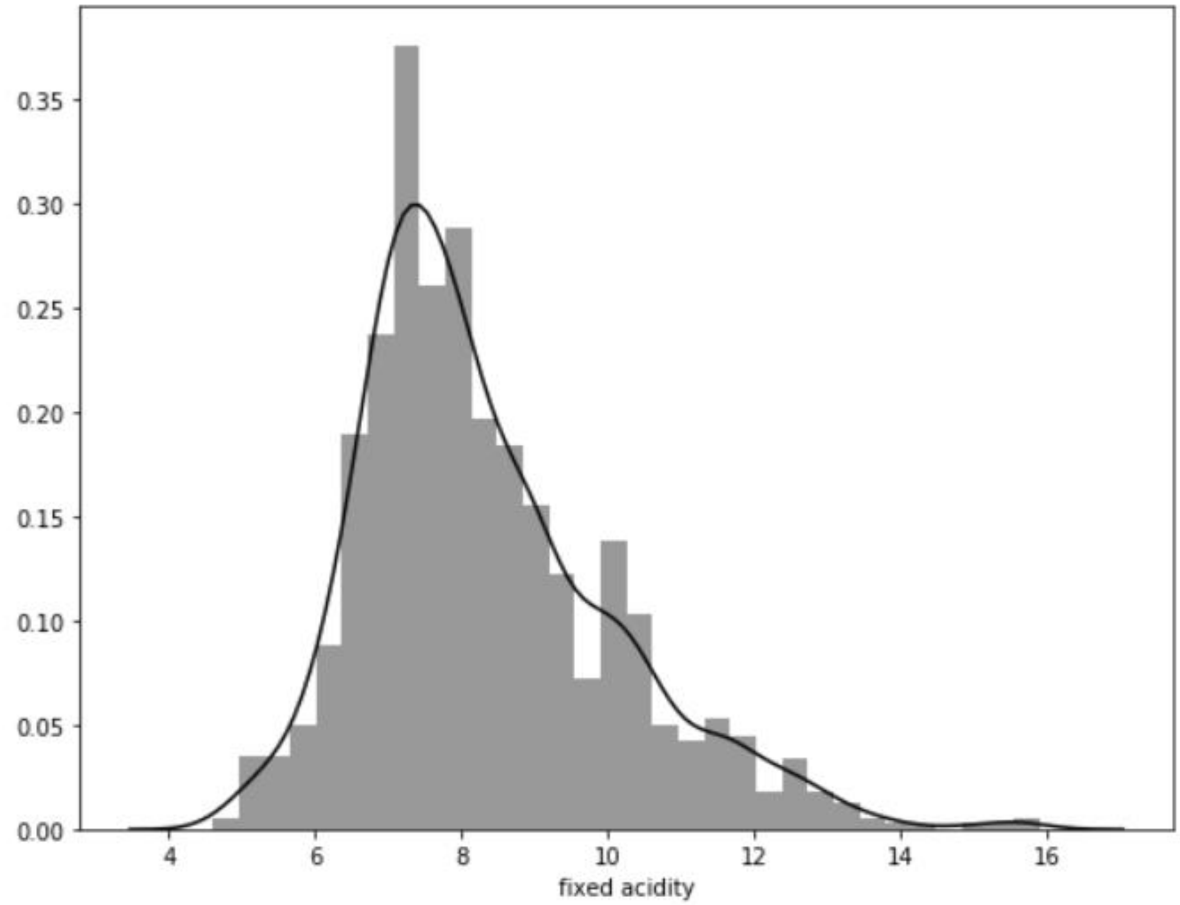
By comparing all the different approaches we see that the Ann model or the Neural Network model has the highest accuracy and it will be the most suitable model to develop a Prediction Model to determine the Quality of Red Wines distributed.

Data exploration helped to double check the whether the data contained all the variables that were present in the description file of UCI Machine Learning Repository. Apart from that, it also checked whether there were any null values that should be taken into account.

