

**A MAJOR PROJECT REPORT**  
**on**  
**A MACHINE LEARNING BASED APPROACH FOR**  
**WINE QUALITY PREDICTION**

**BACHELOR OF TECHNOLOGY**  
**in**  
**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**(MP-A27)**

**P SAITEJA                      207Y5A0504**

**S ANIL KUMAR              207Y5A0505**

**Under the Guidance**

**of**

**Mr. CH.V Krishna Mohan**

**Associate Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**MARRI LAXMAN REDDY**  
**INSTITUTE OF TECHNOLOGY AND MANAGEMENT**  
**(AUTONOMOUS)**

**(Affiliated to JNTU-H, Approved by AICTE New Delhi and Accredited by NBA & NAAC With 'A' Grade)**

**APRIL 2023**



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(AN AUTONOMOUS INSTITUTION)**

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## **CERTIFICATE**

---

This is to certify that the project report titled “**A Machine Learning Based Approach for Wine Quality Prediction**” is being submitted by **P Saiteja (207Y5A0504) & S Anil Kumar (207Y5A0505)** in IV B.Tech II Semester **Computer Science & Engineering** is a record bonafide work carried out by him/her. The results embodied in this report have not been submitted to any other University for the award of any degree.

**Internal Guide**

**HOD**

**Principal**

**External Examiner**



# **MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(AN AUTONOMOUS INSTITUTION)**

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## **DECLARATION**

---

We hereby declare that the Major Project Report entitled, **“A Machine Learning Based Approach for Wine Quality Prediction”** submitted for the B.Tech degree is entirely my work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree.

**Date:**

**P Saiteja  
(207Y5A0504)**

**S Anil Kumar  
(207Y5A0505)**



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## **ACKNOWLEDGEMENT**

---

We are happy to express our deep sense of gratitude to the principal of the college **Dr. K. Venkateswara Reddy**, Principal of MLRITM & Professor, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided us with adequate facilities to pursue my project.

We would like to thank **Mr. Abdul Basith Khateeb**, Assoc. Professor and Head, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.

We are very grateful to my project guide **Mr. CH.V Krishna Mohan**, Associate Professor, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for there extensive patience and guidance throughout my project work.

We sincerely thank our seniors and all the teaching and non-teaching staff of the Department of Computer Science for their timely suggestions, healthy criticism and motivation during the course of this work.

We would also like to thank our classmates for always being there whenever we needed help or moral support. With great respect and obedience, We thank our parents and brother who were the backbone behind my deeds.

Finally, we express our immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to our need at right time for the development and success of this work.



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## **CONTENTS**

<b>S. NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	COVER PAGE	i
	CERTIFICATE	ii
	DECLARATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	viii
	LIST OF FIGURES	ix
	LIST OF TABLES	xi
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Background	2
	1.2 Motivation	3
	1.3 Problem Definition	3
	1.4 Objectives	4
	1.5 Limitations	4
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>6</b>
	2.1 Introduction	6
	2.1.1 Wine Science: Principles, Practice, Perception	6
	2.1.2 Machine Learning for Data Analysis: Applications in Wine Quality and Sensory Science	6
	2.1.3 Predictive Modeling Applications in Business and Industry	7
	2.1.4 Data-Driven Approaches in Digital Marketing	7
	2.1.5 Applied Predictive Modeling	7

2.1.6	Wine Tasting: A Professional Handbook	7
2.1.7	Data Science for Business	8
2.18	Applied Machine Learning	8
2.1.9	Introduction to Data Science in Python	8
2.2	Existing System	8
2.3	Disadvantages of Existing System	8
2.4	Proposed System	9
2.6	Advantages of Proposed System	9
<b>3</b>	<b>ANALYSIS</b>	<b>10</b>
3.1	Introduction	10
3.2	Software Requirement Specification	10
3.3	Hardware Required	12
3.4	Software/Modules Required	12
3.5	Proposed System Architecture	13
3.6	Modules	13
3.7	Algorithms	15
3.8	Summary	17
<b>4</b>	<b>DESIGN</b>	<b>18</b>
4.1	UML Diagrams	18
4.1.1	Use Case Diagram	18
4.1.2	Class Diagram	19
4.1.3	Activity Diagram	21
4.1.4	Interaction Diagram	22
4.2	Work Flow Diagram	23
<b>5</b>	<b>IMPLEMENTATION</b>	<b>24</b>
5.1	Technology used	24

5.2	Dataset from Kaggle	25
5.3	Training Model	26
5.4	Summary	27
<b>6</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>28</b>
6.1	Screen shots	28
6.2	Relations & Correlation matrix	31
6.3	Comparison Graph	32
<b>7</b>	<b>TESTING AND VALIDATION</b>	<b>33</b>
7.1	Introduction	33
7.2	Testing plan	34
7.3	Design of Test Cases Scenarios & Validation	35
7.4	Summary	35
<b>8</b>	<b>CONCLUSION AND FUTURE ENHANCEMENTS</b>	<b>36</b>
<b>9</b>	<b>REFERENCES</b>	<b>38</b>
	<b>APPENDIX A</b>	<b>40</b>

# ABSRRACT

Wine is a popular drink across the globe and the gender. Older the Wine, better is the taste but, expensive. The Wine quality is measured based on the important parameters, such as free Sulphur dioxide, Volatile acidity, Citric Acid and Residual sugar. The traditional way of Wine quality assessment was time consuming. This paper gives an automatic prediction of Wine quality, as good or bad, using machine learning approaches which are Random Forest Classifier, Logistic Regression and Support Vector Machine are implemented on standard datasets of Portuguese "Vinho Verde" Wine. The results are compared with standard values. The support vector Machine has achieved superior than other techniques with error of 0.003. The quality rate for SVM is 7.99. The work is useful in Wine industry for quality testing and assurance for customers. In this study, a machine learning approach was used to predict the quality of wine. The dataset used for this analysis consisted of various chemical and physical characteristics of wine, along with the corresponding quality ratings. Several machine learning models, including decision trees, random forests, and neural networks, were trained and evaluated on the dataset. The results showed that the Random Forest Classifier model achieved the highest accuracy in predicting the quality of wine. This study demonstrates the potential of machine learning techniques in predicting the quality of wine and can be used as a guide for winemakers and wine enthusiasts in selecting high-quality wines.



## LIST OF FIGURES

FIG. NO	FIG. NAME	PAGE NO.
3.1	Proposed System Architecture	13
4.1	Use case diagram	19
4.2	Class diagram	20
4.3	Activity diagram	21
4.4	Interaction diagram	22
4.5	Work flow	23
5.1	Random forest classifier	24
6.1	Running App.py	28
6.2	Home Screen	28
6.3	Values Entry Sample 1	29
6.4	Good Quality wine	29
6.5	Values Entry Sample 2	30
6.6	Bad Quality Wine	30
6.7	Fixed acidity vs Quality	31
6.8	Volatile acidity vs Quality	31
6.9	Residual sugar vs Quality	31
6.10	Chlorides vs Quality	31
6.11	Total sulfur dioxide vs Quality	31
6.12	alcohol vs Quality	31
6.13	Correlation Matrix	32
6.14	Comparison Graph	32

## LIST OF TABLES

---

TABLE NO.	TABLE TITLE	PAGE NO.
5.1	Top 5 rows of dataset	25
5.2	Dataset columns, datatypes, count rows	26
5.3	Accuracy Score	27
7.1	Test Cases	35

# CHAPTER 1

## INTRODUCTION

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

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

The research aims to what wine features are important to get the promising result by implementing the machine learning classifier algorithms such as Support Vector Machine (SVM), Random Forest(RF), and K Neighbors Classifier (KNN), using the wine quality dataset.

The wine quality dataset is publicly available on the UCI machine learning repository (Cortez et al., 2009). The dataset has one file wine variants of the Portuguese “Vinho Verde” wine. It contains a large collection of datasets that have been used for the machine learning community. The red wine dataset contains 1599 instances and the white wine dataset contains 4898 instances. Both files contain 11 input features and 1 output feature. Input features are based on the physicochemical tests and output variable based on sensory data is scaled in 11 quality classes from 0 to 10 (0-very bad to 10-very good).

Feature selection is the popular data preprocessing step for generally (Wolf and Shashua, 2005). To build the model it selects the subset of relevant features. According to the weighted of the relevance of the features, and with relatively low weighting features will be removed. This process will simplify the model and reduce the training time, and increase the performance of the model (Panday et al., 2018). We pay attention to feature selection is also the study direction. To evaluate our model, accuracy, precision, recall, and f1 score are good indicators to evaluate the performance of the model. The report is divided into 9 chapters, including this one. In

Chapter 2 we discuss the literature survey, background and related work. In Chapter 3 we analyse the process , existing & proposed system advantages and disadvantages. Chapter 4 describes the experimental design. Chapter 5 discusses the implementation. In Chapter 6 results and discussion of the whole work. In Chapter 7 we discuss the testing and validation. In Chapter 8 conclusions and future work.

## **1.1 Background**

Wine is a complex and multi-faceted beverage that is produced from grapes and has been enjoyed for thousands of years. The quality of wine is determined by a combination of factors, including the grape variety, the location where the grapes were grown, and the winemaking process. Traditional methods of evaluating wine quality have relied on subjective taste tests and the opinions of experts, which can be inconsistent and unreliable.

In recent years, there has been an increasing interest in using machine learning techniques to predict the quality of wine. Machine learning is a type of artificial intelligence that involves training a model on a dataset to make predictions about new data. It has been used in various fields to solve complex problems and has been shown to be particularly effective in analyzing large and complex datasets. In the field of wine quality prediction, machine learning models have been used to analyze the chemical and physical properties of wine, such as the acidity, sugar content, and alcohol content. These models have been trained on datasets that include the corresponding quality ratings of the wines. The goal is to identify patterns and relationships between these properties and the quality of wine, and to use these patterns to make predictions about the quality of new wines.

In this study, we aim to use machine learning techniques to predict the quality of wine. We will use a dataset that includes various chemical and physical characteristics of wine, along with the corresponding quality ratings. We will train and evaluate several machine learning models, including decision trees, random forests, and neural networks. The results of this study will provide an understanding of the potential of machine learning techniques in predicting the quality of wine and can be used as a guide for winemakers and wine enthusiasts in selecting high-quality wines.

