RED WINE QUALITY PREDICTION

Mini Project Report Submitted in partial fulfilment

of the requirement for under graduate degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

By

K. B. Indraneel (221710301030)

D. Akhil Kumar (221710301012)

O. Sai Ram Reddy (221710301046)

A. Kartheek (221710301004)

Under the Guidance of

Ms. G. Mounika

Assistant Professor

GITAM
(DEEMED TO BE UNIVERSITY)
VISAKHAPATNAM • HYDERABAD • BENGALURU

Department Of Computer Science and Engineering
GITAM School of Technology
GITAM (Deemed to be University)
Hyderabad-502329
July 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF TECHNOLOGY

GITAM

(Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS

DECLARATION

We submit this mini project work entitled **"Red wine quality prediction**" to GITAM (Deemed to be University), Hyderabad in partial fulfilment of the requirements for the award of the degree of "**Bachelor of Technology**" in "**Computer Science and Engineering**". We declare that it was carried out independently by us under the guidance of **(Ms. G. Mounika)**, Asst. Professor, GITAM (Deemed to be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: Hyderabad

Date:

Name and Signature of Candidate

K. B. Indraneel (221710301030)

D. Akhil Kumar (221710301012)

O. Sai Ram Reddy (221710301046)

A. Kartheek (221710301004)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS

CERTIFICATE

This is to certify that the Mini Project Report entitled - "**Red Wine Quality Prediction**" is being submitted by **K. B. Indraneel (221710301030)**, **D. Akhil Kumar (221710301012), O. Sai Ram Reddy (221710301046), A. Kartheek (221710301004)** submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering**.

Guided by

Ms. G. Mounika

Head of the Department

Dr. Phani Kumar

Professor & HOD

ACKNOWLEDGEMENT

Our project would not have been successful without the help of several people, we would like to thank the personalities who were part of our project in numerous ways, those who gave us outstanding support from the birth of the project.

We would like to thank our honorable Pro-Vice-Chancellor, **Prof. N. Siva Prasad** for providing necessary infrastructure and resources for the accomplishment of our project.

We would like to thank respected **Prof. N. Seetharamaiah**, Principal, School of Technology, for his support during the tenure of the project.

We would like to thank respected **Prof. S. Phani Kumar**, Head of the Department of Computer Science & Engineering for providing the opportunity to undertake this project and encouragement in the completion of this project.

We would like to thank respected **Ms. G. Mounika**, Assistant Professor in Computer Science Department for the esteemed guidance, moral support and invaluable advice provided by her for the success of the project.

We would like to thank our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Sincerely,

K. B. Indraneel (221710301030)

D. Akhil Kumar (221710301012)

O. Sai Ram Reddy (221710301046)

A. Kartheek (221710301004)

ABSTRACT

Nowadays people try to lead a luxurious life. These days the consumption of red wine is very common to all. So it became important to analyze the quality of red wine before its consumption to preserve human health. The quality of any wine is intrinsically dependent on the quality and composition of the grapes used to produce it. In traditional winemaking countries such as Germany and France, wine quality is determined by geographic origin or the terroir (factors such as the soil, topography and climate) of the wine. Hence this project's aim is a step towards the quality prediction of the red wine using its various attributes.

The dataset used is red wine quality and is taken from kaggle website. The attributes of the wine such as Volatility, citric acid, residual sugar, chloride content, sulfur content, Ph value, alcohol content and color are taken as input variables and quality of the wine is obtained as output. The Machine learning algorithms used in this process of testing the quality of wine are Random Forest, KNN and Decision tree. The dataset is tested against the three algorithms and the algorithm which gives the best accuracy is considered. Various measures are calculated and the results are compared among training set, testing set and accordingly the best out of the three models depending on the testing set results is predicted.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

MACHINE LEARNING

1.1 INTRODUCTION

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is also defined as the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions. It is seen as a subset of artificial intelligence.

Machine learning algorithms build a model based on sample data known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers but not all machine learning is statistical learning.

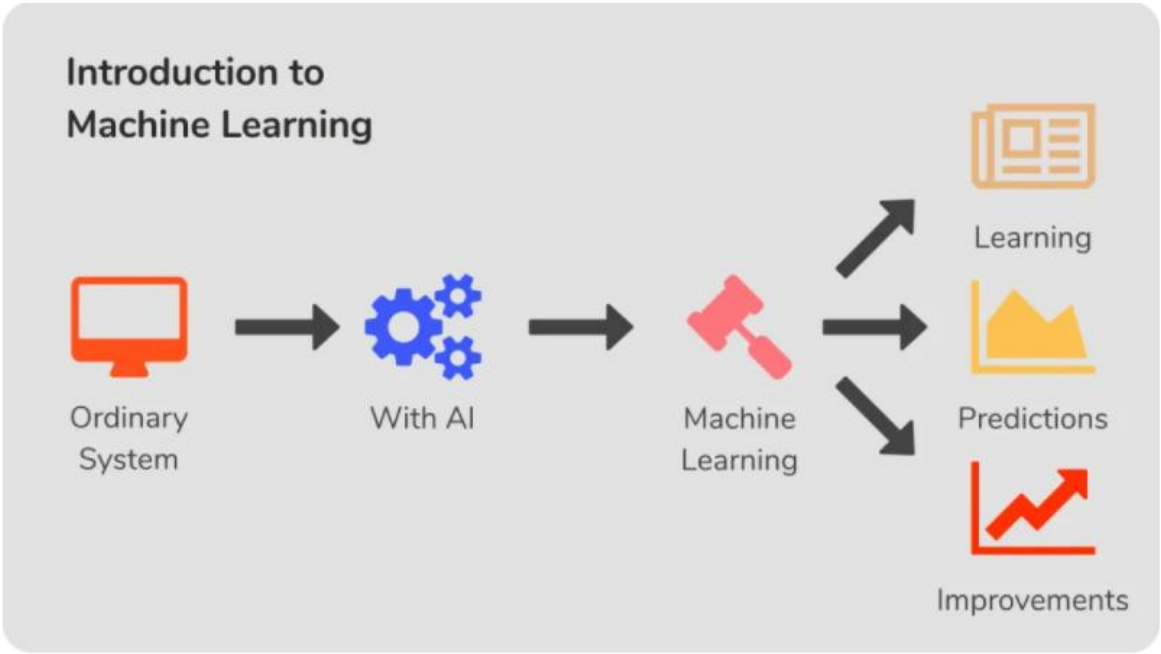

Fig 1.1 Introduction to Machine Learning

1.2 IMPORTANCE OF MACHINE LEARNING

In a near future, automation process will superimpose most of the human-work in manufacturing. To match human capabilities, devices need to be intelligent and Machine Learning is at the core of AI. Machine learning allows companies to transform processes that were previously only possible for humans to perform like responding to customer service calls, bookkeeping, and reviewing resumes for everyday businesses. Machine learning can also scale to handle larger problems and technical questions like image detection for self-driving cars, predicting natural disaster locations and timelines, and understanding the potential interaction of drugs with medical conditions before clinical trials. That's why machine learning is important.

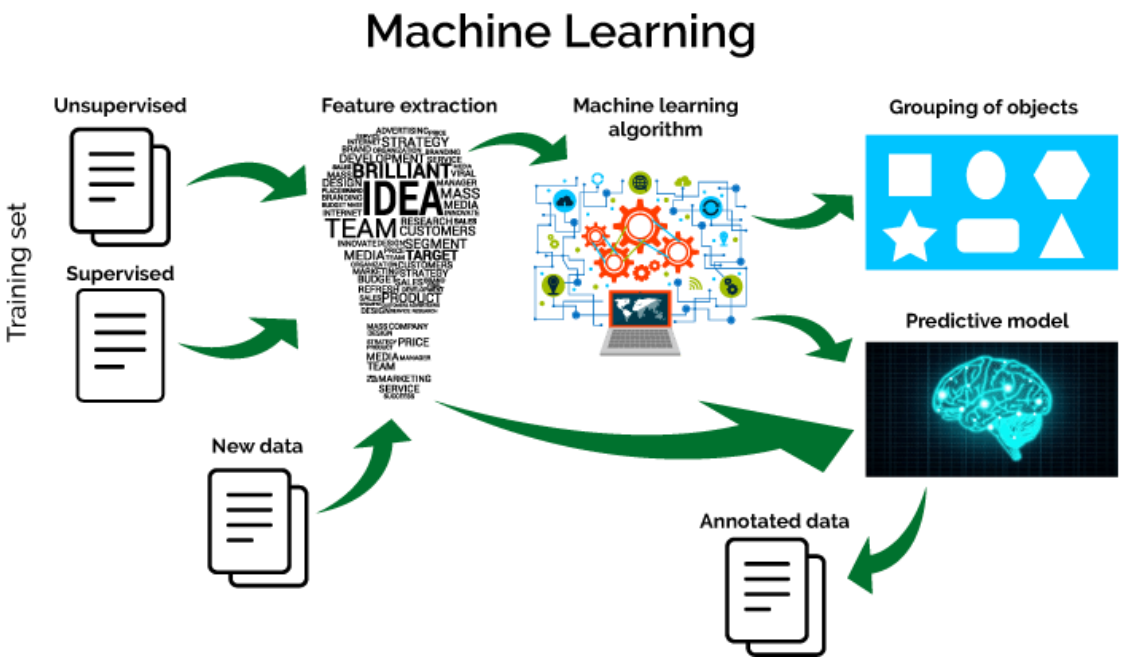

Fig 1.2 The Process Flow

1.3 APPLICATIONS OF MACHINE LEARNING

1. Healthcare: The sensors and devices that monitor everything from pulse rates and steps walked to oxygen and sugar levels and even sleeping patterns have generated a significant volume of data that enables doctors to assess their patients health in real-time. ML algorithms can detect cancerous tumors on mammograms, identifies skin cancer, can analyze retinal images to diagnose diabetic retinopathy.

2. Government: Systems that use machine learning enable government officials to use data to predict potential future scenarios and adapt to rapidly changing situations. ML can help to improve cyber security and cyber intelligence, support counterterrorism efforts, optimize operational preparedness, logistics management, and predictive maintenance, and reduce failure rates.

3. Marketing and sales: Machine learning is even revolutionizing the marketing sector as many companies have successfully implemented artificial intelligence and machine learning to increase and enhance customer satisfaction by over 10%. In fact, according to Forbes, 57% of enterprise executives believe that the most important growth benefit of AI and ML will be improving customer experiences and support.