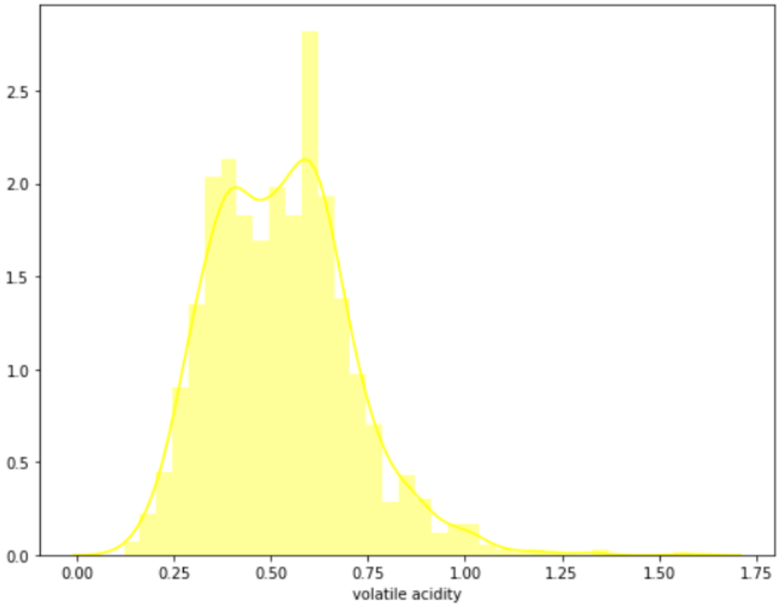
VISUALISING THE DATA :

One way to do this is to look at the distribution of certain variables in the dataset and build scatter plots to see correlations.