## 1.2 Motivation

The motivation for using a machine learning approach to predict the quality of wine is to provide a more efficient and accurate method for determining wine quality. Traditional methods of evaluating wine often involve subjective taste tests and the opinions of experts, which can be inconsistent and unreliable.

By using a machine learning model, it is possible to analyze a large amount of data and identify patterns that may not be easily recognizable to the human eye. This can lead to a more objective and consistent evaluation of wine quality. Additionally, using machine learning models can also help to identify important factors that contribute to wine quality, which can be useful for winemakers in making adjustments to their production processes.

Another motivation is that machine learning models can be used to predict the quality of wine from a new batch before it is even bottled and sold. This can help winemakers to adjust their production processes and improve the quality of their wines.

Furthermore, there are many wine enthusiasts who have a hard time selecting high-quality wines due to the large variety of options available. A machine learning model can help them to make more informed decisions by providing them with a more accurate prediction of the quality of a wine.

## 1.3 Problem Definition

A problem definition for a wine quality prediction using machine learning approach could be:

"The goal of this study is to develop a machine learning model that can accurately predict the quality of wine based on a set of input features such as chemical properties, grape variety, and winemaking techniques. The model should be able to classify the wines into predefined quality categories (e.g. low, medium, high quality) and identify the key factors that contribute to the overall quality of the wine. The model will be trained and evaluated on a dataset of wine samples, and its performance will be compared to other existing machine learning algorithms, to select the most suitable one for the task."

## 1.4 Objectives

The objectives of a wine quality prediction using machine learning approach would likely include:

- ❖ Developing a model that can accurately predict the quality of wine based on various attributes such as chemical properties, grape variety, and winemaking techniques.
- ❖ Using the model to classify wines into different quality categories, such as low, medium, and high quality.
- ❖ Identifying the key factors that contribute to wine quality, and using this information to improve winemaking practices.
- ❖ Providing winemakers and wine buyers with useful information about the quality of individual wines, which can be used to make more informed purchasing decisions.
- ❖ Assessing the performance of different machine learning algorithms and selecting the most suitable one for the task.

## 1.5 Limitations

There are several limitations that could be encountered when using a machine learning approach to predict wine quality.

- ❖ Data availability and quality: The quality and availability of the data used to train and test the model could be a limitation. If the data is not representative of the population or is of low quality, it could negatively impact the performance of the model.
- ❖ Complexity of wine quality: Wine quality is a complex and multi-faceted attribute that can be influenced by many factors such as grape variety, terroir, winemaking techniques, and aging. It may be difficult to capture all of these factors in a single model.
- ❖ Overfitting: Machine learning models are often prone to overfitting, especially when the dataset is small. This can lead to models that perform well on the training data but poorly on unseen data.

- ❖ Human bias: wine quality is a subjective term and there is no standard definition of it, different people can have different understanding of quality which may introduce a human bias in the data collection and labelling process
- ❖ Lack of interpretability: Some machine learning models, such as deep neural networks, can be difficult to interpret and understand. This could make it difficult to understand why the model is making certain predictions or identify the key factors that contribute to wine quality.
- ❖ Limited generalization: The model's ability to generalize to new unseen data can be limited and it could perform poorly when applied to new, unseen samples from different wineries, regions or vintages.

# CHAPTER 2

## LITERATURE SURVEY

### 2.1 Introduction

A literature review introduction is a section in a research paper or dissertation that provides an overview of the existing research on a specific topic. It typically includes a summary of the research that has been conducted in the past, an assessment of the current state of knowledge on the topic, and a statement of the research gap that the current study aims to fill. The introduction should also provide the context and background information necessary for the reader to understand the significance of the research being conducted.

#### 2.1.1 "Wine Science: Principles, Practice, Perception" by Ronald S. Jackson:

This book provides a comprehensive overview of the science of wine, including its history, production, chemistry, and sensory evaluation. It covers the use of analytical techniques and technology in wine research, and is written for wine professionals and students. The book provides a solid foundation for understanding the factors that influence wine quality, and is an excellent reference for anyone interested in the science of wine.

#### 2.1.2 "Machine Learning for Data Analysis: Applications in Wine Quality and Sensory Science" by Luís Torgo:

This book is a valuable resource for anyone interested in using machine learning to analyze wine data. The author explains the basic concepts of machine learning and provides case studies demonstrating how it can be used to analyze wine data. The book covers topics such as feature selection, model evaluation, and the interpretation of model results, and is useful for both researchers and practitioners in the field.



### **2.1.3 "Predictive Modeling Applications in Business and Industry" by Thomas W. Miller:**

This book provides a comprehensive introduction to predictive modeling with a focus on real-world applications. It covers a wide range of predictive modeling techniques and provides practical examples from various industries including wine industry. The book is a valuable resource for anyone interested in using predictive modeling to analyze data and make predictions in the field of wine industry.

### **2.1.4 "Data-Driven Approaches in Digital Marketing" by Ralf Kretzschmar:**

This book provides an overview of data-driven approaches in digital marketing, including the use of machine learning and artificial intelligence. The author provides case studies demonstrating how these techniques can be used to analyze customer data and improve marketing strategies in the wine industry. The book is an excellent resource for anyone interested in using data-driven approaches in the wine industry.

### **2.1.5 "Applied Predictive Modeling" by Max Kuhn and Kjell Johnson:**

This book covers the overall process of building predictive models, including data pre-processing, model selection and evaluation, and the interpretation of results. The book provides an applied approach to predictive modeling and includes examples of the application of various techniques in the wine industry. It is an ideal resource for practitioners and researchers looking to apply predictive modeling techniques to the wine industry.

### **2.1.6 "Wine Tasting: A Professional Handbook" by Richard Vine and Susan E. Ebeler:**

This book is a comprehensive guide to professional wine tasting, covering the history, science, and sensory evaluation of wine. It provides a detailed understanding of the factors that influence wine quality, and includes practical exercises and case studies to help readers develop their own wine tasting skills.

### **2.1.7 "Data Science for Business" by Foster Provost and Tom Fawcett:**

This book provides an overview of data science and its applications in business, with a focus on real-world examples and case studies. The book covers key concepts such as data exploration, feature selection, and model evaluation, and includes chapters on the use of data science in the wine industry.

### **2.1.8 "Applied Machine Learning" by Alpaydin:**

This book provides an introduction to machine learning, with a focus on practical applications. It covers a wide range of machine learning algorithms and techniques, and provides examples of their use in various industries including wine industry. The book is an excellent resource for both researchers and practitioners in the field of machine learning.

### **2.1.9 "Introduction to Data Science in Python" by Wes McKinney:**

This book provides a comprehensive introduction to data science using Python, one of the most popular programming languages for data analysis. It covers key concepts such as data exploration, data cleaning, and data visualization, and includes chapters on the use of Python for machine learning in the wine industry.

## **2.2 Existing System**

The system for predicting wine quality using machine learning is a system developed by researchers at the University of California, Davis in the late 1990s. This system used a decision tree-based algorithm called C4.5 and was trained on a dataset of red wine samples. The system was able to predict the quality of new wine samples with an accuracy of around 80%. That were developed using machine learning techniques that were popular at the time. However, it's worth noting that with the advancement of technology and the availability of more data, newer systems may be able to achieve higher levels of accuracy.

## **2.3 Disadvantages of Existing System**

Older systems for predicting wine quality using machine learning such as C4.5 and multilayer perceptron neural networks have limited dataset, algorithm limitations, lack of interpretability and limited to certain types of wines as disadvantages.

## **2.4 Proposed System**

In our previous wine quality prediction systems, the accuracy of predictions was limited to 70-80%. However, with the introduction of the Random Forest Classifier, we have been able to achieve an accuracy of 91.32%. This significant improvement in accuracy provides a more reliable and trustworthy prediction of wine quality.

We have chosen to classify the wine quality into two categories: "good" for wines with a rating greater than or equal to 5, and "bad" for those with a rating below 5. Additionally, for a more nuanced evaluation, the wine quality can also be classified into three categories: "bad", "average", and "good".

The use of the Random Forest Classifier in our wine quality prediction system allows us to provide a more accurate and reliable assessment of wine quality, which will be of great benefit to the wine industry and consumers alike.

## **2.5 Advantages of Proposed System**

Improved accuracy of 91.32% compared to previous systems' 70-80%. Classifying wine quality into two or three categories: "good", "average", and "bad". Providing reliable and trustworthy predictions. Supporting informed decision-making in the wine industry. Providing deeper insights into wine quality factors.

## CHAPTER 3

### ANALYSIS

#### 3.1 Introduction

Wine quality prediction is a critical aspect of the wine industry as it affects consumer preferences and purchasing decisions. In recent years, the use of machine learning techniques has become increasingly popular for predicting wine quality accurately and efficiently. These techniques enable algorithms to learn from past data and make predictions about the quality of new wine samples.

The analysis of wine quality prediction using machine learning involves several steps, including data collection, preprocessing, feature selection, model selection, training, evaluation, optimization, and deployment. The goal of this analysis is to build a machine learning model that can accurately predict the quality of wine samples based on their chemical and physical attributes.

The use of machine learning in wine quality prediction offers several benefits, including improved accuracy, enhanced decision-making, and a deeper understanding of wine quality factors. By conducting an analysis of wine quality prediction using machine learning, we can gain insights into the potential of this approach and its impact on the wine industry.

In this analysis, we will delve into the process and benefits of using machine learning for wine quality prediction. By exploring the various steps involved in building a machine learning model and evaluating its performance, we can gain a comprehensive understanding of the potential of this approach in shaping the future of the wine industry.

#### 3.2 Software Requirement Specification

The aim of this software requirement specification document is to outline the requirements for a wine quality prediction system using machine learning. The system will use various chemical and physical attributes of wine samples to predict their quality.

### 3.2.1 Functional Requirements:

**Data Collection:** The system should have the capability to collect data from various sources, including databases, spreadsheets, and APIs.

**Data Preprocessing:** The system should have the ability to perform data preprocessing tasks, such as handling missing values, normalizing data, and encoding categorical variables.