4. E-commerce: Social media sites use machine learning to analyze customer's buying and search history and make recommendations on other items to purchase, based on the past habits. Many experts theorize that the future of retail will be driven by AI and ML as deep learning business applications become even more adept at capturing, analyzing, and using data to personalize individuals' shopping experiences and develop customized targeted marketing campaigns.

5. Transportation: Efficiency and accuracy are key to profitability within this sector, so is the ability to predict and mitigate potential problems. ML's data analysis and modeling functions dovetail perfectly with businesses within the delivery, public transportation and freight transport sectors. ML uses algorithms to find factors that positively and negatively impact a supply chain's success, making machine learning a critical component within supply chain management.

6. Financial services: The insights provided by ML in this industry allow investors to identify new opportunities or know when to trade. Data mining pinpoints high-risk clients and informs cyber surveillance to find and mitigate signs of fraud. ML can help calibrate financial portfolios or assess risk for loans and insurance underwriting.

7. Oil and gas: ML and AI are already working to find new energy sources and analyze mineral deposits in the ground, predict refinery sensor failure, and streamline oil distribution to increase efficiency and shrink costs. ML is revolutionizing the industry with its case-based reasoning, reservoir modeling and drill floor automation. Machine learning is helping to make this dangerous industry safer.

8. Manufacturing: Machine learning is no stranger to the vast manufacturing industry, either. Machine learning applications in manufacturing are about accomplishing the goal of improving operations from conceptualization to final delivery, significantly reducing error rates, improving predictive maintenance and increasing inventory turn.

1.4 CLASSIFICATION OF LEARNING MODELS

Machine learning implementations are classified into three major categories, depending on the nature of the learning.

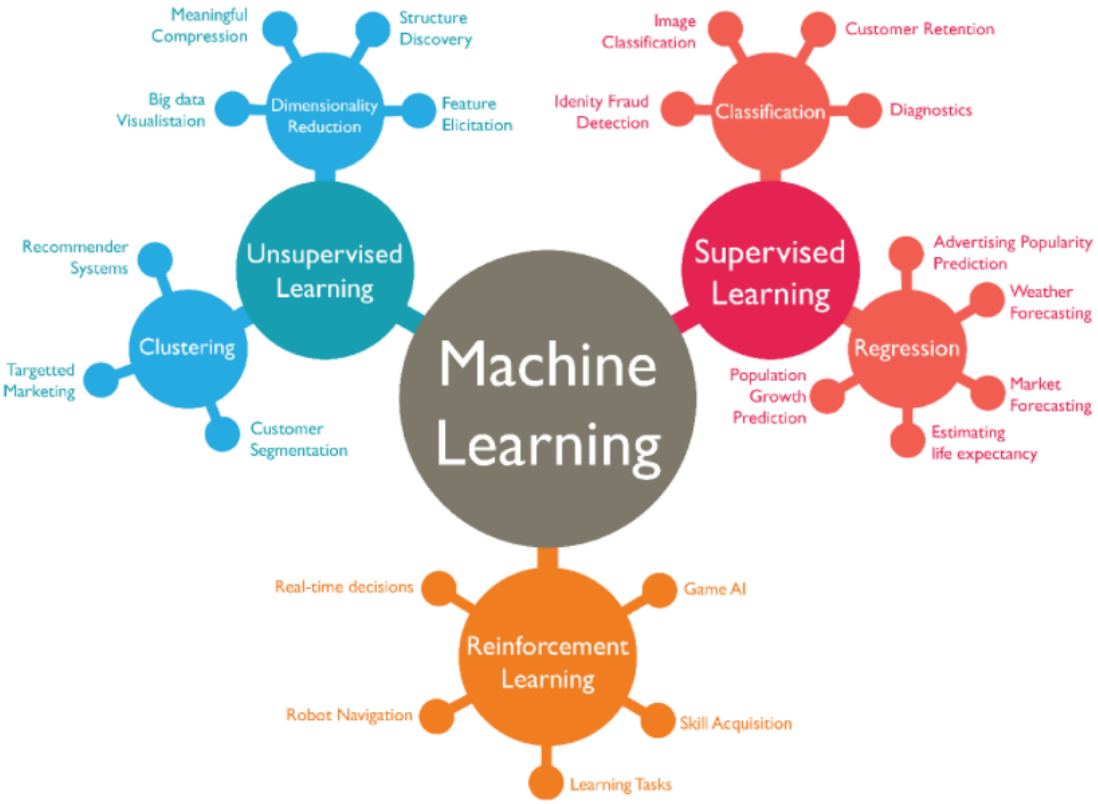

Fig 1.3 Classification of Learning Models

1.4.1 Supervised learning: Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

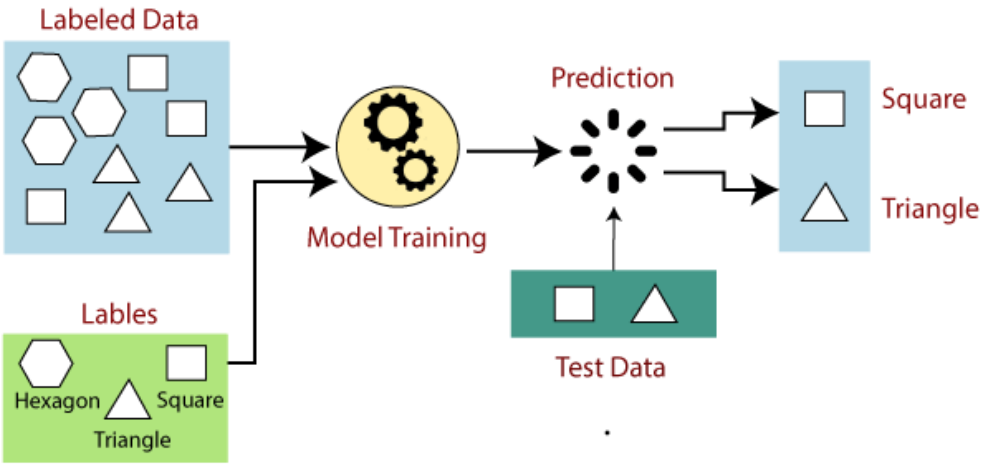

Fig 1.4 Supervised Learning

1.4.2 Unsupervised learning: Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own it is known as unsupervised learning.

This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

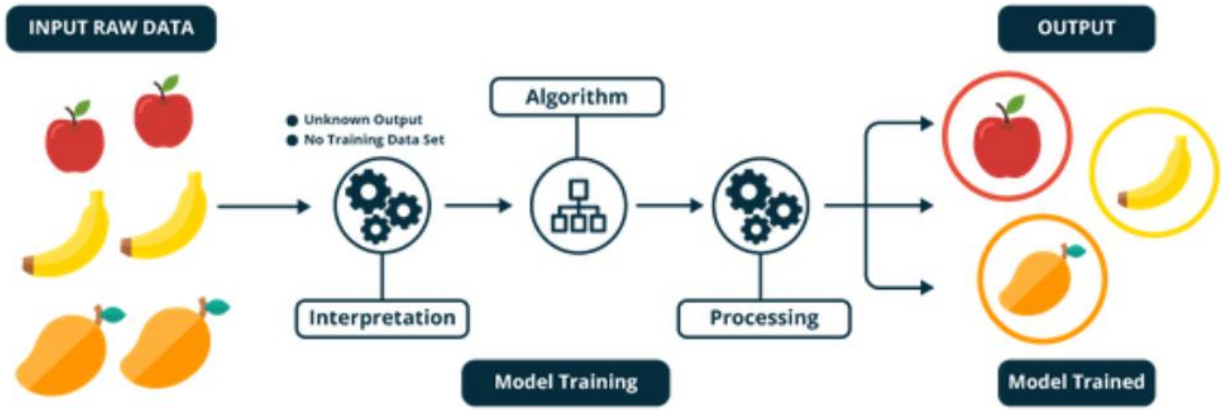

Fig 1.5 Unsupervised Learning

1.4.3 Reinforcement learning: When the algorithm is presented with examples that lack labels, as in unsupervised learning. However, one can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error.

Errors help one to learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching that a certain course of action is less likely to succeed than others. An interesting example of reinforcement learning occurs when computers learn to play video games by themselves.

In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival

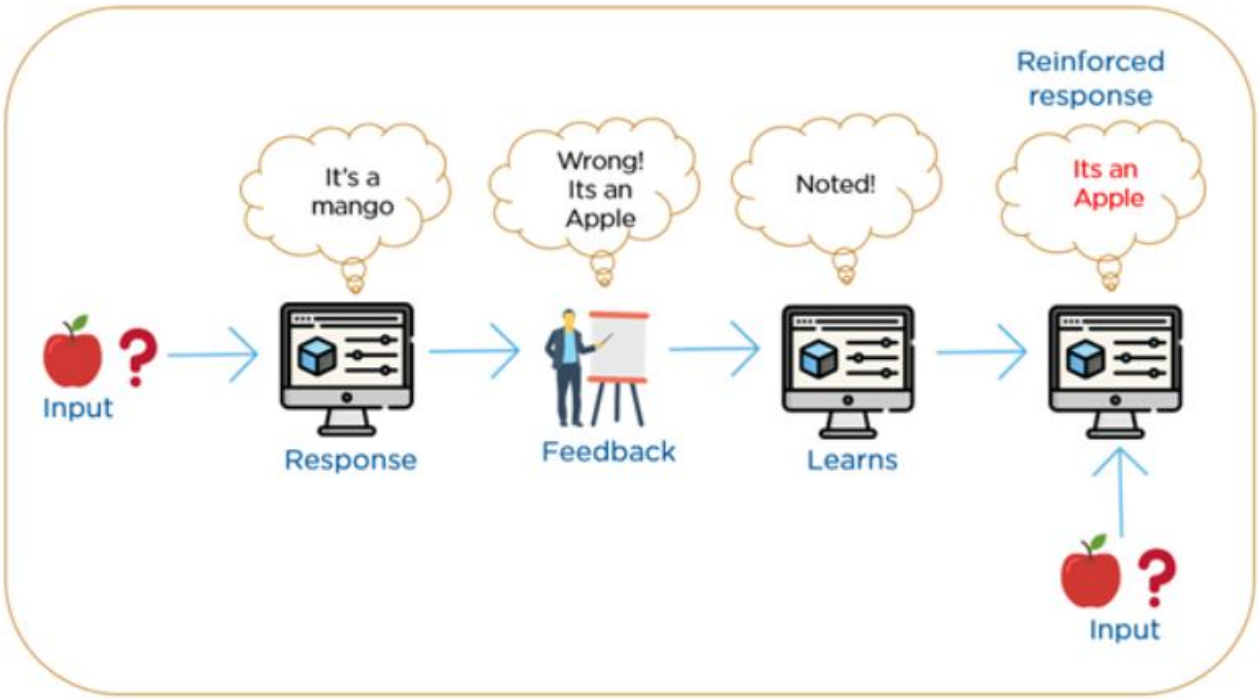

Fig 1.6 Reinforced Learning

1.5 TYPES OF SUPERVISED LEARNING

Supervised learning can be further classified into classification and regression.