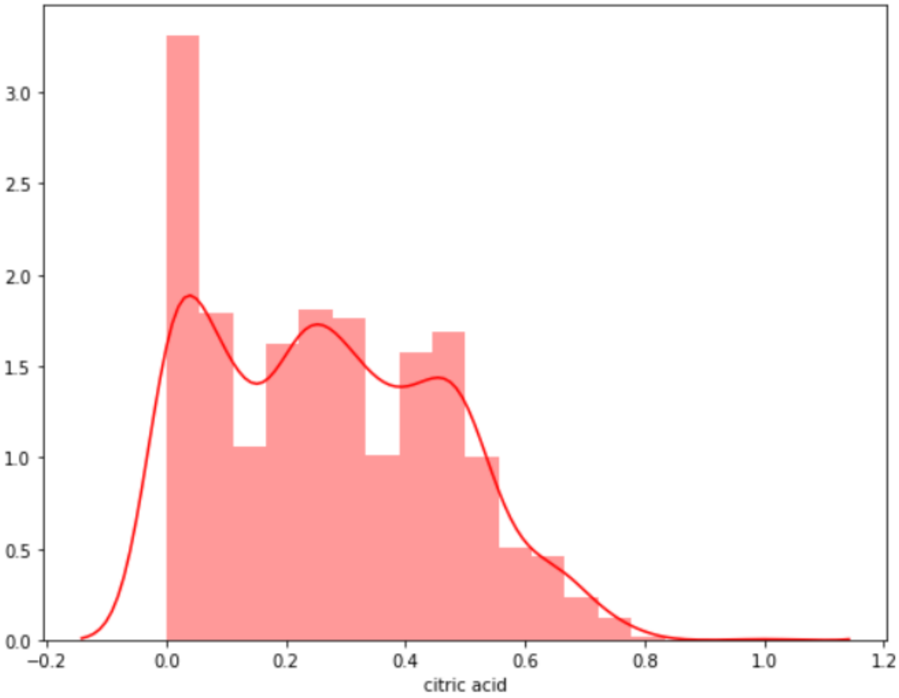
<u>FIXED ACIDITY :</u>



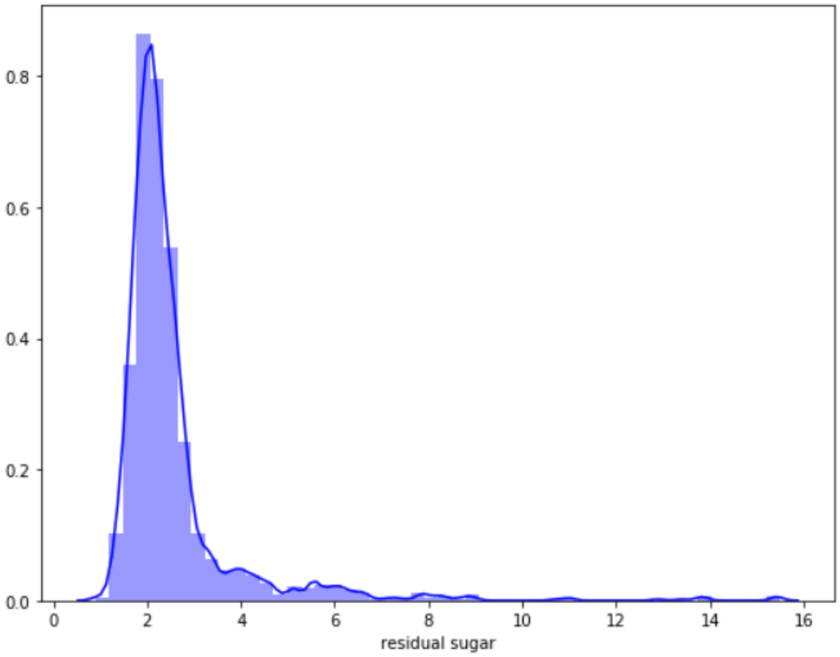
VOLATILE ACIDITY :



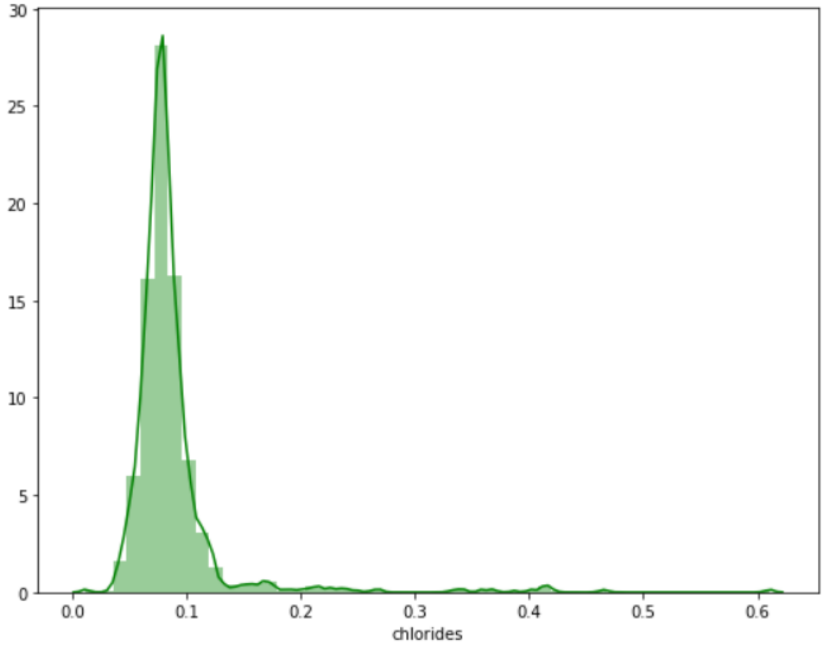
CITRIC ACID :