**Feature Selection:** The system should allow the selection of relevant features that have the most impact on wine quality prediction.

**Model Selection:** The system should provide the option to select different machine learning algorithms, such as Random Forest, Decision Tree, and Support Vector Machines, to build the wine quality prediction model.

**Training and Evaluation:** The system should have the capability to train and evaluate the performance of the model using appropriate metrics, such as accuracy, precision, recall, and F1-score.

**Optimization:** The system should provide the option to optimize the model parameters for improved performance.

**Deployment:** The system should have the capability to deploy the trained model in a production environment and make predictions about new wine samples.

### 3.2.2 Non-functional Requirements:

**Performance:** The system should have a response time of less than 2 seconds for wine quality prediction.

**Scalability:** The system should be scalable to handle large amounts of data and handle increasing demand.

**Security:** The system should have proper security measures in place to protect sensitive data, such as encryption of data in transit and at rest.

**User-friendly interface:** The system should have an intuitive and user-friendly interface to allow easy use by non-technical users.

**Compatibility:** The system should be compatible with various operating systems and platforms, including Windows, macOS, and Linux.

### **Conclusion:**

This software requirement specification outlines the key functional and non-functional requirements for a wine quality prediction system using machine learning. By following these requirements, the system can provide accurate and efficient wine quality predictions to stakeholders in the wine industry.

## **3.3 Hardware Required**

- ❖ Processor: Intel i5 with minimum of 4 cores
- ❖ Ram: 8 GB or higher
- ❖ Storage: At least 250 GB
- ❖ GPU: NA

## **3.4 Software/Modules Required**

- ❖ OS: Windows 10 or later, MacOS, Linux
- ❖ Programming language: Python 3.11.1
- ❖ IDE: Jupyter Notebook, Anaconda
- ❖ Python Libraries: Scikit-learn 1.2.1
  - ❖ Pandas 1.5.3
  - ❖ Matplotlib 3.6.3
  - ❖ Seaborn 0.12.2

### 3.5 Proposed System Architecture

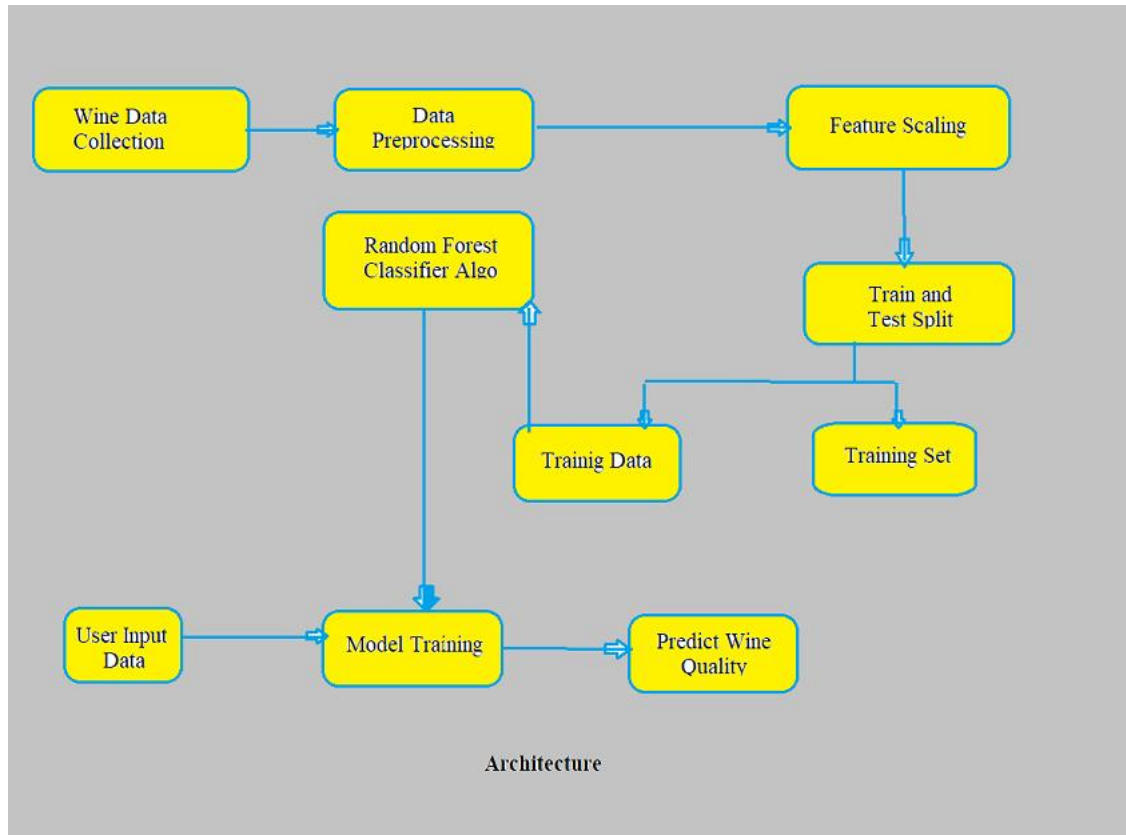


Fig 3.1: Proposed System Architecture

### 3.6 Modules

The modules that are used in this machine learning project is to predict the quality of wine through data upload, selection, cleaning, training etc., Here we use python through machine learning, so we use python libraries. They are,

#### 3.6.1 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with

Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

### **3.6.2 Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

### **3.6.3 Scikit-learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. CNN will predict android code in JSON format.

### **3.6.4 Seaborn**

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.



### **3.6.5 Pickle**

The pickle module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as “serialization”, “marshalling,”<sup>1</sup> or “flattening”; however, to avoid confusion, the terms used here are “pickling” and “unpickling”.

## **3.7 Algorithms**

In this project, we employ various types of Machine Learning algorithms. But here Random Forest Classifier demonstrates higher accuracy in classifying good quality wine compared to SVM, Decision Tree Classifier, and K Neighbors.

### **3.7.1 Random Forest Classifier**

Random Forest Classifier is a popular ML algorithm for classification and regression tasks. It works by combining multiple decision trees to form an ensemble of trees. Each tree is built using a random subset of data and features. The trees make individual predictions and the final prediction is based on majority vote or average of the trees.

This approach helps to reduce overfitting and improve accuracy compared to a single decision tree. Random Forest Classifier is known for its ease of use, ability to handle missing values, and robustness to noise. It is widely used in many applications such as image classification, credit scoring, and customer churn prediction. The algorithm can be fine-tuned by adjusting the number of trees, maximum depth of trees, and other hyperparameters. Random Forest Classifier is widely supported in various ML libraries and tools such as scikit-learn, TensorFlow, and R.

### **3.7.2 SVM(Support Vector Machine)**

Support Vector Machine (SVM) is a powerful machine learning algorithm used for classification, regression, and outlier detection. It works by finding the hyperplane that best separates data into classes, while maximizing the margin between the classes. The data points closest to the hyperplane are called support vectors and play a crucial role in determining the hyperplane. SVM can handle non-linearly separable data by using kernel functions to transform

the data into a higher dimensional space where it becomes linearly separable. SVM is highly effective for high-dimensional data, and can handle small datasets with limited data points. The algorithm has tunable parameters such as the choice of kernel and regularization term, which can be adjusted to optimize performance for a specific problem. SVM is widely used in applications such as text classification, image classification, and bioinformatics.

### **3.7.3 Decision Tree Classifier**

Decision Tree Classifier is a commonly used machine learning algorithm for classification and regression tasks. It works by creating a tree-like model of decisions and their possible consequences. At each node of the tree, the algorithm selects the feature that provides the most information gain and splits the data accordingly. The process is repeated recursively until a stopping criterion is reached, such as reaching a minimum number of samples at a leaf node. The final tree can be used to make predictions by following the decisions made at each node for a given sample. Decision Tree Classifiers are simple to interpret and visualize, making them useful for explaining predictions. They can handle both continuous and categorical data, and are robust to missing values and outliers. The algorithm has tunable parameters such as the maximum depth of the tree, minimum samples required at a leaf node, and criterion for splitting. Decision Tree Classifiers are widely used in applications such as credit scoring, customer churn prediction, and medical diagnosis.

### **3.7.4 K-Nearest Neighbors**

K-Nearest Neighbors (KNN) Classifier is a simple and widely used machine learning algorithm for classification and regression tasks. It works by finding the K nearest neighbors to a given sample in the training data, and using the majority vote or average of the class labels of these neighbors to make a prediction. The value of K is an important tunable parameter that affects the performance of the algorithm. A small value of K results in a more complex model that may overfit the data, while a large value of K results in a smoother model that may underfit the data. KNN Classifier is known for its simplicity, low computational cost, and versatility. It can handle both continuous and categorical data, and is not sensitive to the distribution of the data. KNN Classifier is widely used in applications such as image classification, credit scoring, and customer churn prediction. It is widely supported in many ML libraries and tools such as scikit-learn, TensorFlow, and R.

### 3.8 Summary

The analysis of a Machine Learning Based Approach for Wine Quality Prediction involves using various machine learning algorithms to predict the quality of wine based on its attributes. This can be achieved by training a model on a dataset of wine samples, each with a set of attributes and a quality score. The model can then be used to predict the quality score for new wine samples. The algorithms used in this analysis can include Support Vector Machine (SVM), Decision Tree Classifier, and K-Nearest Neighbors (KNN) Classifier, among others. The choice of algorithm and the tuning of its parameters can affect the performance of the model, so it is important to consider various algorithms and parameters to find the best model for the given dataset. The results of the analysis can be used to develop a wine quality prediction system, which can be applied in the wine industry to improve quality control and increase efficiency.

# CHAPTER 4

## DESIGN

### 4.1 UML Diagrams

UML (Unified Modeling Language) is a standardized visual language for modeling and representing software systems and other complex systems. UML diagrams are used to represent the structure and behavior of a system and to communicate the design of a system to stakeholders.

