CHAPTER 1

INTRODUCTION

1.1 Overview

The quality of the wine is a very important part for the consumers as well as the manufacturing industries. Industries are increasing their sales using product quality certification. Nowadays, all over the world wine is a regularly used beverage and the industries are using the certification of product quality to increases their value in the market. Previously, testing of product quality will be done at the end of the production, this is time taking process and it requires a lot of resources such as the need for various human expertsfor 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's experts it is a challenging task.

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

The research aims to what wine features are important to get the promising result by implementing the machine learning classification algorithms Random Forest, using the wine quality dataset.

The wine quality dataset is publically available on the UCI machine learning repository (Cortez et al., 2009). The dataset has two files red wine and white wine variants of the Portuguese "Vinho Verde" wine. It contains a large collection of datasets that have been usedfor 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 pre-processing 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.

1.2 EXISTING SYSTEM

The current system is a time-consuming process and requires the assessment given by human experts, which makes this process very expensive. Also, the price of red wine depends on a rather abstract concept of wine appreciation by wine tasters, opinion among whom may have a high degree of variability. Another vital factor in red wine certification and quality assessment is physicochemical tests, which are laboratory-based and consider factors like acidity, pH level, sugar, and other chemical properties.

1.2.1 DISADVANTAGES OF EXISTING SYSTEM:

- Hardware dependent
- Complex Algorithms are foreseen disadvantages of Neural Networks
- Even when the results are accurate human analysts can't track and check the derivations. Most neural networks are black-box systems generating results based on experience and not on specified programs, making it difficult for modifications.
- Various theorems are used to give only a probable value. All the theories used are not entirely suitable to give results possible for all situations, and the desired output may not be obtained. This uncertainty is among the eye-opening problems with Neural Networks.
- Whatever data is fed to the machine, it acts accordingly. The more amount of data is used during training, the more accurate the results are. Dependency on data is one of the leading disadvantages of Neural Networks, as some have to be on the maintenanceside to watch it. Since there are errors in the data, the result will be faulty, which poses serious threats.

1.3 PROPOSED SYSTEM

In a proposed system, Machine-learning techniques are employed to predict wine quality in this study. The processes in the suggested methodology is depicted in flow diagram. Pre-processing is done on the first wine dataset. The data is further divided into training (80%) and testing (20%) sets, with the training set being utilized to train the model utilizing RandomForest and Decision Tree Classifier algorithms. The testing set is used to determine the accuracy of several models, and then conclusions are generated to choose the optimal model for predicting wine quality. The trained model is used to determine the testing set's correctness

1.3.1 Advantages of Proposed System:

- The Accuracy of our proposed system wine quality prediction system model isgenerally very high.
- The proposed system efficiency is particularly notable in Large Data sets.
- The proposed system model provides an estimate of important variables in classification.
- Forests Generated can be saved and reused.
- Unlike other models it doesn’t overfit with more features
- The proposed system model provides an effective way of handling missing data.
- The proposed system model is comparatively less impacted by noise.
- The proposed system model is usually robust to outliers and can handle them automatically.

CHAPTER 2

LITERATURE SURVEY

2.1 Overview

Over a decade, a horde of exploration effort allocated through recommender system in dissimilar provinces then sundry elucidations devour stayed anticipated in the prose. In mandate to realize this project, basis has remained conceded out. The info is achieved from various source such as files, tutelages, journals, as well as internet. The acquired info was worthwhile then exploited in the project. The deliberation of info is founded on the practice, hardware then software used in the project.

2.2 Literature review

1) A machine learning application in wine quality prediction

AUTHORS: Piyush Bhardwaj , Parul Tiwari , Kenneth Olejar Jr , Wendy Parr and Don Kulasiri

The wine business relies heavily on wine quality certification. The excellence of New Zealand Pinot noir wines is well-known worldwide. Our major goal in this research is to predict wine quality by generating synthetic data and construct a machine learning model based on this synthetic data and available experimental data collected from different and diverse regions across New Zealand. We utilised 18 Pinot noir wine samples with 54 different characteristics (7 physiochemical and 47 chemical features). We generated 1381 samples from 12 original samples using the SMOTE method, and six samples were preservedfor model testing. The findings were compared using four distinct feature selection approaches. Important attributes (referred as essential variables) that were shown to berelevant in at least three feature selection methods were utilised to predict wine quality.Seven machine learning algorithms were trained and tested on a holdout original sample. Adaptive Boosting (AdaBoost) classifier showed 100% accuracy when trained and evaluated without feature selection, with feature selection (XGB), and with essential variables (features found important in at least three feature selection methods). In the presence of essential variables, the Random Forest (RF) classifier performance was increased.