1.5.1 Regression

Regression algorithms predict a continuous value based on the input variables. The main goal of regression problems is to estimate a mapping function based on the input and output variables. If target variable is a quantity like income, scores, height or weight, or the probability of a binary category like the probability of rain in particular regions, then one should use the regression model. The different types of regression algorithms include:

A. Simple linear regression: With simple linear regression, one can estimate the relationship between one independent variable and another dependent variable using a straight line, given both variables are quantitative.

B. Multiple linear regression: An extension of simple linear regression, multiple regression can predict the values of a dependent variable based on the values of two or more independent variables.

C. Polynomial regression: The main aim of polynomial regression is to model or find a nonlinear relationship between dependent and independent variables.

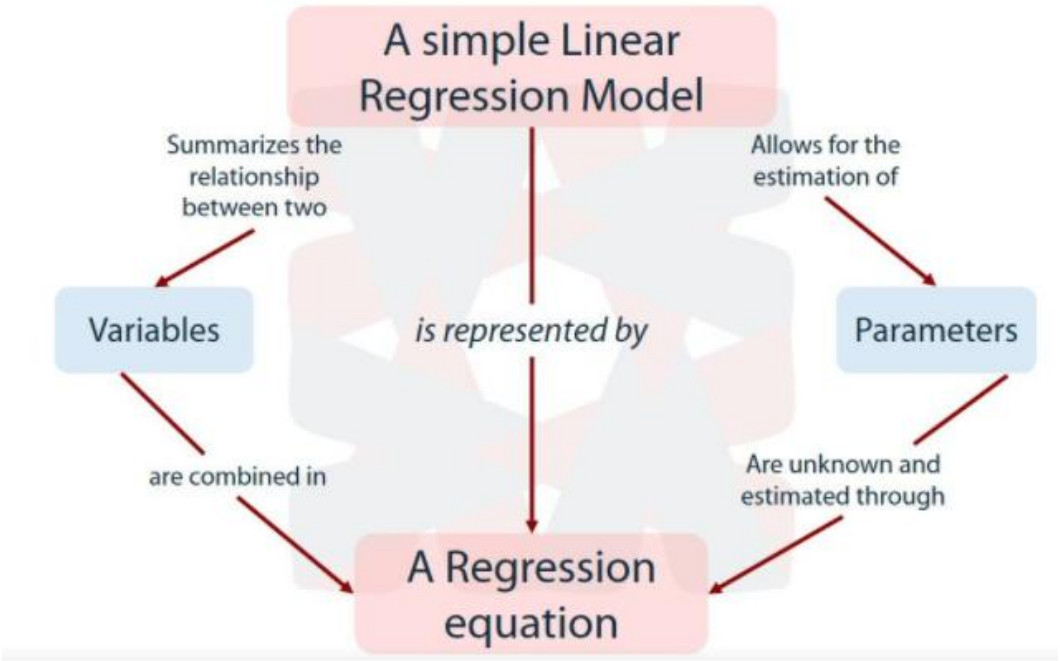

Fig 1.7 Regression Model

1.5.2 CLASSIFICATION

Classification is a predictive model that approximates a mapping function from input variables to identify discrete output variables that can be labels or categories. The mapping function of classification algorithms is responsible for predicting the label or category of the given input variables. A classification algorithm can have both discrete and real-valued variables, but it requires that the examples be classified into one of two or more classes.

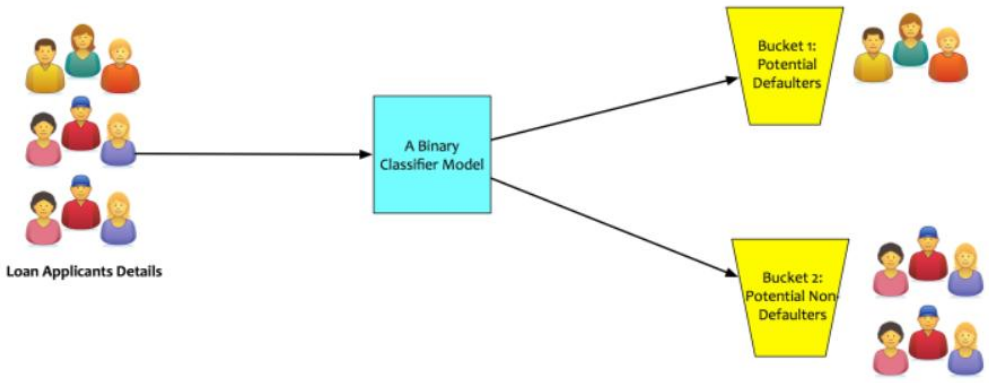

Fig 1.8 Classification Model

The different types of classification algorithms include:

1. Decision tree classification

In this algorithm, a classification model is created by building a decision tree where every node of the tree is a test case for an attribute and each branch coming from the node is a possible value for that attribute.

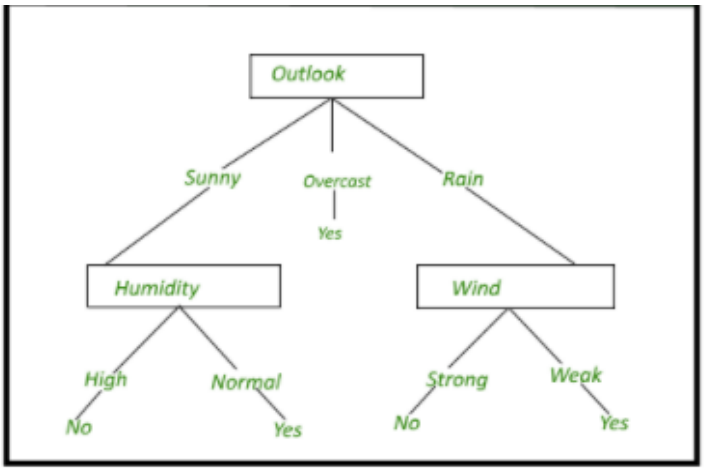

Fig 1.9 Decision Tree for Playing Tennis

2. Random forest classification

This tree-based algorithm includes a set of decision trees which are randomly selected from a subset of the main training set. The random forest classification algorithm aggregates outputs from all the different decision trees to decide on the final output prediction, which is more accurate than any of the individual trees.

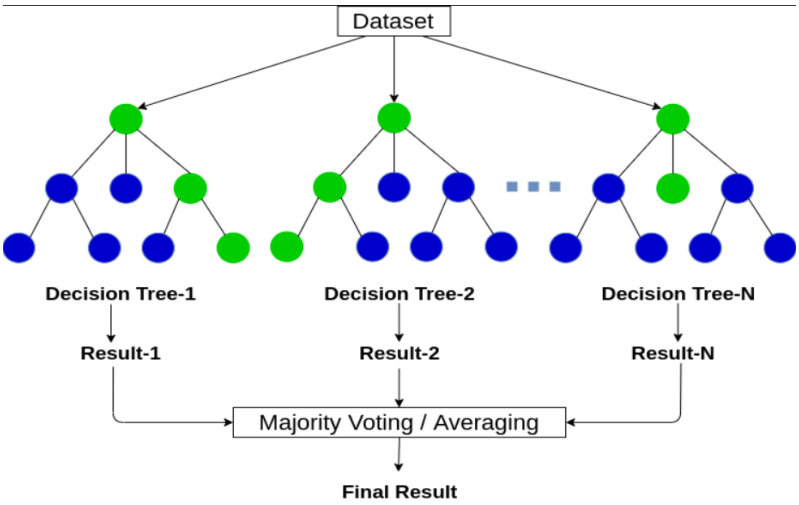

Fig 1.10 Random Forest

3. K-nearest neighbor

The K-nearest neighbor algorithm assumes that similar things exist in close proximity to each other. It uses feature similarity for predicting values of new data points. The algorithm helps grouping similar data points together according to their proximity. The main goal of the algorithm is to determine how likely it is for a data point to be a part of the specific group.

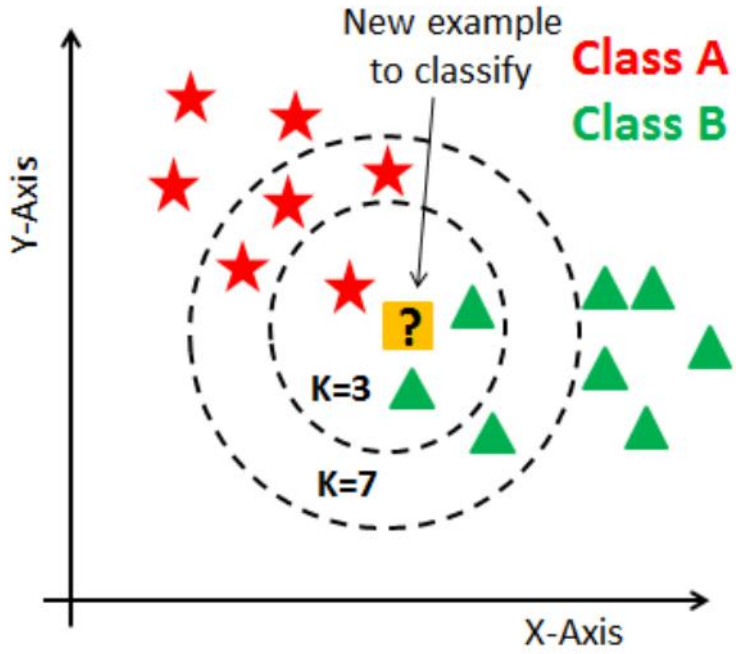

Fig 1.11 K-Nearest Neighbor

CHAPTER-2

PYTHON

2.1 INTRODUCTION

Python is an open-source (free) programming language that is used in web programming, data science, artificial intelligence, and many scientific applications. Learning Python allows the programmer to focus on solving problems, rather than focusing on syntax. Its relative size and simplified syntax give it an edge over languages like Java and C++, yet the abundance of libraries gives it the power needed to accomplish great things.