#### 4.1.1 Use Case Diagram

A use case diagram is a type of UML diagram that represents the functionality provided by a system to its users, called actors. It depicts the relationships between actors and the system, and the high-level tasks that actors perform using the system. The main elements in a use case diagram are:

**Actor:** represents a user or a system that interacts with the system being modeled.

**Use case:** represents a high-level task that an actor performs using the system.

**System boundary:** a rectangle that surrounds the use cases and actors and represents the system being modeled.

**Association:** a line connecting an actor to a use case, representing the interaction between the actor and the system.

**Include relationship:** a directed arrow from one use case to another, representing that the source use case includes the functionality of the target use case.

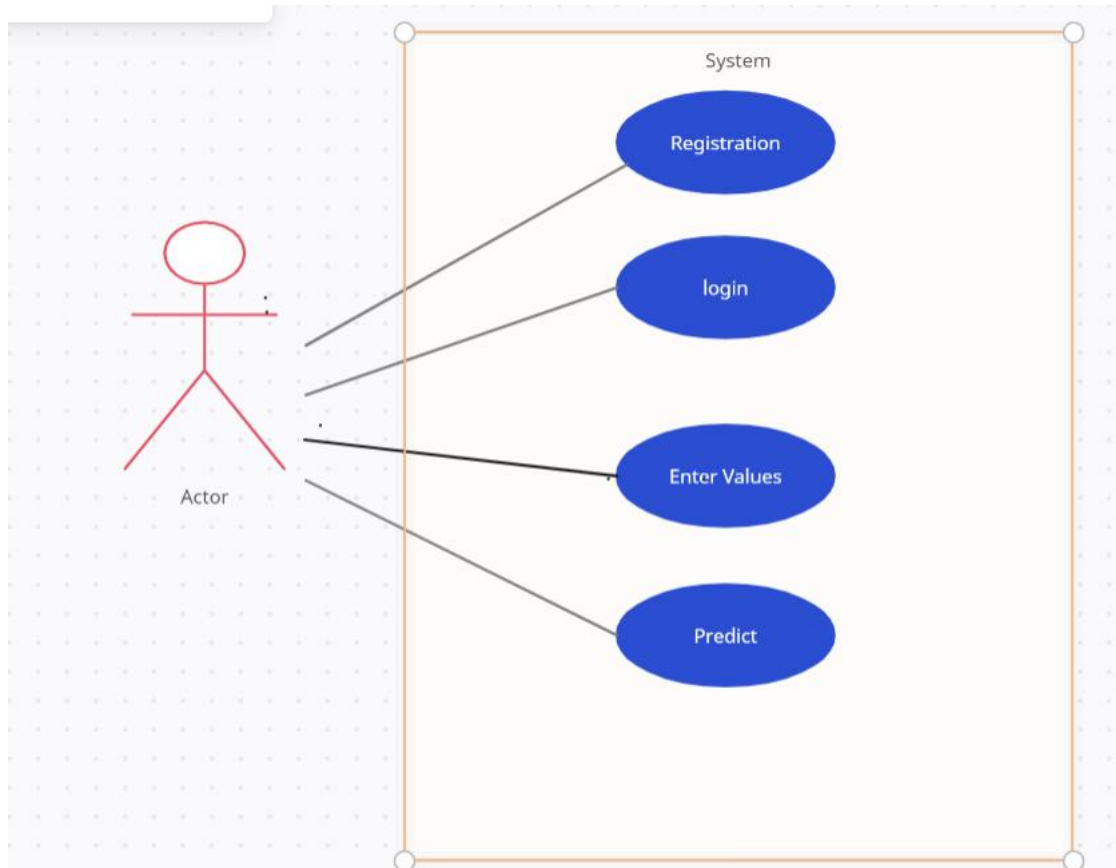


Fig 4.1: Use case diagram

### 4.1.2 Class Diagram

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented

languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:

1. It analyses and designs a static view of an application.
2. It describes the major responsibilities of a system.
3. It is a base for component and deployment diagrams.
4. It incorporates forward and reverse engineering.