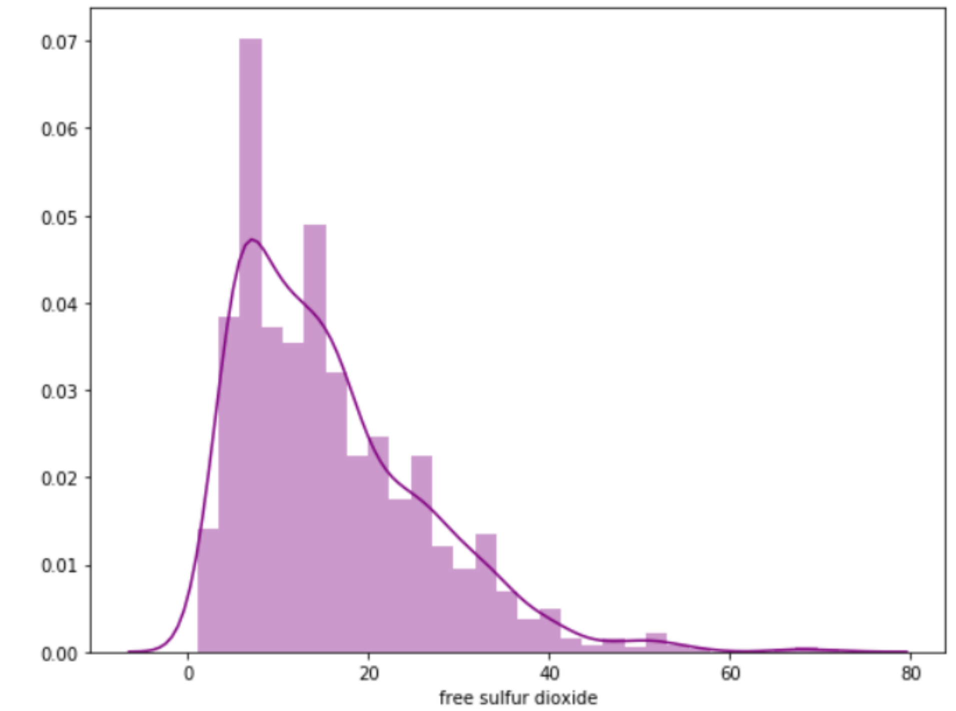
RESIDUAL SUGAR :



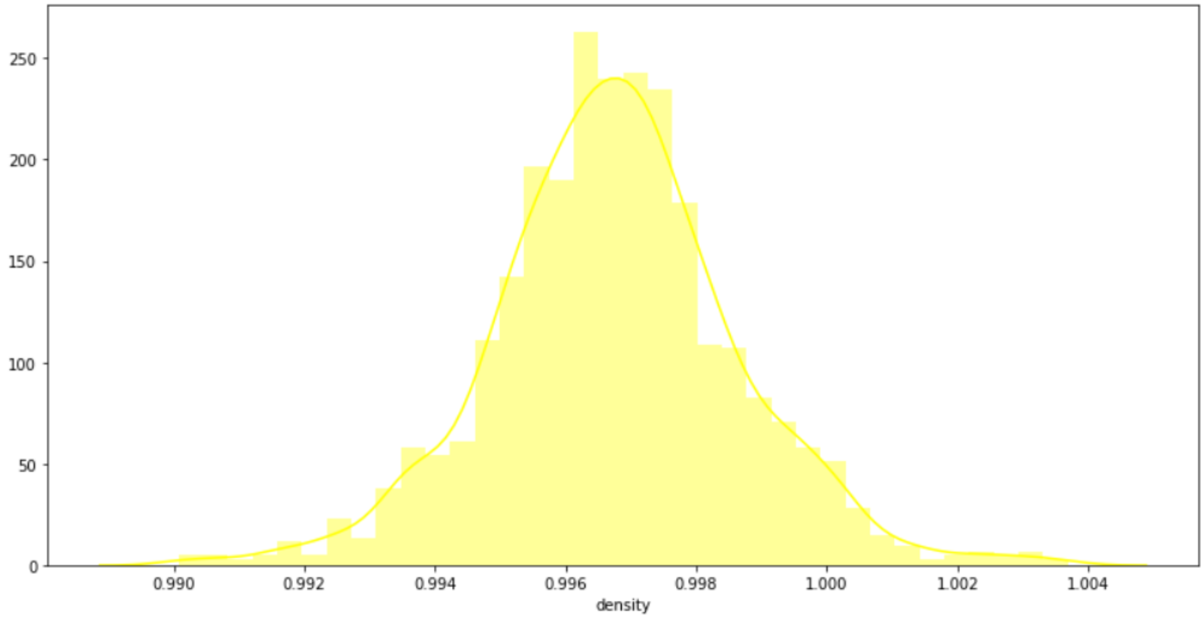
CHLORIDES :