Following are the features of python:

1. Python is a high-level, interpreted, interactive and object-oriented scripting language.
2. Python is a general purpose programming language that is often applied in scripting roles.
3. Python is interpreted since it is processed at runtime by the interpreter.
4. Python is Object-Oriented since it supports the Object-Oriented style of programming that encapsulates code within objects.

2.2 HISTORY OF PYTHON

1. Python was developed by GUIDO VAN ROSSUM in early 1990's
2. Python 2.0 was released on 16 October 2000 with many major new features which includes a cycle-detecting garbage collector and support for Unicode.
3. Python 3.0 was released on 3 December 2008. Many of its major features were back ported to python 2.6 and 2.7 version series.

2.3 PYTHON VARIABLES

1. Variables are nothing but reserved memory locations to store values. This means that when a variable is created some space in memory is reserved for it.
2. Python has no command for declaring a variable. A variable is created the moment when a value is assigned to it.
3. Python variables do not explicitly reserve memory space. The declaration happens automatically when a value is assigned to variable.

4. Python has five standard data types

- Strings
- Lists
- Tuples
- Sets
- Dictionary

2.4 STRINGS IN PYTHON

1. Strings in Python are arrays of bytes representing unicode characters.
2. Python does not have a character data type, a single character is simply a string with a length of 1.
3. Strings in python are surrounded by either single quotation marks or double quotation marks i.e. a = "Hello" or a= 'Hello'.
4. Assigning a string to a variable is done with the variable name followed by an equal sign and the string.
5. Square brackets can be used to access elements of the string.

2.5 PYTHON LISTS

1. Lists are used to store multiple items in a single variable irrespective of their data type and are created using square brackets.
2. List items are ordered, changeable and allow duplicate values.
3. List items are indexed, the first item has index [0], second item has index [1] and so on.
4. The list is changeable, meaning one can change, add and remove items in a list after it has been created.
5. Examples of list:

   list1 = ["apple", "banana", "cherry"]

   list2 = [1, 5, 7, 9, 3]

   list3 = [True, False, False]

2.6 PYTHON TUPLES

1. Tuples are used to store multiple items in a single variable and are represented with round brackets.
2. Tuple items are ordered, unchangeable, and allow duplicate values.
3. Tuples are unchangeable, meaning that one cannot change, add or remove items after the tuple has been created.
4. Tuple items are indexed, first item has index [0], second item has index [1] and so on.
5. Examples of tuple

```
Tuple = ("apple", "banana", "cherry")
```

2.7 PYTHON SETS

1. Sets are used to store multiple items in a single variable.
2. A set is a collection which is both unordered and unindexed and are represented with curly brackets.
3. Set items are unordered, unchangeable, and do not allow duplicate values.
4. Unordered means that set items can appear in a different order every time when it is used and cannot be referred to by index or key.
5. Sets are unchangeable, meaning that we cannot change the items after the set has been created.
6. Example of set

```
set = {"apple", "banana", "cherry", "apple"}
```

2.8 PYTHON DICTIONARIES

1. Dictionaries are used to store data values in key: value pairs.
2. A dictionary is a collection which is unordered, changeable and does not allow duplicates.
3. Dictionaries are represented with curly brackets and have keys and values.
4. Dictionary items are presented in key: value pairs, and can be referred to by using the key name.
5. Dictionaries are changeable, meaning that one can change, add or remove items after the dictionary has been created.
6. Example: Dict = { "brand": "Ford", "model": "Mustang", "year": 1964, "year": 2020 }

CHAPTER-3

LITERATURE SURVEY

The project entitled “Red Wine Quality Prediction” is based on the research paper published on IEEE of DOI 10.1109/ICCCI48352.2020.9104095. The algorithms used in this paper were Support Vector Machine, Random Forest and Naïve Bayes in order to predict the quality of red wine. After the implementation it was found that Support Vector Machine was the best among three in predicting the quality of wine with an accuracy of 67.25% followed by Random forest and Naïve Bayes with 65.83% and 55.91%.

In this project the algorithms which are used to predict the quality of red wine would be Decision Tree, Random Forest and K-Nearest Neighbor. Random Forest is intrinsically suited for multiclass problems while SVM is intrinsically used for two class classifications. For multiclass problem one need to reduce it into multiple binary classification problems in order to use SVM. Similarly Decision Tree comes under tree based methods which are considered non-parametric, making no assumption on the distribution of data and the structure of the true model and they require less data cleaning when compared to others. Naive Bayes is a linear classifier and in general Naive Bayes is highly accurate when applied to big data. The main advantage of KNN is it doesn’t require any training one can just load the dataset and off it runs while on the other hand Naive Bayes does require training. Hence, considering all the pros and cons of the above algorithms one can predict that Random forest or Decision tree would predict the quality of the red wine in better way with higher accuracy.

CHAPTER-4

CASE STUDY

4.1 PROBLEM STATEMENT

Red wine quality prediction: Nowadays people try to lead a luxurious life. The consumption of red wine is very common to all. So it became important to analyze the quality of red wine before its consumption to preserve human health. The quality of any wine is intrinsically dependent on the quality and composition of the grapes used to produce it. The main goal of this problem is to test the quality of the wine with the help of its attributes like acidity, ph value, density, so_2 etc. Since wine is a huge profit making market there is always a scope of adultering it's quality which shows negative impact on health of the one who consumes it. Generally it takes few days in order to test the quality of wine in laboratory and involves huge cost overheads in order to maintain the laboratory. So with the help of this model time and money can be saved in testing the quality of wine.

4.2 DATA SET

The data set used in this project has been acquired from "Kaggle" website.

The data set consists of 1599 rows and 12 columns or attributes.

1. **Fixed acidity**: Most acids involved with wine are fixed or nonvolatile (does not evaporate readily).

2. **Volatile acidity**: The amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste.

3. **Citric acid**: Found in small quantities, citric acid can add freshness and flavor to wines.

4. **Residual sugar**: The amount of sugar remaining after fermentation stops. It's rare to find wines with sugars less than 1 gram/liter and wines with sugars greater than 45 grams/liter are considered sweet.

5. **Chlorides**: The amount of salt in the wine.

6. **Free sulfur dioxide**: The free form of SO_2 exists in equilibrium between molecular SO_2 and bisulfite ion. It prevents microbial growth and the oxidation of wine.

7. **Total sulfur dioxide**: Amount of free and bound forms of S02 in low concentrations. SO_2 is mostly undetectable in wine but at free SO_2 concentrations over 50 ppm, SO_2 becomes evident in the nose and taste of wine.

8. **Density**: The density of wine is close to that of water depending on the percent alcohol and sugar content.

9. **PH Value**: Describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic). Most wines are between 3-4 on the ph scale.

10. **Sulphates:** A wine additive which can contribute to sulfur dioxide gas levels, which acts as an antimicrobial and antioxidant.

11. **Alcohol**: The percent alcohol content of the wine.

12. **Quality**: Output variable based on sensory data, score is between 0 and 10.

fixed acidi	volatile ac	citric acid	residual su	chlorides	free sulfur	total sulfu	density	pH	sulphates	alcohol	quality
7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5
7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	7
7.8	0.58	0.02	2	0.073	9	18	0.9968	3.36	0.57	9.5	7
7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	5
6.7	0.58	0.08	1.8	0.097	15	65	0.9959	3.28	0.54	9.2	5
7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	5
5.6	0.615	0	1.6	0.089	16	59	0.9943	3.58	0.52	9.9	5
7.8	0.61	0.29	1.6	0.114	9	29	0.9974	3.26	1.56	9.1	5
8.9	0.62	0.18	3.8	0.176	52	145	0.9986	3.16	0.88	9.2	5
8.9	0.62	0.19	3.9	0.17	51	148	0.9986	3.17	0.93	9.2	5
8.5	0.28	0.56	1.8	0.092	35	103	0.9969	3.3	0.75	10.5	7
8.1	0.56	0.28	1.7	0.368	16	56	0.9968	3.11	1.28	9.3	5
7.4	0.59	0.08	4.4	0.086	6	29	0.9974	3.38	0.5	9	4
7.9	0.32	0.51	1.8	0.341	17	56	0.9969	3.04	1.08	9.2	6
8.9	0.22	0.48	1.8	0.077	29	60	0.9968	3.39	0.53	9.4	6
7.6	0.39	0.31	2.3	0.082	23	71	0.9982	3.52	0.65	9.7	5
7.9	0.43	0.21	1.6	0.106	10	37	0.9966	3.17	0.91	9.5	5
8.5	0.49	0.11	2.3	0.084	9	67	0.9968	3.17	0.53	9.4	5
6.9	0.4	0.14	2.4	0.085	21	40	0.9968	3.43	0.63	9.7	6
6.3	0.39	0.16	1.4	0.08	11	23	0.9955	3.34	0.56	9.3	5
7.6	0.41	0.24	1.8	0.08	4	11	0.9962	3.28	0.59	9.5	5
7.9	0.43	0.21	1.6	0.106	10	37	0.9966	3.17	0.91	9.5	5
7.1	0.71	0	1.9	0.08	14	35	0.9972	3.47	0.55	9.4	5
7.8	0.645	0	2	0.082	8	16	0.9964	3.38	0.59	9.8	6

Fig 4.1 CSV File of Wine

4.3 OBJECTIVE OF CASE STUDY

The main objective of the red wine quality dataset is to predict which of the physiochemical features make good wine. With 11 variables and 1 output variable (quality) given, this model examines the dataset and predicts the quality of wine based on the 12 attributes of dataset. Machine learning algorithms such as decision tree, random forest and k- nearest neighbor are used to predict the quality. Each algorithm is used individually to train and test the data. Results of the algorithms are compared and the one which gives the highest accuracy is considered best to predict the quality of red wine.