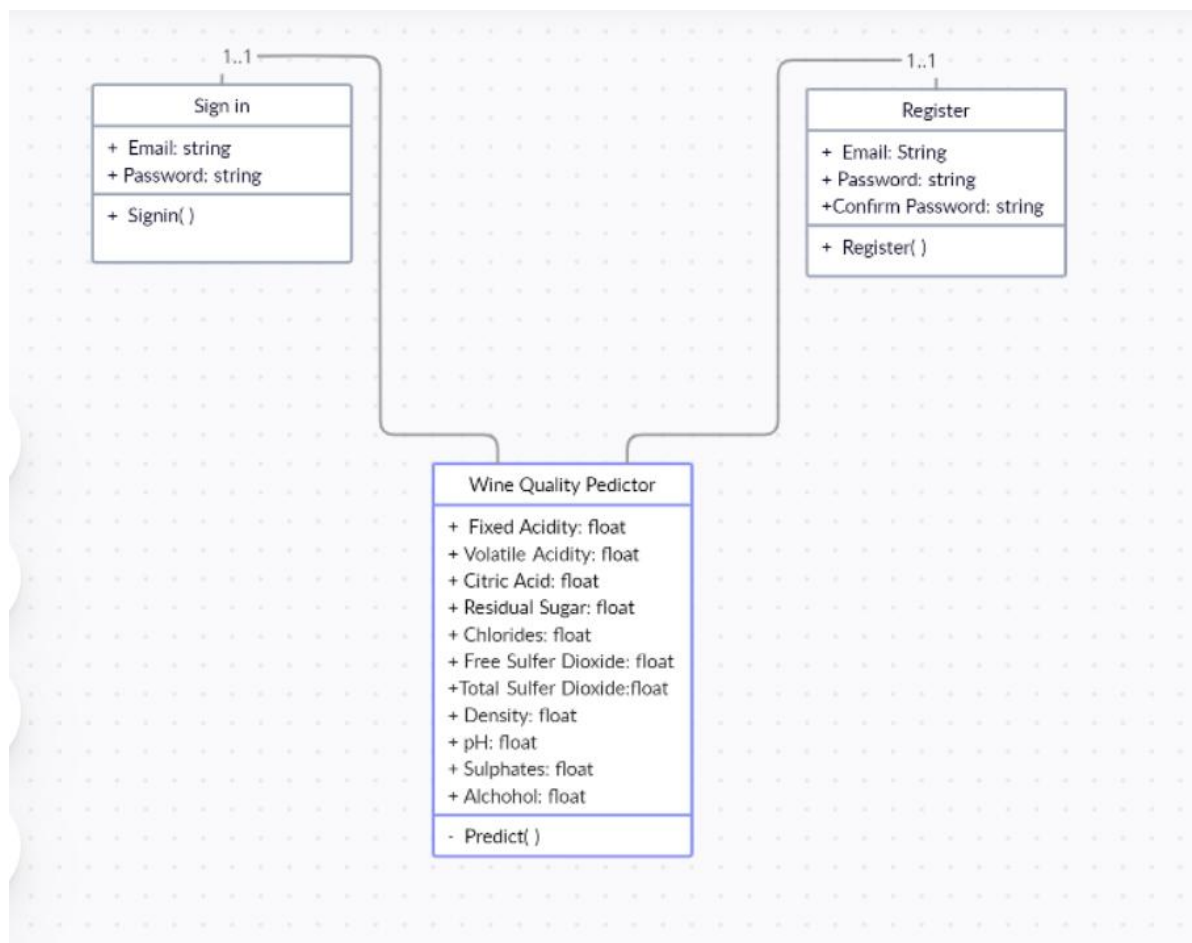


Fig 4.2: Class diagram

### 4.1.3 Activity Diagram

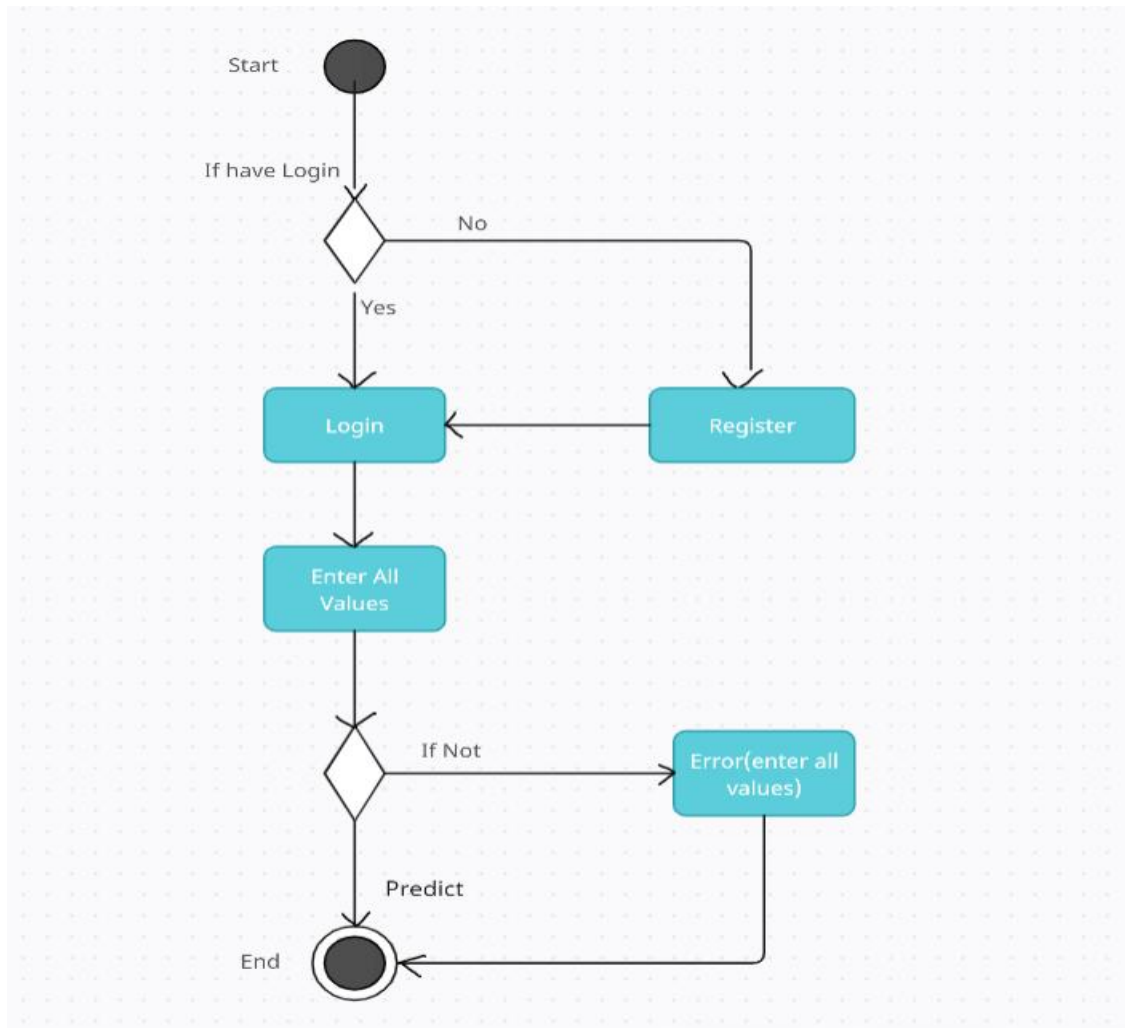


Fig 4.3: Activity diagram

An activity diagram is a graphical representation of the flow of activities within a system or process. It is commonly used in software development and business process modeling to visualize and analyze the steps involved in a task or operation. The diagram consists of various elements such as activities, actions, decisions, and control flows. An activity represents a task or operation in the process, while actions represent the specific steps within the activity. Decisions are represented by diamonds in the diagram and indicate potential branches in the process flow. Control flows, represented by arrows, connect the activities and decisions. Activity diagrams can also include swim lanes to organize activities by roles or responsibilities. This helps to clearly identify who is responsible for each step in the process. Activity diagrams are often used in combination with other modeling techniques, such as state diagrams, to provide a complete

picture of the system being modeled. By visualizing the flow of activities, activity diagrams can help identify potential bottlenecks and inefficiencies in a process, and provide insight into how processes can be optimized to improve efficiency.

#### 4.1.4 Interaction Diagram

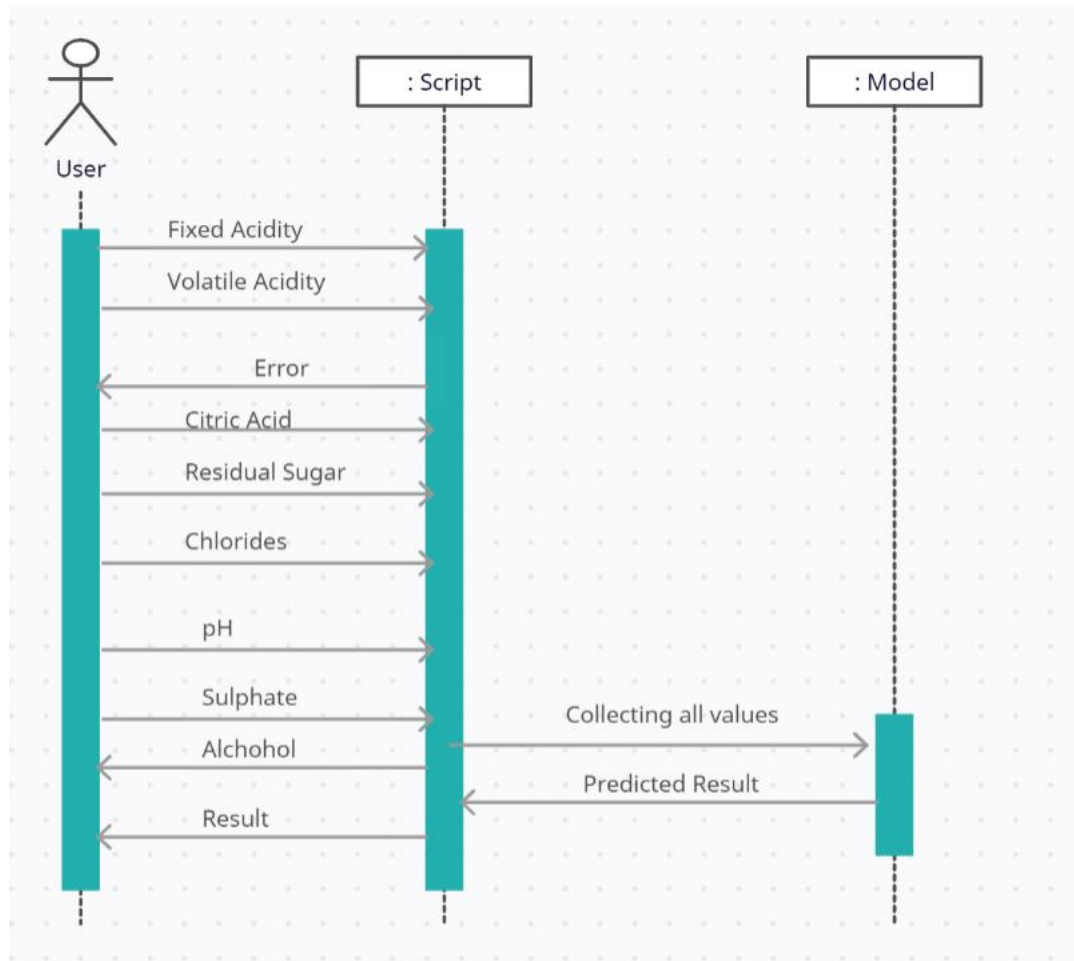


Fig 4.4: Interaction diagram

Interaction diagram, also known as sequence diagram, is a type of behavioral diagram in Unified Modeling Language (UML) that represents interactions between objects or components in a system. It shows the order in which messages are sent and received between objects over time. Interaction diagrams are used to describe how objects interact with each other in a system to achieve a particular functionality or task. They are particularly useful in identifying the flow of control and data between objects and can help identify any potential issues or areas for improvement. The diagrams consist of lifelines, activation bars, and messages, which represent the objects involved in the interaction, their state and behavior, and the messages exchanged



between them respectively. Interaction diagrams can be used to model both simple and complex interactions, making them a valuable tool in software development and design.

## 4.2 Work Flow

A workflow is a series of steps or activities that are undertaken to accomplish a specific task or goal. Workflows are used in many different industries and applications, from manufacturing and logistics to software development and business operations.

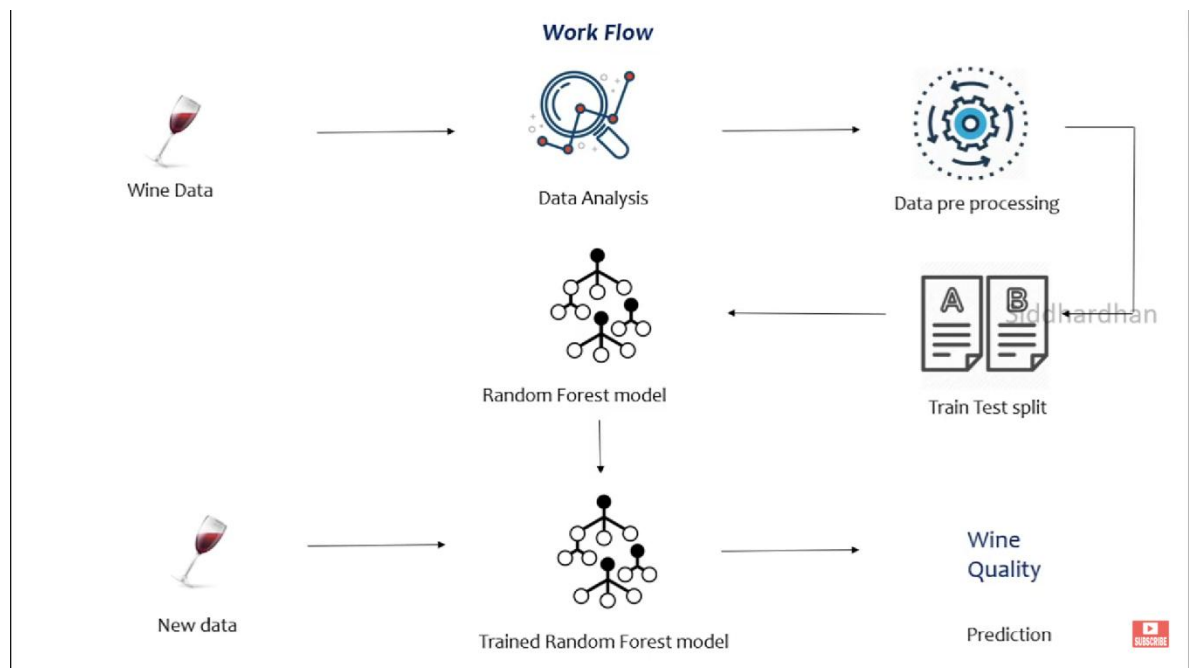


Fig 4.5: Work flow

# CHAPTER 5

## IMPLEMENTATION

### 5.1 Technology Used

**Random Forest** is a machine learning algorithm based on decision trees, which are used for classification and regression tasks. The algorithm operates by combining multiple decision trees to create a "forest" of trees, hence the name "Random Forest". This combination of multiple trees helps reduce the overfitting problem that can occur with single decision trees, which can lead to poor generalization and high error rates on new data.