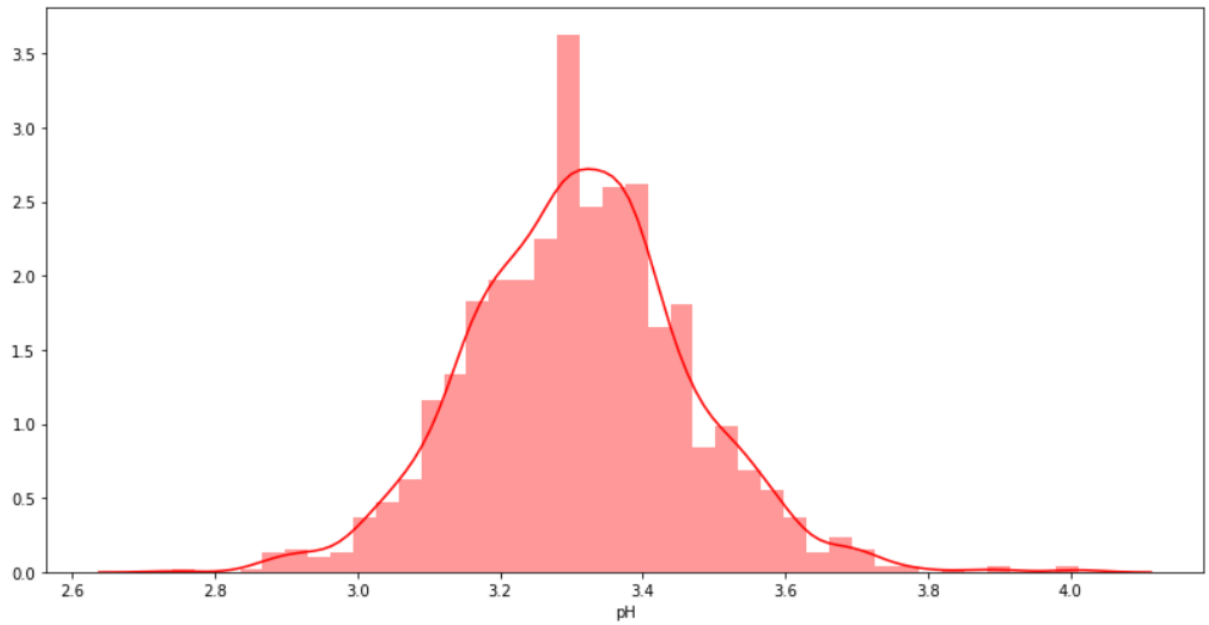
FREE SULPHUR DIOXIDE :



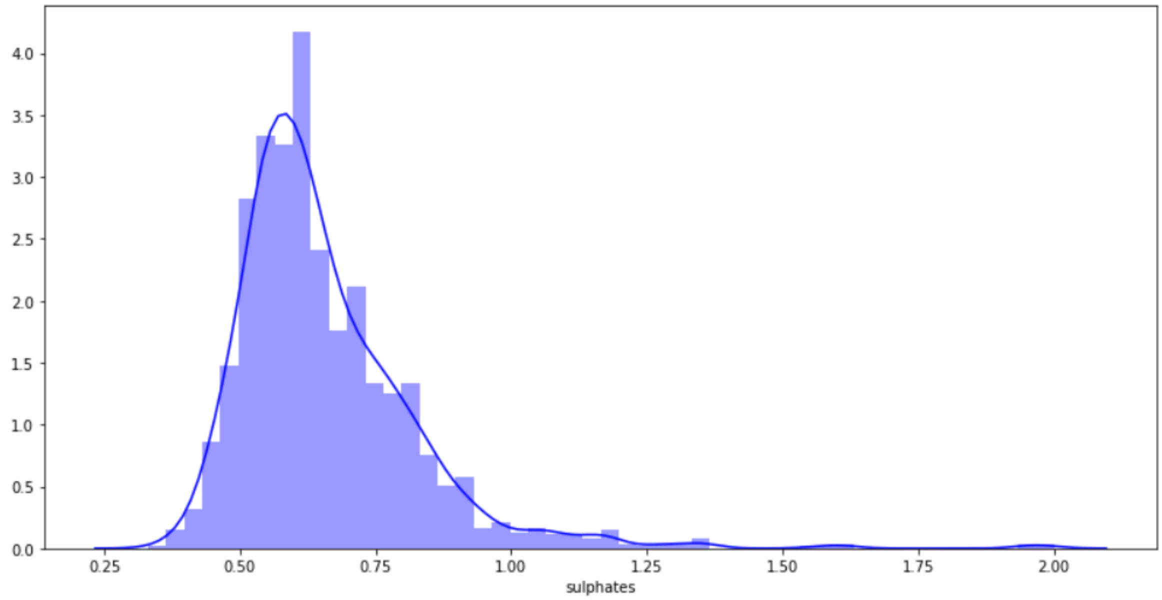
DENSITY :