CHAPTER-5

DESIGN

5.1 ARCHITECTURE

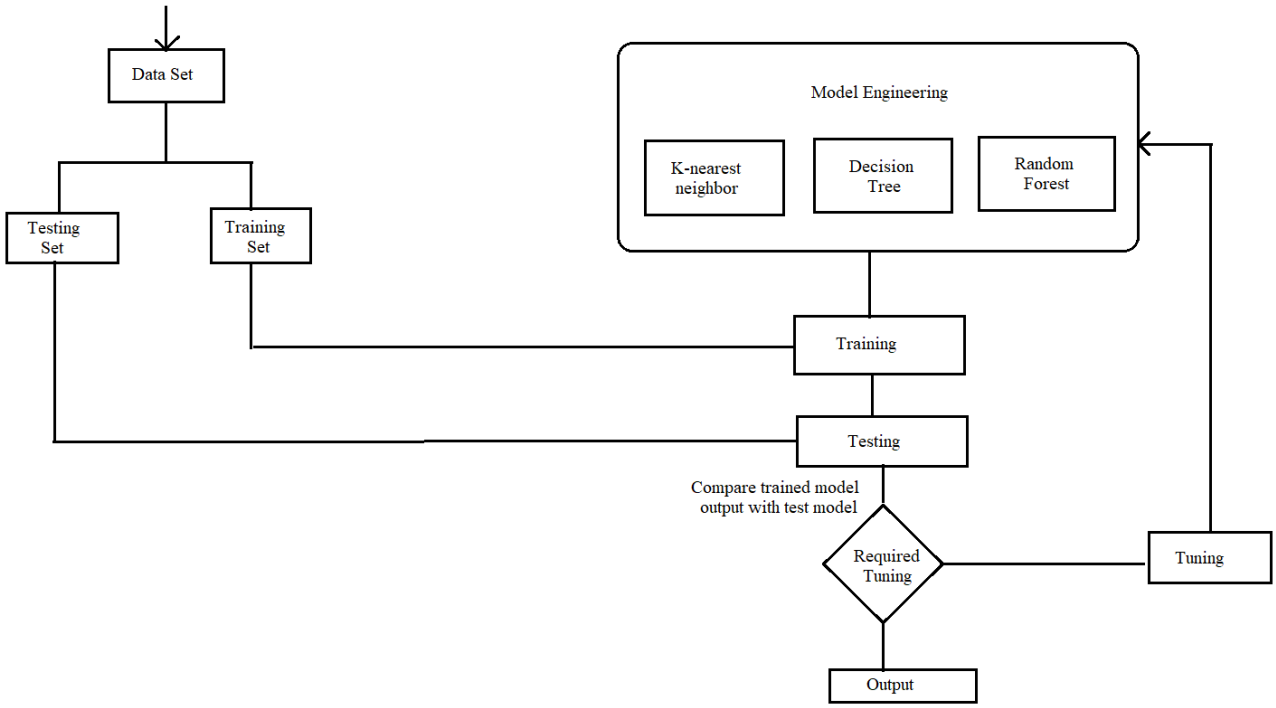

Fig 5.1 Architecture Diagram

5.2 PROCESS FLOW

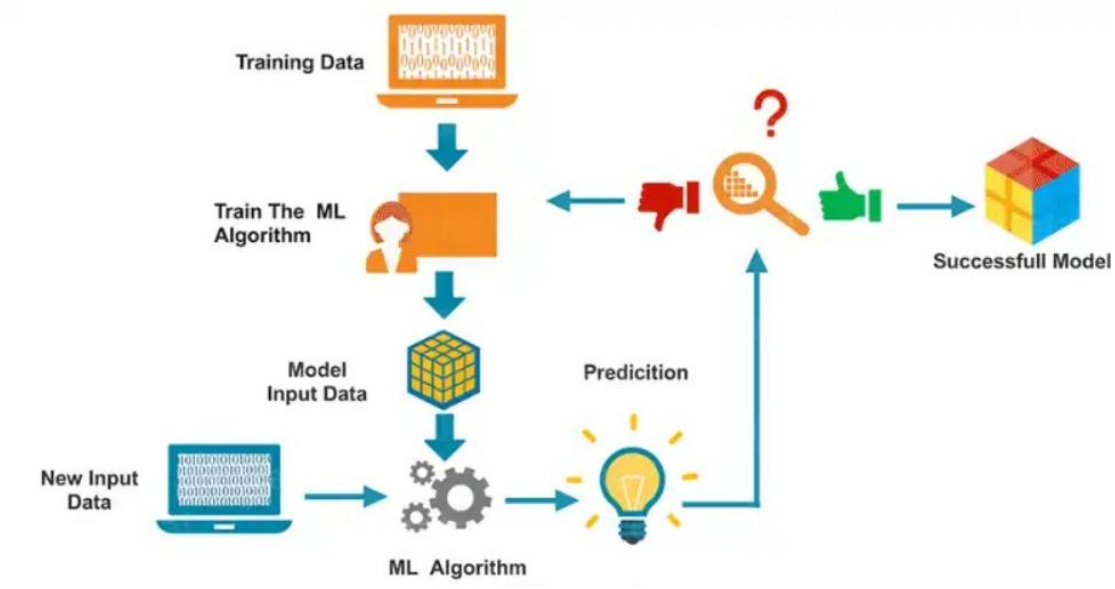

Fig 5.2 Process Flow Diagram

5.3 DATA FLOW

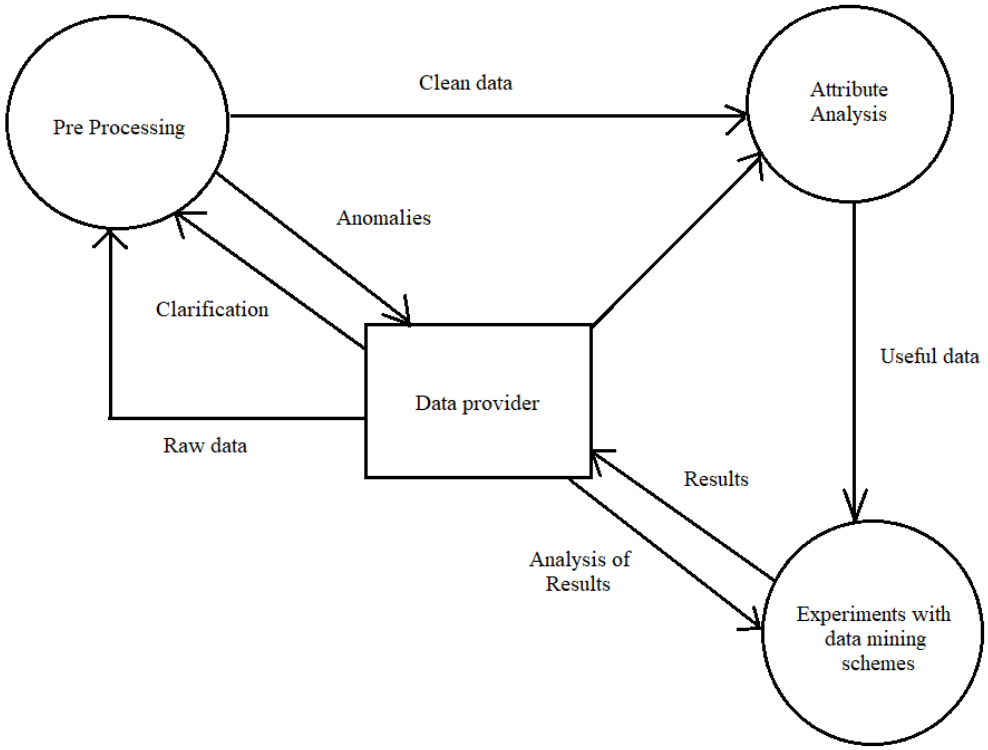

Fig 5.3 Data Flow Diagram

5.4 FLOW CHART

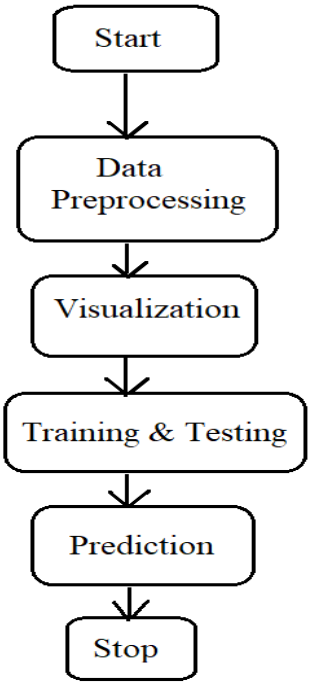

Fig 5.4 Flow Chart

CHAPTER-6

IMPLEMENTATION

6.1 IMPORTING LIBRARIES

```
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.tree import DecisionTreeClassifier
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.metrics import classification_report,accuracy_score
```

- Numpy package contains multi-dimensional array and matrix data structures which can be used to perform mathematical operations.
- Pandas package provide fast, flexible, and expressive data structures which are used to process data frames.
- Seaborn package can be used to visualize data in the form of various effective graph and plots.
- Sklearn is the main package which is used for machine learning and contains all the necessary modules for implementing algorithms.

6.2 DATA READING

```
In [2]: wine = pd.read_csv('winequality_red.csv')
        wine.shape
Out[2]: (1599, 12)

In [3]: wine
```

The dataset used for this project has been acquired from “kaggle” website. The dataset consists of 1599 rows and 12 columns or attributes.

Out[3]:

	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

Fig 6.1 Red Wine Dataset

6.3 DATA INFORMATION

```
In [4]: wine.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
fixed acidity           1599 non-null float64
volatile acidity        1599 non-null float64
citric acid             1599 non-null float64
residual sugar          1599 non-null float64
chlorides               1599 non-null float64
free sulfur dioxide     1599 non-null float64
total sulfur dioxide    1599 non-null float64
density                 1599 non-null float64
pH                      1599 non-null float64
sulphates               1599 non-null float64
alcohol                 1599 non-null float64
quality                 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

```
In [5]: wine.head()
```

Out[5]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

6.4 NULL VALUE CHECKING

```
In [6]: wine.isnull().sum()
```

```
Out[6]: fixed acidity           0
        volatile acidity        0
        citric acid             0
        residual sugar          0
        chlorides               0
        free sulfur dioxide     0
        total sulfur dioxide    0
        density                 0
        pH                      0
        sulphates               0
        alcohol                 0
        quality                 0
        dtype: int64
```

isnull( ): The function detects missing values in the data set. It returns a Boolean values indicating if the values are NA. Missing values gets mapped to True and non-missing value gets mapped to False.

6.5 DEPENDENT VARIABLE ANALYSIS

The variable that depends on other factors that are measured is known as dependent variable. These variables are expected to change as a result of an experimental manipulation of the independent variable or variables.