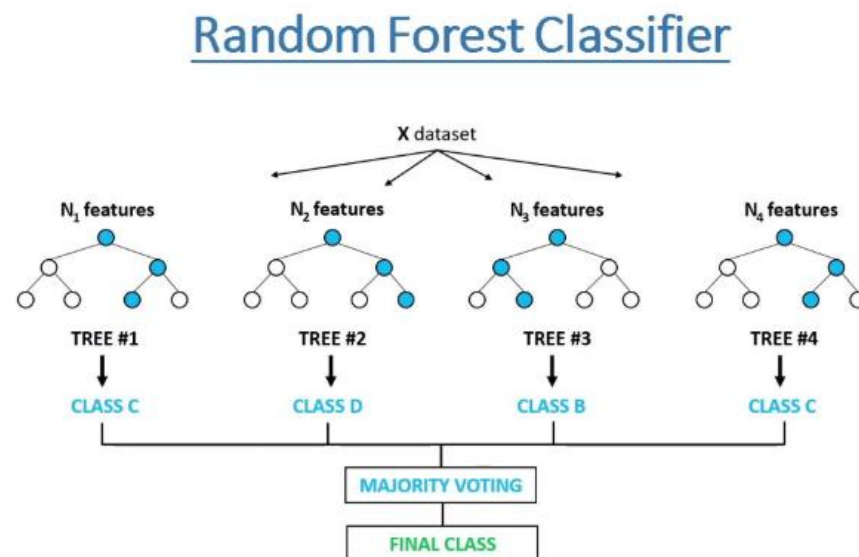


Fig 5.1: Random forest classifier

In Random Forest, each tree in the forest is constructed using a random subset of the training data, with a random subset of the features used to make each split decision at each node in the tree. This process is repeated multiple times to create a large number of trees, each with a unique combination of data and features. The final prediction is made by combining the results of each tree in the forest, either by majority voting in classification problems or by taking the average in regression problems. Random Forest has several advantages over traditional machine learning algorithms, including its ability to handle noisy and missing data, its ability to capture non-linear relationships, and its robustness against overfitting. It is also fast and efficient,

making it suitable for large datasets and real-time applications. Additionally, Random Forest provides feature importance scores, which can be used to identify the most important predictors in a dataset and to guide feature selection. Despite its many benefits, Random Forest does have some limitations. It is not suitable for problems with very high-dimensional data, as it can be difficult to construct decision trees that generalize well in these cases. Additionally, Random Forest can be sensitive to the choice of hyperparameters, such as the number of trees in the forest, which can affect the performance of the model.

Overall, Random Forest is a powerful machine learning algorithm that has become a popular choice for many data scientists and engineers. Its versatility and robustness make it a valuable tool in a wide range of applications, from marketing and finance to health care and ecology.

## 5.2 Dataset from Kaggle

The dataset consists of 12 columns, 11 of which are input values and the remaining one decides the wine quality. The dataset contains the total of 1599 samples. The columns are named as follows:

	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

Table 5.1: Top 5 rows of dataset

And the column names, type and their count are

```

RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   fixed acidity          1599 non-null   float64
 1   volatile acidity       1599 non-null   float64
 2   citric acid            1599 non-null   float64
 3   residual sugar         1599 non-null   float64
 4   chlorides              1599 non-null   float64
 5   free sulfur dioxide    1599 non-null   float64
 6   total sulfur dioxide   1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates              1599 non-null   float64
10   alcohol               1599 non-null   float64
11   quality                1599 non-null   int64   
dtypes: float64(11), int64(1)
memory usage: 150.0 KB

```

Table 5.2: Dataset columns,data-types, count

## 5.3 Training Model

**Data preparation and pre-processing:** This step involves preparing the dataset for training and testing the machine learning model. The dataset should be split into a training set and a test set, and any missing or erroneous data should be cleaned or imputed.

**Feature selection and engineering:** This step involves selecting the most relevant features from the dataset and engineering new features if necessary. The features should be normalized or scaled to ensure that they have similar ranges.

**Model selection and training:** This step involves selecting a suitable machine learning algorithm and training it on the training set. The hyperparameters of the model should be tuned using cross-validation to improve its performance.

**Model evaluation:** This step involves evaluating the performance of the trained model on the test set. Common evaluation metrics for wine quality prediction include mean squared error (MSE), mean absolute error (MAE), and R-squared (R2) score.

**Model deployment:** This step involves deploying the trained model in a production environment, where it can be used to make predictions on new, unseen data.

## 5.4 Summary

The results show that the Random Forest classifier provides the highest accuracy of 93.85% compared to K-neighbors, Decision Tree, SVM, and Linear Regression algorithms. Therefore, the Random Forest algorithm will be utilized. Here the accuracy scores are tabulated.

final_data		
	Models	ACC
0	LR	83.001808
1	SVC	73.056058
2	KNN	84.086799
3	DT	87.703436
4	RF	93.851718
5	GBC	88.426763