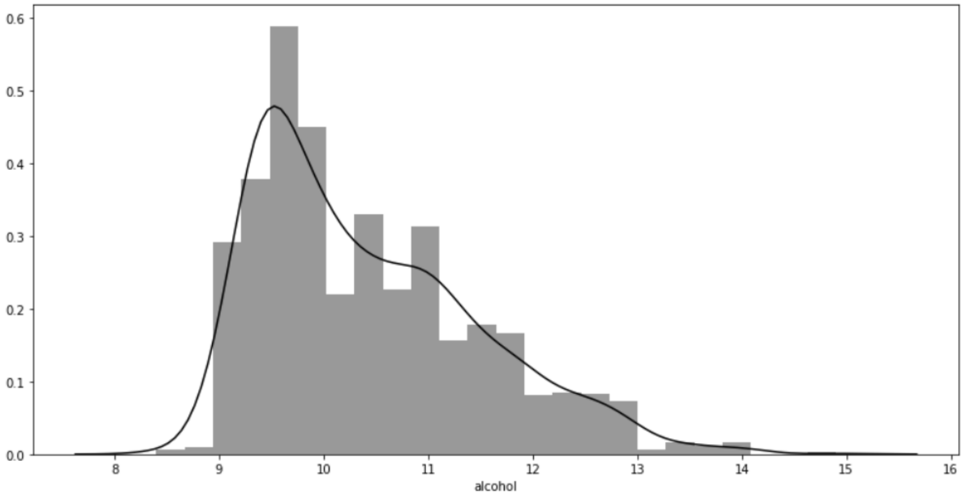
pH :



SULPHATES :