```
In [7]: wine.quality.value_counts()
```

```
Out[7]: 5    681
        6    638
        7    199
        4     53
        8     18
        3     10
        Name: quality, dtype: int64
```

```
In [8]: sns.countplot(x='quality',palette=['crimson'],data=wine)
        plt.show()
```

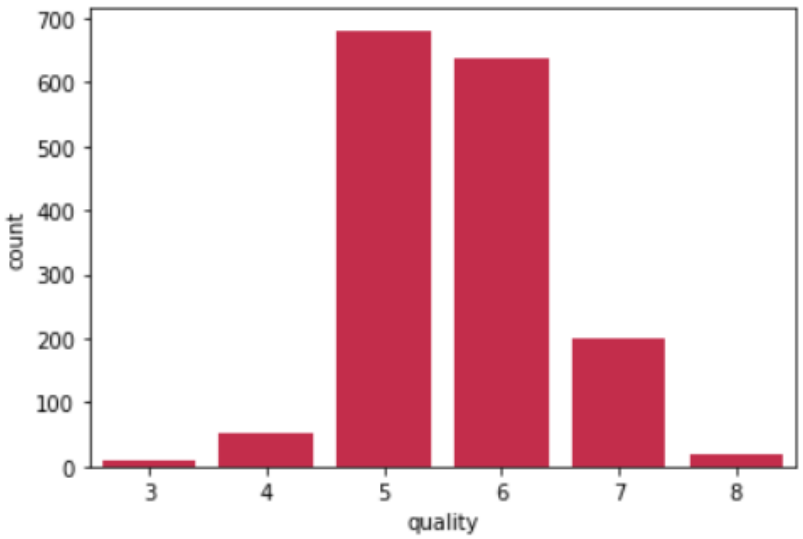

Fig 6.2 Count Plot of Quality

Countplot(): A count plot is similar to a histogram or a bar graph. This method is used to Show the count of observations in each categorical bin using bars.

6.6 REDUCTION OF DEPENDENT VARIABLE CLASSES

In [9]:
```
wine.quality =wine.quality.replace(3,6)
wine.quality =wine.quality.replace(8,5)
wine.quality =wine.quality.replace(4,5)
wine.quality =wine.quality.replace(7,6)

wine.quality.value_counts().to_frame()
```

Out[9]:

	quality
6	847
5	752

In [10]:
```
sns.countplot(x='quality',hue='quality',palette=['crimson'],data=wine)
plt.show()
```

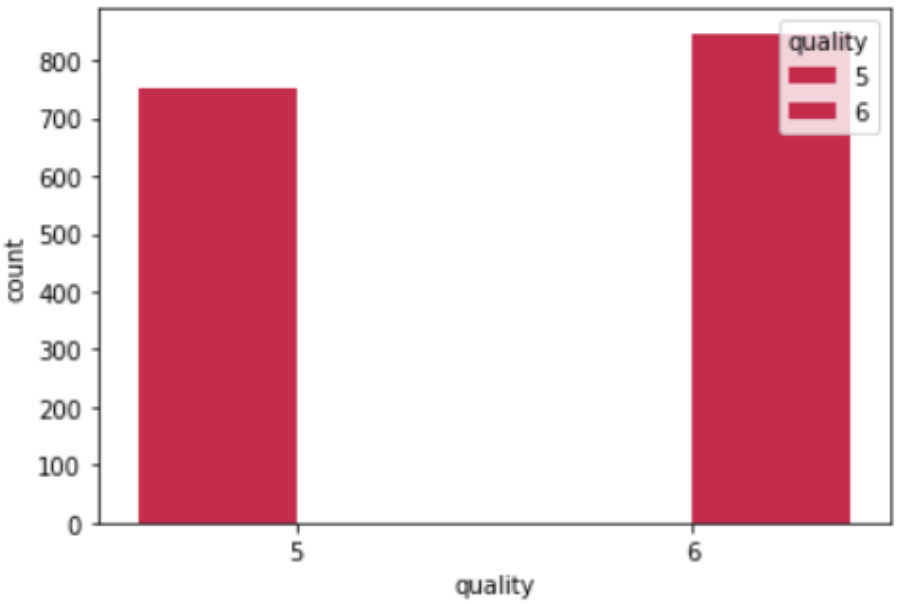

Fig 6.3 Count Plot of Quality After Reduction

6.7 UNIVARIATE ANALYSIS

Univariate analysis is the simplest form of analyzing data. Uni means one, so in other words the data has only one variable. It takes data, summarizes that data and finds patterns in the data.

Distplot( ): A dist plot plots a univariate distribution of observations.

- Figsize is used to describe the number of rows and columns required to plot the bar graph.
- Setstyle is used to define the background colour of the bar graph.

```
In [12]: plt.figure(figsize=(20,8))
         sns.set_style("dark")
         sns.distplot(wine.chlorides,kde=False, bins=int(np.sqrt(1599)),color="r")
         plt.show()
```

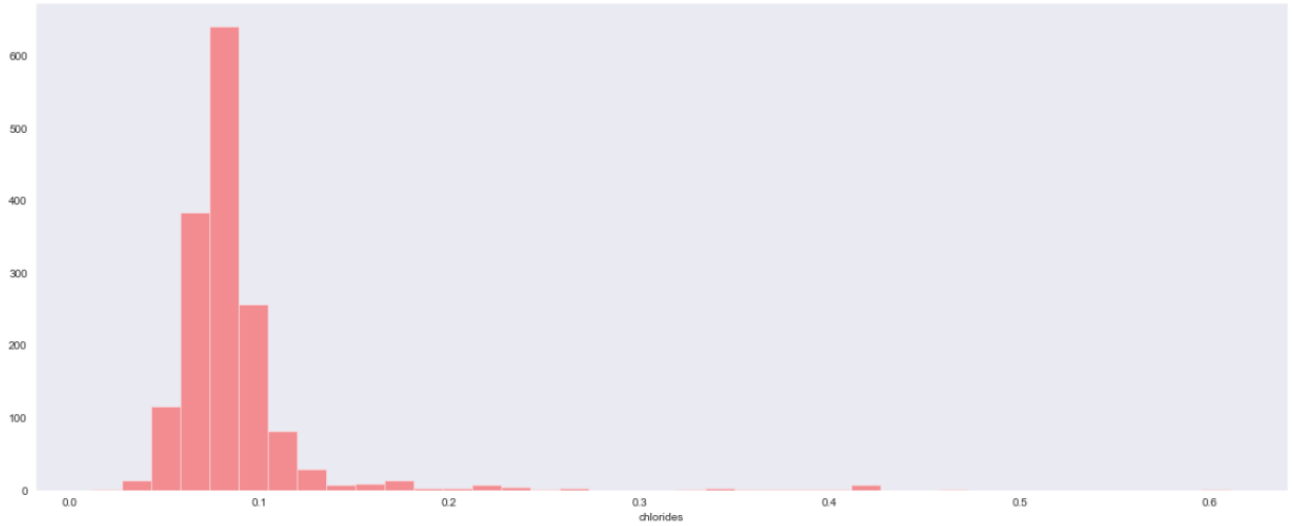

Fig 6.4 Dist Plot of Chlorides

In [13]:
```
plt.figure(figsize=(20,8))
sns.set_style("dark")
sns.distplot(wine.density,kde=False, bins=int(np.sqrt(1599)),color="b")
plt.show()
```

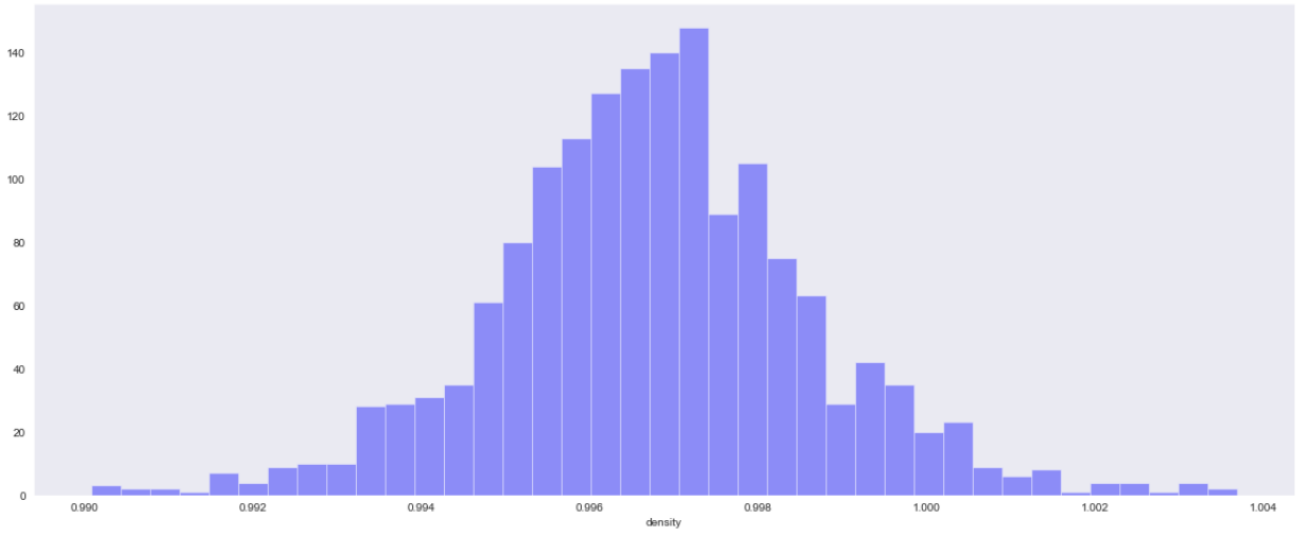

Fig 6.5 Dist Plot of Density

In [14]:
```
plt.figure(figsize=(20,8))
sns.set_style("dark")
sns.distplot(wine.pH,kde=False, bins=int(np.sqrt(1599)),color="g")
plt.show()
```

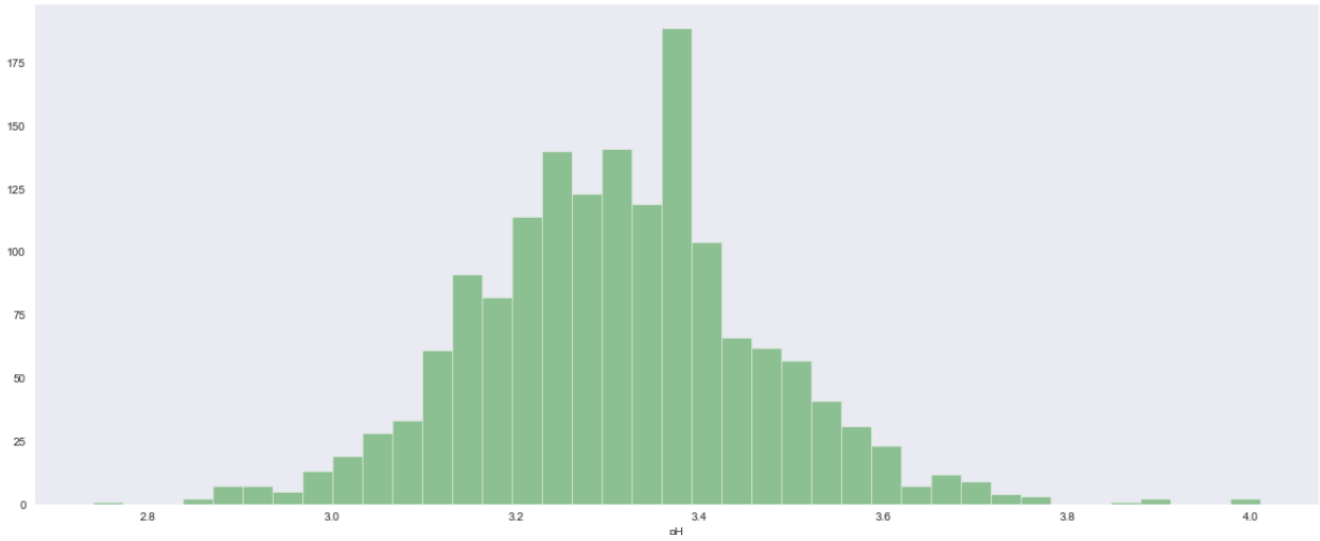

Fig 6.6 Dist Plot of Ph Value