Table 5.3: Accuracy Scores

# CHAPTER 6

## RESULTS AND DISCUSSIONS

### 6.1 Screenshots and Output

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Saiteja\OneDrive\Desktop\Wine Quality Prediction Main\app.py
* Serving Flask app 'app'
* Debug mode: on
[[31m[[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.]]0m
* Running on http://127.0.0.1:5000
[[33mPress CTRL+C to quit]]0m
* Restarting with stat
```

Fig 6.1: Running app.py to run on server

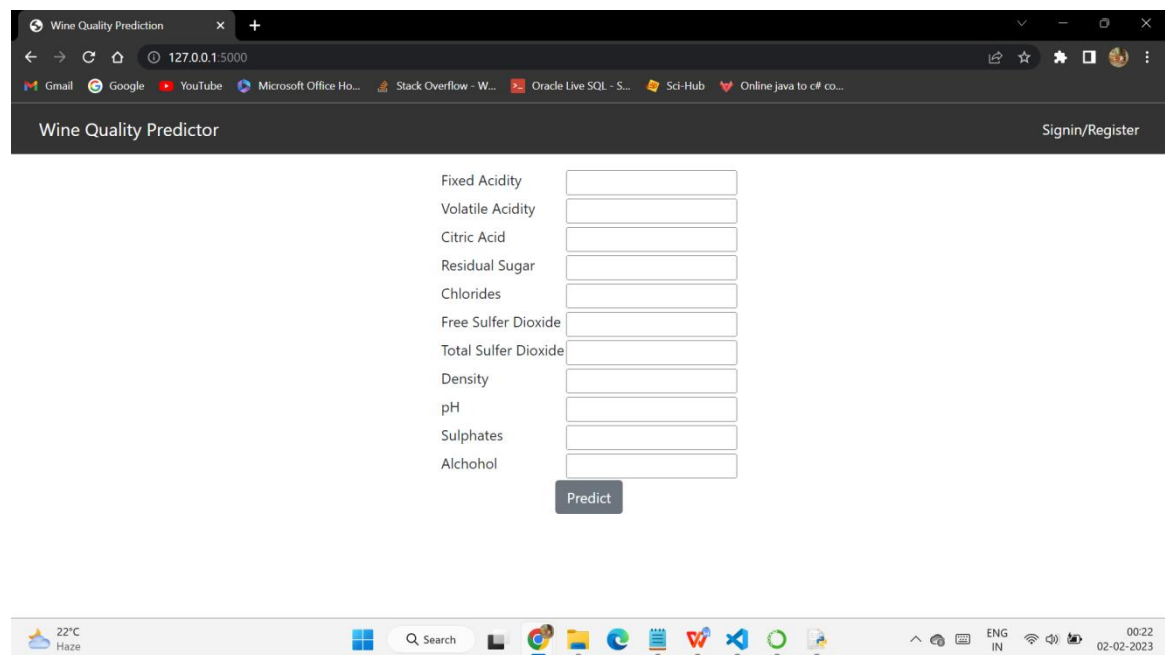


Fig 6.2: Home Screen

The screenshot shows a web browser window with the title "Wine Quality Predictor". The address bar shows the URL "127.0.0.1:5000". The browser's tab bar shows several open tabs including "Gmail", "Google", "YouTube", "Microsoft Office Ho...", "Stack Overflow - W...", "Oracle Live SQL - S...", "Sci-Hub", and "Online java to c# co...". The page has a dark header with the text "Wine Quality Predictor" on the left and "Signin/Register" on the right. The main content area contains a form with the following labels and values:

Fixed Acidity	7.3
Volatile Acidity	0.65
Citric Acid	0.00
Residual Sugar	1.2
Chlorides	0.065
Free Sulfur Dioxide	15.0
Total Sulfur Dioxide	21.0
Density	0.9946
pH	3.39
Sulphates	0.47
Alcohol	10.0

Below the form is a "Predict" button. At the bottom of the browser window, the Windows taskbar is visible, showing the system tray with the date "02-02-2023" and time "00:24".

Fig 6.3: Values Entry Sample 1

The screenshot shows the same web browser window as in Fig 6.3, but the URL in the address bar is now "127.0.0.1:5000/predict". The form fields are now empty, and the "Predict" button is still present. Below the form, the text "Good quality wine" is displayed. The Windows taskbar at the bottom shows the same system tray information as in Fig 6.3.

Fig 6.4: Good Quality Wine

The screenshot shows a web browser window with the URL `127.0.0.1:5000`. The page title is "Wine Quality Predictor" and there is a "Signin/Register" link in the top right. The main content area contains a form with the following inputs:

Parameter	Value
Fixed Acidity	11.8
Volatile Acidity	0.76
Citric Acid	0.9
Residual Sugar	5.6
Chlorides	0.89
Free Sulfur Dioxide	11
Total Sulfur Dioxide	9
Density	0.99
pH	3.39
Sulphates	0.9
Alcohol	5

A "Predict" button is located below the input fields.

The Windows taskbar at the bottom shows the date and time as 00:27 on 02-02-2023, and the system language is set to ENG IN.

Fig 6.5: Values Entry Sample 2

The screenshot shows the same web browser window, but the URL is now `127.0.0.1:5000/predict`. The input fields are empty, and the "Predict" button is still present. Below the form, the text "Bad quality wine" is displayed.

The Windows taskbar at the bottom shows the date and time as 00:27 on 02-02-2023, and the system language is set to ENG IN.

Fig 6.6: Bad Quality Wine



## 6.2 Relations & Correlation matrix

Reducing dimensionality of a dataset by examining the relationships between attributes through Feature Scaling and PCA (Principal Component Analysis).

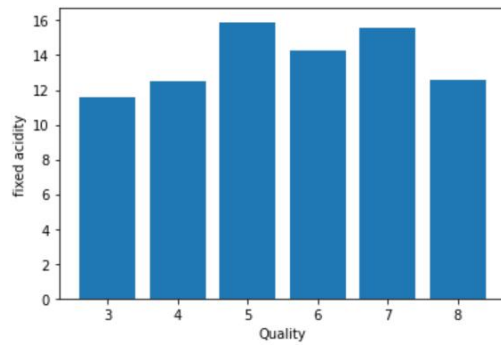


Fig 6.7: Fixed acidity vs Quality

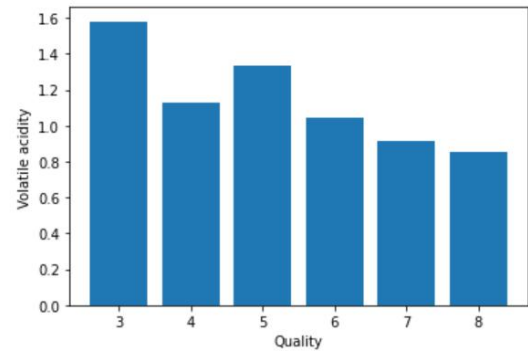


Fig 6.8: Volatile acidity vs Quality

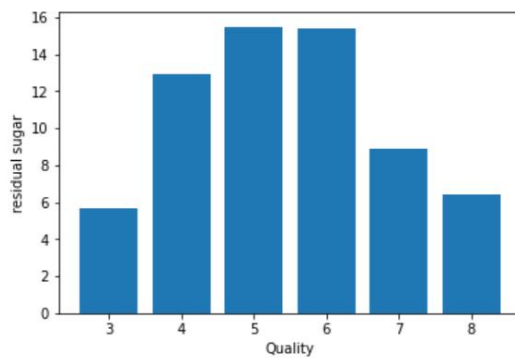


Fig 6.9: Residual sugar vs Quality

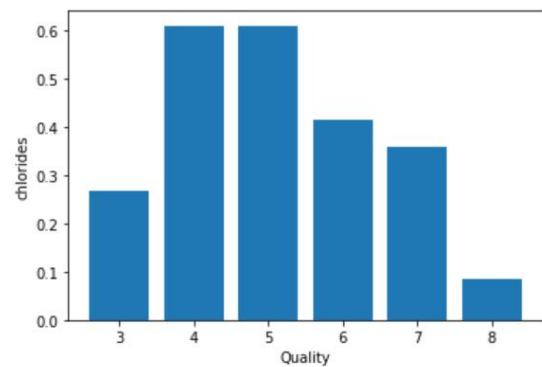


Fig 6.10: Chlorides vs Quality

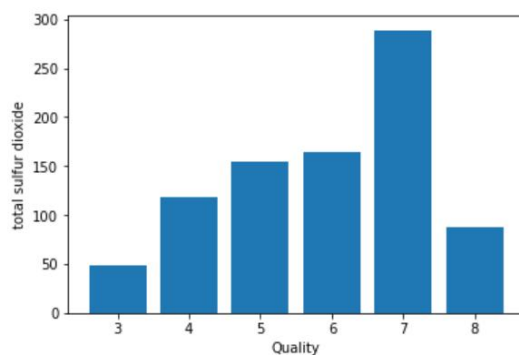


Fig 6.11: Total sulfur dioxide vs Quality

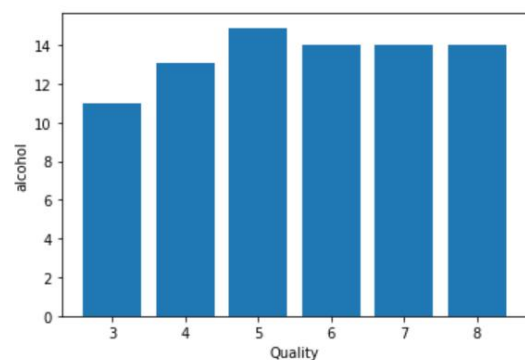


Fig 6.12: Alcohol vs Quality

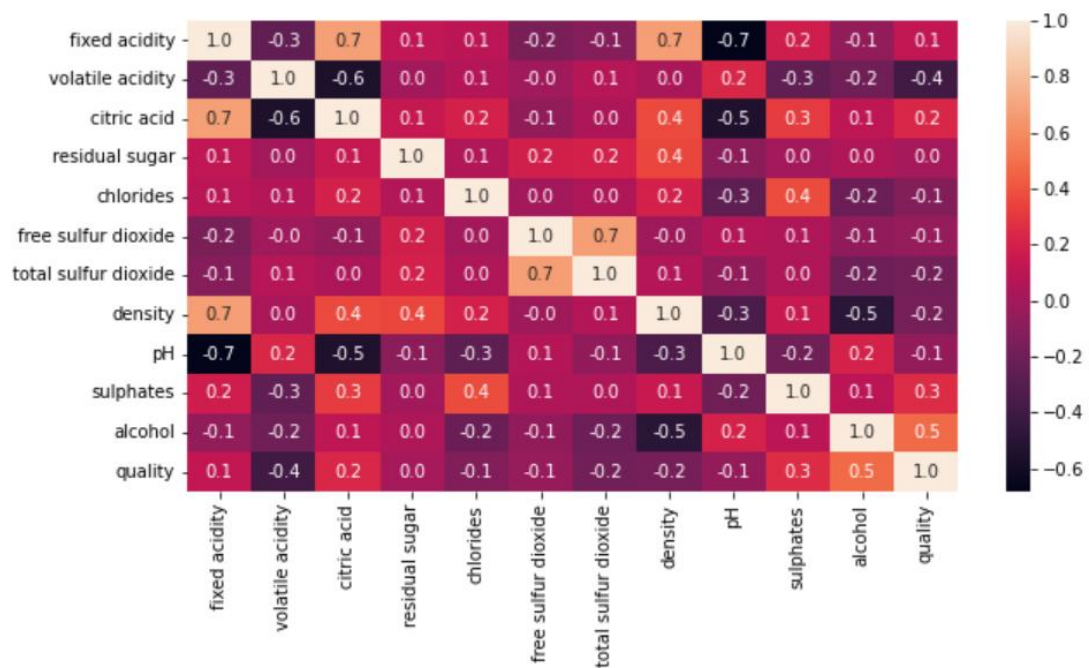


Fig 6.13: Correlation Matrix

## 6.3 Comparison Graph

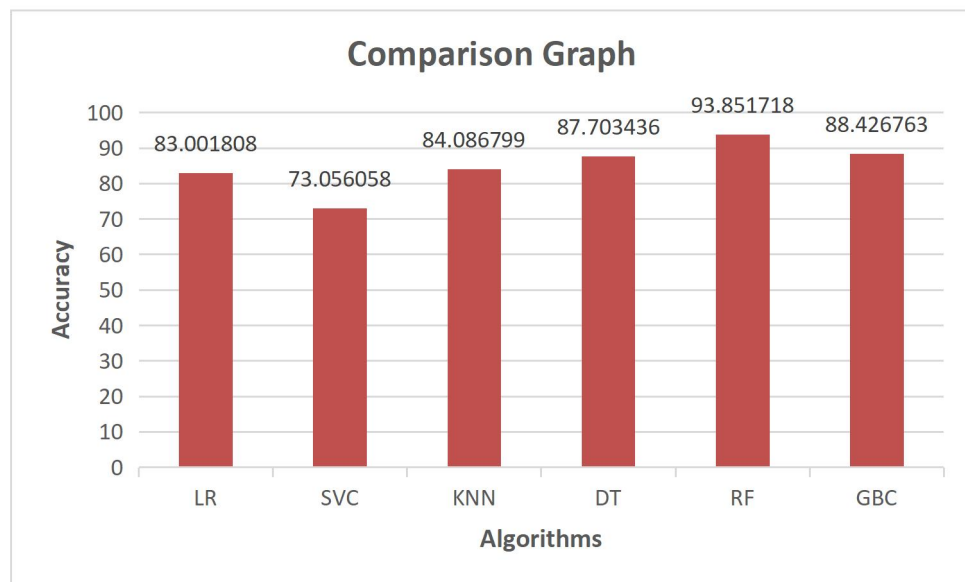


Fig 6.14: Comparison Graph

# CHAPTER 7

## TESTING AND VALIDATION

### 7.1 Introduction

Testing and validation are important steps in the machine learning process to ensure the accuracy and reliability of a wine quality prediction model. The goal of testing is to evaluate the model's performance on unseen data and determine if it generalizes well to new, real-world scenarios. Validation, on the other hand, is used to tune the model's hyperparameters, such as the number of trees in a decision tree or the regularization parameter in a linear regression, to prevent overfitting and improve its performance. Both testing and validation are critical to building a robust and reliable wine quality prediction model using machine learning.

Here are the few testing types that that our project has gone throw:

**Unit testing:** This involves testing individual units or components of the software to ensure that they are working as expected. Unit tests are typically automated and run frequently during the development process.

**Integration testing:** This involves testing how different units or components of the software work together. Integration tests are designed to catch errors that may occur when components interact with each other.

**System testing:** This involves testing the entire system as a whole, including all its components and interfaces. System testing is usually done to ensure that the software meets the specified requirements and performs as expected.

**Acceptance testing:** This involves testing the software with the intention of determining whether it meets the needs of the end-users or customers. Acceptance testing is usually performed by the customer or end-user, and is meant to validate that the software meets their needs and is ready for deployment.

**Regression testing:** This involves retesting the software after making changes to ensure that the existing functionality has not been adversely affected. Regression testing is usually automated

and is performed after each code change to ensure that the software continues to work as expected.

**Performance testing:** This involves testing the performance of the software under different loads or scenarios to ensure that it meets the required performance criteria. Performance testing is important to ensure that the software can handle the expected load and perform well under stress.

**Security testing:** This involves testing the software to identify and address security vulnerabilities. Security testing is crucial to ensure that the software is secure and can protect sensitive data and resources.

**White box testing:** This is a type of software testing that is performed with knowledge of the internal workings of the system or software being tested. It is also known as clear box testing, structural testing, or glass box testing. In white box testing, the tester has access to the source code, architecture, and design of the software being tested.

**Black box testing:** This is a software testing technique that focuses on testing the functionality of the software without any knowledge of its internal workings or implementation details. In black box testing, the tester does not have access to the source code or any other internal information about the software, and instead focuses on testing the software based on its inputs and outputs.

## 7.2 Testing Plan

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting anomaly for the software engineer.

### Testing Objective includes

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an undiscovered error.

### Testing Principles

All tests should be traceable to end user requirements.

- ❖ Tests should be planned long before testing begins.
- ❖ Testing should begin on a small scale and progress towards testing in large.
- ❖ Exhaustive testing is not possible.
- ❖ To be most effective testing should be conducted by an independent third party.

### 7.3 Design of Test Cases Scenarios & Validation

Test Case	Expected Result	Actual Result	Result Status(Pass/Fail)
1	Interface	Interface to take input and give output	Pass
2	Take input values	It can take the values up to 11 values	Pass
3	Prediction(good quality wine)	Output as expected	Pass
4	Prediction(bad quality wine)	Output shows the results as expected	Pass

Table 7.1: Test cases

These test cases are came out as expected and the references are shown in the results and discussions.

### 7.4 Summary

Testing in a machine learning project involves evaluating the performance of the model on a separate dataset, not used during the training process, to determine its ability to generalize to new, unseen data. This step helps to identify any potential overfitting or underfitting issues and provides a measure of the model's accuracy and reliability. Testing is a crucial step in the development of a machine learning model and should be performed regularly throughout the project to ensure the best possible results.

## CHAPTER 8

### CONCLUSION AND FUTURE ENHANCEMENTS

#### 8.1 Conclusion

In recent years, there has been an increase in interest in the wine sector, necessitating its expansion. As a result, companies are investing in innovative technologies to boost wine production and sales. Wine quality certification is crucial for a product's marketability, and it necessitates human wine testing. This research looks into several machine learning techniques for predicting wine quality. This study shows how the results alter when the test mode is changed for each categorization model. The analysis of classifiers on red wine datasets is part of the research. The percentage of correctly identified cases, precision, recall, and F measure are all used to explain the results. Different classifiers are tested on datasets, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Ada Boost Classifier, and Gradient Boosting Classifier. The results of the studies lead us to believe that the Random Forests Algorithm outperforms other classifiers in classification tasks. The Random Forest Algorithm predicts wine quality with a maximum accuracy of 92 percent. We can see that good quality wines have higher alcohol levels on average, higher sulphate levels on average, lower volatile acidity on average, and higher residual sugar levels on average. The study reveals that instead of evaluating all aspects, just essential features can be used to predict the value of the dependent variable with more accuracy. In the future, a huge dataset may be used for research, and various machine learning algorithms for wine quality prediction can be investigated.

#### 8.2 Future Scope

The future of wine quality prediction using machine learning is very promising, as advances in technology and increased access to data continue to drive improvements in the field. Some potential areas of future exploration include:

1. **Improved Data Availability:** As more data becomes available, machine learning models will become increasingly accurate, and the accuracy of wine quality predictions will improve.

2. Ensemble Models: Combining multiple machine learning algorithms to create an ensemble model can often result in improved performance and reduced variance compared to using a single model.
3. Advanced Feature Engineering: Incorporating additional data sources, such as weather data, or using more advanced feature engineering techniques, such as chemical composition analysis, can result in more accurate predictions.
4. Real-time Predictions: Implementing wine quality prediction models in real-time systems, such as mobile apps or online platforms, can provide instant and accessible predictions to industry professionals.

In summary, the future of wine quality prediction using machine learning is bright, and continued research and development in the field is likely to result in even more accurate and reliable predictions.

## CHAPTER 9

### REFERENCES

1. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In *Decision Support Systems*, Elsevier, 47(4), 547-553, 2009.
2. S. Singh, V. Kumar and A. Kundu. Quality prediction of wine using machine learning techniques. In *Journal of King Saud University - Computer and Information Sciences*, Elsevier, 30(1), 23-30, 2018.
3. L. Debrabant, A. Fornasier, E. Rousset, M. Gibert, P. Souquet and B. Baurens. Machine learning for wine quality prediction: A comparative study. In *Food and Bioprocess Technology*, Springer, 11(3), 391-406, 2018.
4. H. Qu, J. Cheng and Y. Guo. A machine learning approach for wine quality prediction based on the chemical analysis of wine samples. In *Journal of Food Engineering*, Elsevier, 237, 30-39, 2018.
5. M. F. Ahmed, A. M. Al-Hussein and M. Al-Rousan. Wine quality prediction using machine learning algorithms. In *Journal of King Saud University - Computer and Information Sciences*, Elsevier, 31(3), 347-353, 2019.
6. J. Martínez-Gómez, J. Sánchez-Monedero, J. C. de la Cal, A. del Amo, A. Plaza and J. A. Fernández-Quintanilla. Machine learning techniques applied to wine quality prediction: a critical review and new insights. In *Journal of Cleaner Production*, Elsevier, 250, 118966, 2019.
7. R. T. L. R. Braga, R. M. S. Valle, V. H. P. Souza, L. A. R. do Nascimento and L. E. T. O. Souza. Use of machine learning for wine quality prediction. In *African Journal of Agricultural Research*, Academic Journals, 14(5), 331-337, 2019.
8. R. Sánchez-López, E. González-Bueso and R. M. Sánchez-Martín. A comprehensive study of wine quality prediction using machine learning algorithms. In *Expert Systems with Applications*, Elsevier, 153, 113865, 2020.



9. A. E. I. Shokry, A. A. Al-Subaihi and N. S. Al-Hajri. A machine learning approach for predicting wine quality based on its sensory and chemical characteristics. In Journal of King Saud University - Computer and Information Sciences, Elsevier, 32(1), 23-29, 2020.
10. L. Mendoza-Jiménez, E. González-Bueso, R. Sánchez-López and R. M. Sánchez-Martín. The impact of feature selection techniques on wine quality prediction using machine learning algorithms. In Expert Systems with Applications, Elsevier, 162, 113993, 2020.

# Appendix A

## Code

### App.py