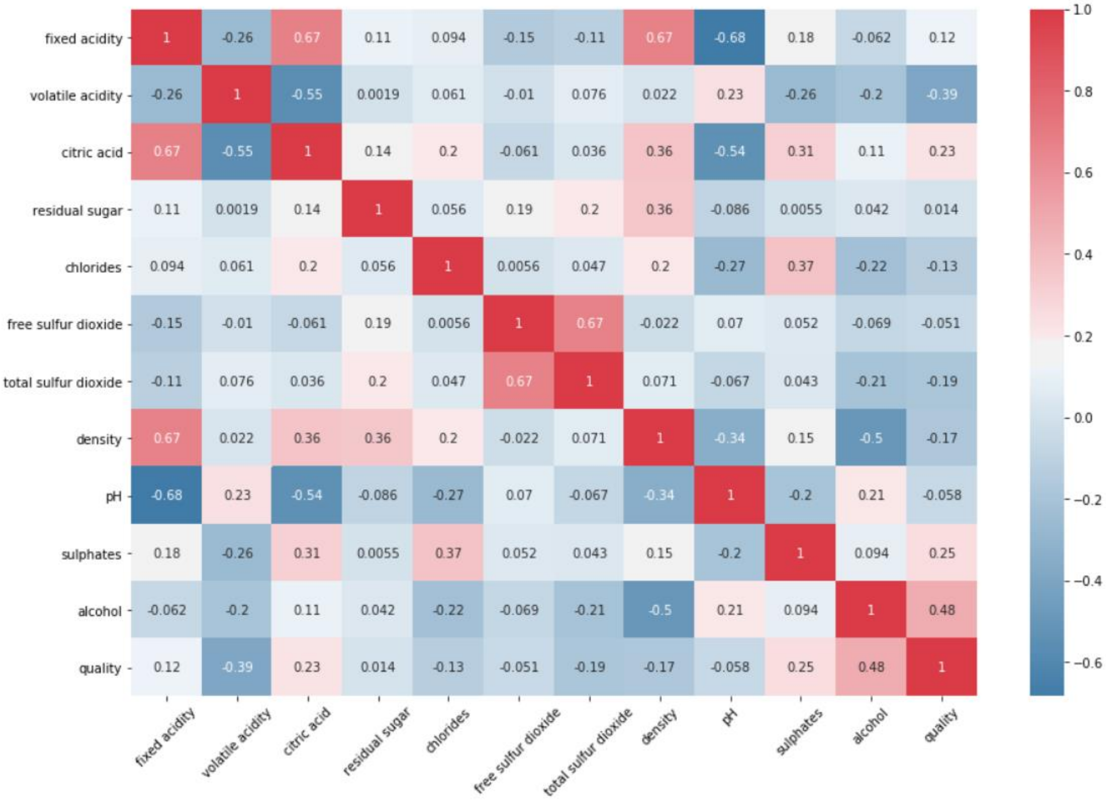


ALCOHOL:



CORRELATION MATRIX :

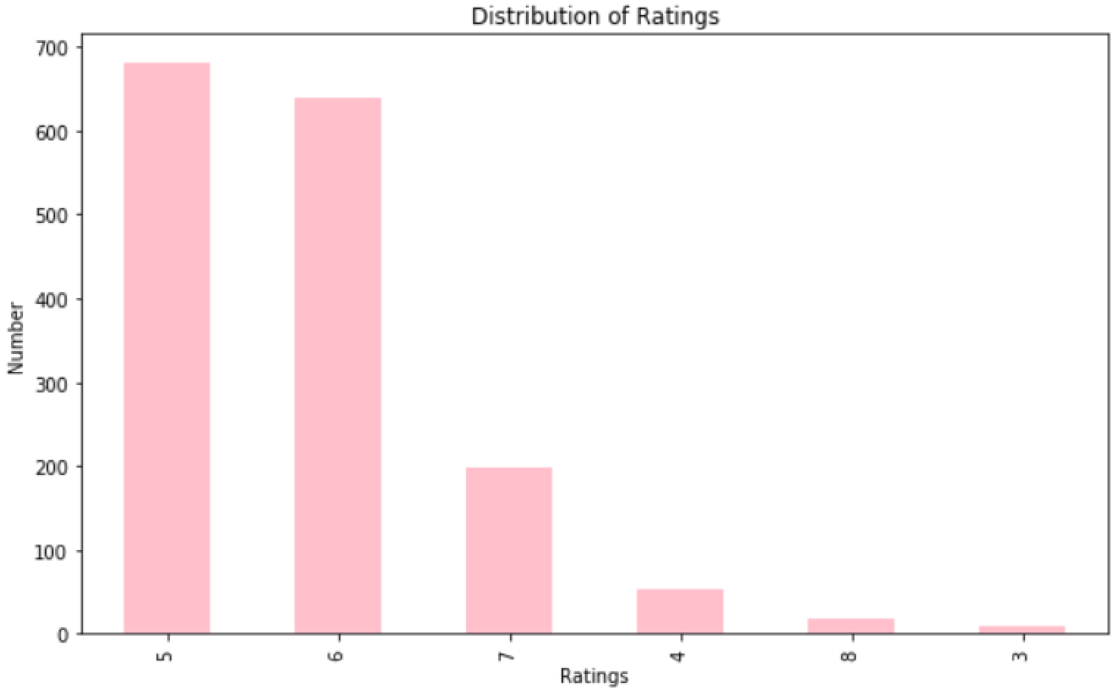
Because reading graphs can be very difficult, it's also a good idea to plot a matrix of correlations. This will offer insights on which variables correlate more quickly.