```
In [15]: plt.figure(figsize=(20,8))
         sns.set_style("dark")
         sns.distplot(wine.sulphates,kde=False, bins=int(np.sqrt(1599)),color="y")
         plt.show()
```

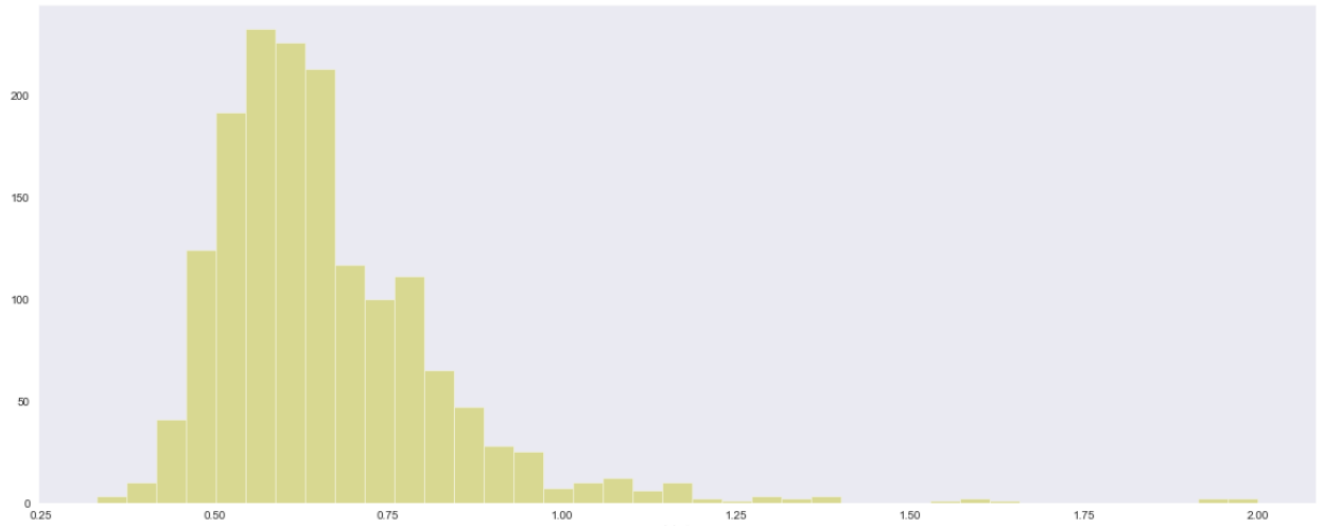

Fig 6.7 Dist Plot Sulphates

6.8 MULTIVARIATE ANALYSIS

Multivariate data analysis is a set of statistical models that examine patterns in multidimensional data by considering several data variables at once. It is an expansion of bivariate data analysis which considers only two variables at a time in its models**.**

```
In [16]: for i in ['volatile acidity','citric acid','residual sugar']:
             plt.figure(figsize=(20,8))
             sns.set_style("dark")
             sns.scatterplot(data=wine,y = i, x='alcohol',hue ='quality',palette=["yellow","cyan"])
             plt.show()
```

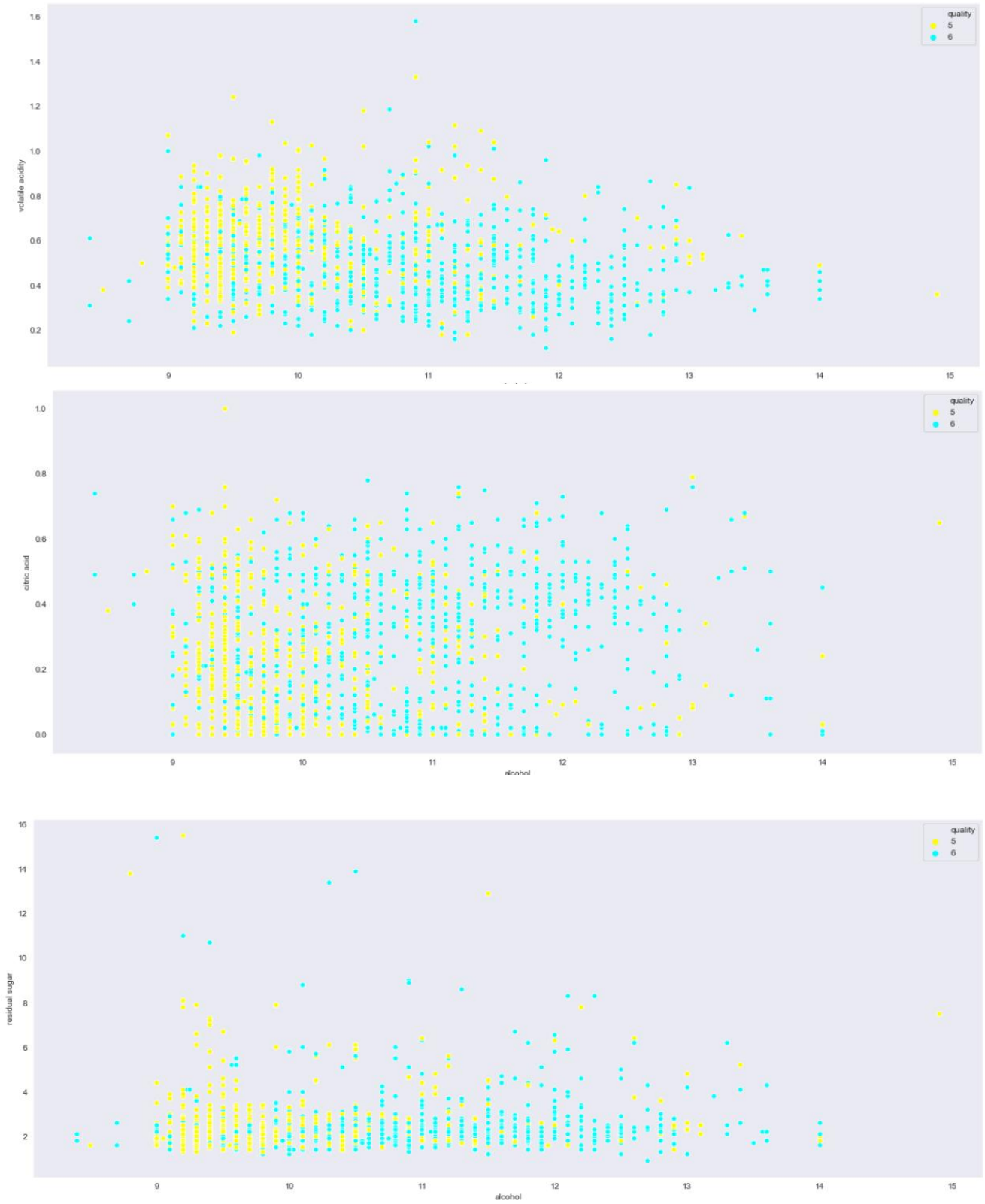

Fig 6.8 Scatter Plot Between Alcohols, Volatile Acid, Citric Acid, Residual Sugar

```
In [17]: for i in ["chlorides","free sulfur dioxide","fixed acidity"]:
             plt.figure(figsize=(20,8))
             sns.set_style("dark")
             sns.scatterplot(data=wine,y = i, x='alcohol',hue ='quality',palette=["orange","cyan"])
             plt.show()
```

Fig 6.9 Scatter Plot Between Alcohols, Chlorides, Free So_2, Fixed Acidity

```
In [18]: for i in ["density","pH","sulphates"]:
             plt.figure(figsize=(20,8))
             sns.set_style("dark")
             sns.scatterplot(data=wine,y = i, x='alcohol',hue ='quality',palette=["green","purple"])
             plt.show()
```

Fig 6.10 Scatter Plot Between Alcohols, Density, Ph, Sulphates

6.9 NUMERICAL DATA DESCRIPTION

```
In [19]: wine.describe().transpose()
```

Out[19]:

	count	mean	std	min	25%	50%	75%	max
fixed acidity	1599.0	8.319637	1.741096	4.60000	7.1000	7.90000	9.200000	15.90000
volatile acidity	1599.0	0.527821	0.179060	0.12000	0.3900	0.52000	0.640000	1.58000
citric acid	1599.0	0.270976	0.194801	0.00000	0.0900	0.26000	0.420000	1.00000
residual sugar	1599.0	2.538806	1.409928	0.90000	1.9000	2.20000	2.600000	15.50000
chlorides	1599.0	0.087467	0.047065	0.01200	0.0700	0.07900	0.090000	0.61100
free sulfur dioxide	1599.0	15.874922	10.460157	1.00000	7.0000	14.00000	21.000000	72.00000
total sulfur dioxide	1599.0	46.467792	32.895324	6.00000	22.0000	38.00000	62.000000	289.00000
density	1599.0	0.996747	0.001887	0.99007	0.9956	0.99675	0.997835	1.00369
pH	1599.0	3.311113	0.154386	2.74000	3.2100	3.31000	3.400000	4.01000
sulphates	1599.0	0.658149	0.169507	0.33000	0.5500	0.62000	0.730000	2.00000
alcohol	1599.0	10.422983	1.065668	8.40000	9.5000	10.20000	11.100000	14.90000

Fig 6.11 Numerical Data Description

6.10 DATA SPLITTING

In machine learning in order to access the performance of the classifier one need to train the classifier using training set and then test the performance of the classifier on unseen testing set. The whole dataset is generally divided into two parts i.e. training set and testing set. In this project the data set has been divided in the ratio of 7:3. 70% of the data set is split into training set and remaining 30% of the data set is used for testing.