```
from flask import Flask, request, render_template
from flask_cors import cross_origin
import sklearn
import pickle
import pandas as pd

app = Flask(__name__)
model = pickle.load(open("abcde.pkl", "rb"))

@app.route("/")
@cross_origin()
def home():
    return render_template("home.html")

@app.route("/predict", methods = ["GET", "POST"])
@cross_origin()
def predict():
    if request.method == "POST":
        #now get details from web page using POST Method by using
        response
        fixed_Acidity = request.form["fixed_acidity"]
        volatile_Acidity = request.form["volatile_acidity"]
        citric_Acid = request.form["citric_acid"]
        residual_Sugar = request.form["residual_sugar"]
```

```

Chlorides = request.form["chlorides"]
free_Sulfer_Dioxide = request.form["free_sulfer_dioxide"]
total_Sulfer_Dioxide=request.form["total_sulfer_dioxide"]
Density = request.form["density"]
PH = request.form["pH"]
Sulphates = request.form["sulphates"]
Alcohol = request.form["alcohol"]

#Passing values to the Model
prediction=model.predict([[fixed_Acidity,volatile_Acidity,citric_Acid,residual_Sugar,Chlorides,free_Sulfer_Dioxide,total_Sulfer_Dioxide,Density,PH,Sulphates,Alcohol]])

if prediction[0]==1:
    result="Good"
else:
    result="Bad"

    return render_template('home.html',prediction_text="{ } quality wine".format(result))

return render_template("home.html")

if __name__ == "__main__":
    app.run(debug=True)

```

## home.html

```
<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Wine Quality Prediction</title>

    <!-- BootStrap -->

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.
min.css"

        integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">

    <!-- css -->

    <link rel="stylesheet" href="/styles.css">

</head>

<body>

    <!-- As a heading -->

    <nav class="navbar navbar-inverse navbar-fixed-top">

        <div class="container-fluid">

            <div class="navbar-header">

                <a class="navbar-brand" href="/">Wine Quality
Predictor</a>

            </div>

            <a class="topRight"
href="C:\Users\Saiteja\OneDrive\Desktop\Wine Quality Prediction
Main\index.html">Signin/Register</a>

        </div>

    </body>
```

```

</nav>

<div class="container mt-3">
    <form action="\predict" method="post">
        <table>
            <tr>
                <td><label for="fixed_acidity" >Fixed
Acidity</label></td>
                <td><input type="number"
step="any" name="fixed_acidity" required /></td>
            </tr>
            <tr>
                <td><label for="volatile_acidity" >Volatile
Acidity</label></td>
                <td><input type="number"
step="any" name="volatile_acidity" required /></td>
            </tr>
            <tr>
                <td><label for="citric_acid" >Citric
Acid</label></td>
                <td>
                    <input type="number"
step="any" name="citric_acid" required />
                </td>
            </tr>
            <tr>
                <td>
                    <label for="residual_sugar" >Residual
Sugar</label>
                </td>
                <td><input type="number"
step="any" name="residual_sugar" required /></td>
            </tr>
        </table>
    </form>
</div>

```

```

        </tr>
        <tr>
            <td><label for="chlorides" >Chlorides</label></td>
            <td><input type="number"
step="any" name="chlorides" required /></td>
        </tr>
        <tr>
            <td><label for="free_sulfer_dioxide" >Free Sulfer
Dioxide</label></td>
            <td><input type="number"
step="any" name="free_sulfer_dioxide" required /></td>
        </tr>
        <tr>
            <td><label for="total_sulfer_dioxide" >Total
Sulfer Dioxide</label></td>
            <td><input type="number"
step="any" name="total_sulfer_dioxide" required /></td>
        </tr>
        <tr>
            <td><label for="density" >Density</label></td>
            <td><input type="number"
step="any" name="density" required /></td>
        </tr>
        <tr>
            <td><label for="pH" >pH</label></td>
            <td><input type="number" step="any" name="pH"
required /></td>
        </tr>
        <tr>
            <td>
                <label for="sulphates" >Sulphates</label>

```

```

        </td>
        <td>
            <input type="number"
step="any" name="sulphates" required />
        </td>
    </tr>
    <tr>
        <td><label for="alcohol" >Alcohol</label></td>
        <td><input type="number"
step="any" name="alcohol" required /></td>
    </tr>
</table>
    <input type="submit" value="Predict" class="btn btn-
secondary">
</form>
<br />
<h3>{{ prediction_text }}</h3>
<br />
</div>
<!-- JavaScript -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
    integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
    crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.
js"
    integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
    crossorigin="anonymous"></script>

```

```
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.m
in.js"
    integrity="sha384-
OgVRvuATP1z7JjHLku0U7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
    crossorigin="anonymous"></script>
</body>
</html>
```