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1	-0.26	0.67	0.11	0.094	-0.15	-0.11	0.67	-0.68	0.18	-0.062	0.12
volatile acidity	-0.26	1	-0.55	0.0019	0.061	-0.01	0.076	0.022	0.23	-0.26	-0.2	-0.39
citric acid	0.67	-0.55	1	0.14	0.2	-0.061	0.036	0.36	-0.54	0.31	0.11	0.23
residual sugar	0.11	0.0019	0.14	1	0.056	0.19	0.2	0.36	-0.086	0.0055	0.042	0.014
chlorides	0.094	0.061	0.2	0.056	1	0.0056	0.047	0.2	-0.27	0.37	-0.22	-0.13
free sulfur dioxide	-0.15	-0.01	-0.061	0.19	0.0056	1	0.67	-0.022	0.07	0.052	-0.069	-0.051
total sulfur dioxide	-0.11	0.076	0.036	0.2	0.047	0.67	1	0.071	-0.067	0.043	-0.21	-0.19
density	0.67	0.022	0.36	0.36	0.2	-0.022	0.071	1	-0.34	0.15	-0.5	-0.17
pH	-0.68	0.23	-0.54	-0.086	-0.27	0.07	-0.067	-0.34	1	-0.2	0.21	-0.058
sulphates	0.18	-0.26	0.31	0.0055	0.37	0.052	0.043	0.15	-0.2	1	0.094	0.25
alcohol	-0.062	-0.2	0.11	0.042	-0.22	-0.069	-0.21	-0.5	0.21	0.094	1	0.48
quality	0.12	-0.39	0.23	0.014	-0.13	-0.051	-0.19	-0.17	-0.058	0.25	0.48	1

DATA DISTRIBUTION :

To see how they are distributed, we can measure data distributions or value count and draw a histogram for each variable. For the quality data, the following figure shows the histogram.



CHAPTER 6

CONCLUSION

While participating in this research, and for the duration of this project we have come to the conclusion after our regress analysis and study on the matter that food and beverage industries like- Red-Wine industry, are in desperate and immediate needs of implementation of automated and machine-enabled quality assessment processes. Further research in this field should take place which will help these companies save cost and help with QA for the customers. We consider our work and report to be a success and with given opportunities try to expand upon our work.

CHAPTER 7

BIBLIOGRAPHY

1. https://upload.wikimedia.org/wikipedia/commons/7/76/Random_forest_diagram_complete.png
2. https://analyticsindiamag.com/7-types-classification-algorithms/
3. https://en.wikipedia.org/wiki/Gradient_boosting/
4. https://medium.com/analytics-vidhya/predicting-red-wine-quality-using-machine-learning-model-34e2b1b8d498
5. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiKptvhyJTtAhVRJHIKHR17CIMQFjAHegQICBAC&url=https%3A%2F%2Fblog.paperspace.com%2Fadaboost-optimizer%2F%23%3A~%3Atext%3DAdaBoost%2520is%2520an%2520ensemble%2520learning%2Cturn%2520them%2520into%2520strong%2520ones.&usg=AOvVaw0csuWfJ3tF4MkSJFfznE_w
6. https://miro.medium.com/proxy/1*m2UHkzWWJ0kfQyL5tBFNsQ.png

7. https://gblobscdn.gitbook.com/assets%2F-LvBP1svpACTB1R1x_U4%2F-Lw6zezdliKWkGknCJ6R%2F-Lw70EB_T-Y3OCO-L_4o%2Fimage.png?alt=media&token=a3edaf4a-d3d2-4c84-9a10-3d870c21d641
8. https://towardsdatascience.com/predicting-wine-quality-with-several-classification-techniques-179038ea6434#ca0a
9. https://www.kaggle.com/madhurisivalenka/basic-machine-learning-with-red-wine-quality-data
10. https://www.geeksforgeeks.org/prediction-of-wine-type-using-deep-learning/
11. https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/#:~:text=Role%20of%20the%20Activation%20Function,transferred%20to%20the%20next%20layer.
12. http://rstudio-pubs-static.s3.amazonaws.com/438329_edfaab4011ce44a59fb9ae2d216d8dea.html
13. https://www.neuraldesigner.com/learning/examples/wine-quality-improvement
14. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6429329/
15. https://www.koreascience.or.kr/article/JAKO201832073079660.pdf
16. https://www.datacamp.com/community/tutorials/deep-learning-python
17. Artificial neural network architecture (ANN i-h 1-h 2-h n-o). | Download Scientific Diagram (researchgate.net)