```
In [20]: X = wine.drop(['quality'],axis=1)
         Y = wine['quality'].values
         X_train,X_test,y_train,y_test = train_test_split(X,Y, test_size=0.3, random_state=42)
```

```
In [21]: from sklearn.preprocessing import MinMaxScaler

         scaler = MinMaxScaler(feature_range=(0,1))
         rescaledX_train = scaler.fit(X)
         rescaledX_test = scaler.fit(X)
```

6.11 DECISION TREE CLASSIFIER

In [22]:
```
dt_model = DecisionTreeClassifier(criterion='gini',max_depth=10,
                                  min_samples_leaf=3, splitter='best',class_weight='balanced')
dt_model.fit(X_train,y_train)


print("Trainig accuracy",dt_model.score(X_train,y_train))
print()
print("Testing accuracy",dt_model.score(X_test, y_test))
print()
```

```
Trainig accuracy 0.8999106344950849

Testing accuracy 0.6895833333333333
```

Decision tree algorithm belongs to the family of supervised learning algorithms. This algorithm can be used for solving regression and classification problems. The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior training data.

A decision tree is a flowchart-like tree structure where an internal node represents a feature or attribute, the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions tree in recursive manner called recursive partitioning. It's visualization is like a flowchart diagram which easily mimics the human level thinking. Hence decision trees are always considered easy to understand and interpret.

In Decision Trees, in order to predict a class label for a record one starts from the root of the tree. The values of the root attribute with the record's attribute are compared. On the basis of comparison, it follows the branch corresponding to that value and jump to the next node.

6.12 K - NEAREST NEIGHBORS

```
In [24]: knn = KNeighborsClassifier(n_neighbors=9, weights='uniform', algorithm='auto', leaf_size=30, p=2)
         knn.fit(X_train,y_train)
         print("Trainig accuracy",knn.score(X_train,y_train))
         print()
         print("Testing accuracy",knn.score(X_test, y_test))
         print()

         Trainig accuracy 0.7202859696157283

         Testing accuracy 0.6229166666666667
```

K-nearest neighbors algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression problems. However, it is mainly used for classification predictive problems.

- **Lazy learning algorithm**: KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm**: KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

K-nearest neighbor algorithm uses feature similarity to predict the values of new data points. The new data point will be assigned a value based on how closely it matches the points in the training set.

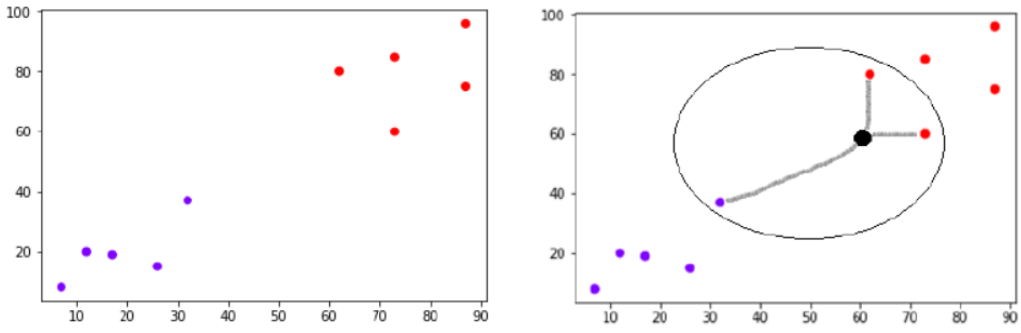

Fig 6.12 K-Nearest Neighbor When K Equals 3

Example: If the blue dots are considered as class A and red dots are considered as class B then when the k values is 3, the algorithm will compare the black dot with the three nearest dots to it. Based on the majority the new black dot is now sorted into class B.

6.13 RANDOM FOREST

In [23]:
```
rf_model = RandomForestClassifier(criterion='entropy',random_state=42,max_depth=7,max_features=None,
                                  min_samples_leaf=4,min_samples_split=6,
                                  n_estimators=1000,oob_score=True,class_weight='balanced')
rf_model.fit(X_train,y_train)
y_pred_rf = dt_model.predict(X_test)
print("Trainig accuracy",rf_model.score(X_train,y_train))
print()
print("Testing accuracy",rf_model.score(X_test, y_test))
print()
```

```
Trainig accuracy 0.871313672922252

Testing accuracy 0.75
```

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging.

The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. It is a type of learning where you join different types of algorithms or same algorithm can be used multiple times to form a more powerful prediction model.

Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

CHAPTER 7

RESULT ANALYSIS

1. Precision: It is defined as the fraction of relevant instances among the retrieved instances.

2. Recall: It is defined as the fraction of the total amount of relevant instances that were actually retrieved.

3. F1 Score: It is the weighted average of precision and recall. This score takes both false positives and false negatives into account.

4. Support: It is defined as the number of actual occurrences of the class in the specified dataset.

7.1 DECISION TREE

	precision	recall	f1-score	support
5	0.87	0.93	0.90	534
6	0.93	0.88	0.90	585
accuracy			0.90	1119
macro avg	0.90	0.90	0.90	1119
weighted avg	0.90	0.90	0.90	1119

Fig 7.1 Classification Report Of Decision Tree Training Set

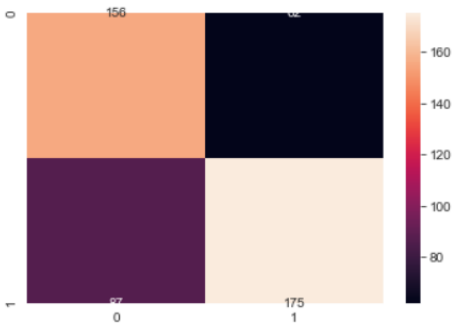

Fig 7.2 Heat Map Of Decision Tree Confusion Matrix

	precision	recall	f1-score	support
5	0.64	0.72	0.68	218
6	0.74	0.67	0.70	262
accuracy			0.69	480
macro avg	0.69	0.69	0.69	480
weighted avg	0.69	0.69	0.69	480

Fig 7.3 Classification Report Of Decision Tree Testing Set

```
Trainig accuracy 0.8999106344950849

Testing accuracy 0.6895833333333333
```

Fig 7.4 Decision Tree Accuracy

7.2 RANDOM FOREST

	precision	recall	f1-score	support
5	0.85	0.89	0.87	534
6	0.90	0.85	0.87	585
accuracy			0.87	1119
macro avg	0.87	0.87	0.87	1119
weighted avg	0.87	0.87	0.87	1119

Fig 7.5 Classification Report Of Random Forest Training Set

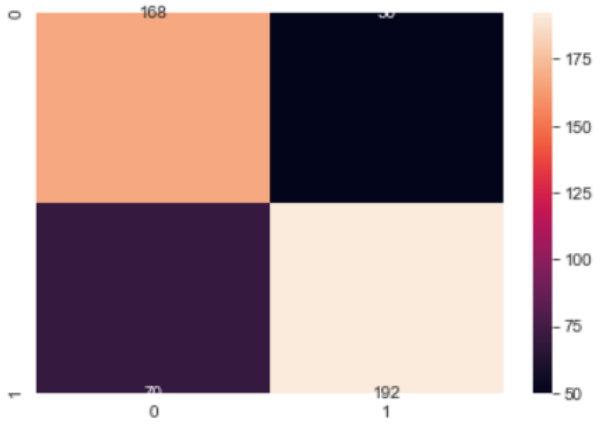

Fig 7.6 Heat Map Of Random Forest Confusion Matrix

	precision	recall	f1-score	support
5	0.71	0.77	0.74	218
6	0.79	0.73	0.76	262
accuracy			0.75	480
macro avg	0.75	0.75	0.75	480
weighted avg	0.75	0.75	0.75	480

Fig 7.7 Classification Report Of Random Forest Testing Set

```
Trainig accuracy 0.7202859696157283

Testing accuracy 0.6229166666666667
```

Fig 7.8 Random Forest Accuracy

7.3 K-NEAREST NEIGHBOR

	precision	recall	f1-score	support
5	0.70	0.72	0.71	534
6	0.74	0.72	0.73	585
accuracy			0.72	1119
macro avg	0.72	0.72	0.72	1119
weighted avg	0.72	0.72	0.72	1119

Fig 7.9 Classification Report Of Knn Training Set

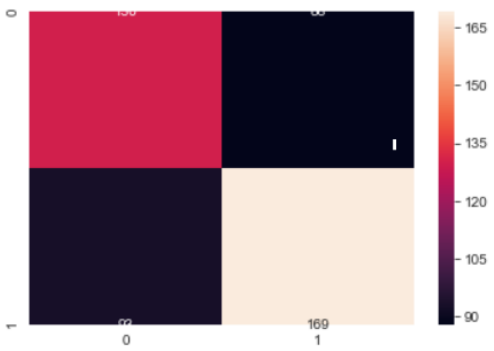

Fig 7.10 Heat Map Of Knn Confusion Matrix

	precision	recall	f1-score	support
5	0.58	0.60	0.59	218
6	0.66	0.65	0.65	262
accuracy			0.62	480
macro avg	0.62	0.62	0.62	480
weighted avg	0.62	0.62	0.62	480

Fig 7.11 Classification Report Of Knn Testing Set

```
Trainig accuracy 0.871313672922252

Testing accuracy 0.75
```

Fig 7.12 Knn Accuracy

CHAPTER 8

CONCLUSION

Firstly, the data set used for this project has been acquired from “kaggle” website. Secondly, several packages like numpy, pandas, seaborn, matplotlib have been used in order to perform the statistical analysis, visualization and data splitting. Thirdly, the data set has been split in the ratio of 7:3 i.e training and testing set using which three machine learning algorithms Decision Tree, Random Forest and KNN have been trained and tested. Based on the results it is concluded that Random Forest is the most suitable algorithm for predicting the quality of red wine with an accuracy of “0.75”.

The data set acquired from “kaggle” website consists of 1599 rows and 12 columns. If the data set consists of more number of rows then the machine learning models can be trained more precisely which would give even better results. Since the model would be trained with bigger data set i.e more data it would peform better with testing data and would give higher accuracy. The model would be more precise in predicting the quality of red wine.

REFERENCES

1. https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009
2. https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm
3. https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm
4. https://www.geeksforgeeks.org/decision-tree/
5. https://searchenterpriseai.techtarget.com/definition/machine-learning-ML#:~:text=Machine%20learning%20(ML)%20is%20a,to%20predict%20new%20output%20values.