2) Prediction of Wine Quality Using Machine Learning Algorithm

AUTHORS: K. R. Dahal, J. N. Dahal, H. Banjade and S. Gaire

As a subfield of Artificial Intelligence (AI), Machine Learning (ML) aims to understand the structure of the data and fit it into models, which later can be used in unseen data to achieve the desired task. ML has been widely used in various sectors such as in Businesses, Medicine, Astrophysics, and many other scientific problems. Inspired by the success of ML in different sectors, here, we use it to predict the wine quality based on the various parameters. Among various ML models, we compare the performance of Ridge Regression (RR), Support Vector Machine (SVM), Gradient Boosting Regressor (GBR), and multi-layer Artificial Neural Network (ANN) to predict the wine quality. Multiple parameters that determine the wine quality are analyzed. Our analysis shows that GBR surpasses all other models' performance with MSE, R, and MAPE of 0.3741, 0.6057, and 0.0873 respectively. This work demonstrates, how statistical analysis can be used to identify the components that mainly control the wine quality prior to the production. This will help wine manufacturer to control the quality prior to the wine production.

3) Wine Quality Analysis Using Machine Learning

AUTHORS: Shaw, B., Suman, A.K. and Chakraborty, B

Almost from the beginning of mankind, there has been the existence of different kinds of wine. It has also become very important for us to know the quality of the wine, before consuming it. In the last few decades, the food industry has grown enormously and so are the food quality analysis and its "rating" process. We often come across cases in which aconsumer falls sick because of consuming low-quality food, so it has become a necessary evilfor us to have a quality analysis of a product before selling the product, "evil" because it addsup extra cost to the production of the final product. Similarly, it is also necessary to do a quality analysis of wine and there have been different methods used to determine the quality of the wine, but we often get confused regarding which method to rely on! This paper focuseson the comparative study over different classification algorithms for wine quality analysis which are: SVM, random forest and multilayer perceptron and to know which of the above- mentioned classification algorithms give more accurate result.

4) Wine Quality Analysis Using Machine Learning Algorithms

AUTHORS: Mahima, Ujjawal Gupta and Yatindra Patidar

Wines are being produced since thousands of years. But, it is a complex process to determine the relation between the subjective quality of a wine and its chemical composition. Industries use Product Quality Certification to promote their products and become concern for every individual who consumes any product. It is not possible to ensure quality with experts with such a huge demand of product as it will increase the cost. Wine-makers need a permanent solution to optimize the quality of their wine. This paper explores the space to easy out and make the whole process cost-effective and more trustworthy using machine learning. Itallows to build a model with user interface which predicts the wine quality by selecting the important parameters of wine which play a significant role in determining the wines quality. Random forest algorithm is used in determining wines' quality whose correctness would further be escalated using KNN which makes our model dynamic. Output of this proposed model is used to determine the wines' quality on a scale of Good, Average or Bad. This proposed model can further be applied to several other products which need quality certification. Our prediction model provides ideal solution for the analysis of wine, which makes this whole process more efficient and cheaper with less human interaction.

5) Prediction of Different Types of Wine Using Nonlinear and Probabilistic Classifiers

AUTHORS: Satyabrata Aich, Mangal Sain and Jin-Han Yoon

In the past few years, machine-learning techniques have garnered much attention across disciplines. Most of these techniques are capable of producing highly accurate results that compel a majority of scientists to implement the approach in cases of predictive analytics. Few works related to wine data have been undertaken using different classifiers, and thus far, no studies have compared the performance metrics of the different classifiers with different feature sets for the prediction of quality among types of wine. In this chapter, an intelligent approach is proposed by considering a recursive feature elimination (RFE) algorithm for feature selection, as well as nonlinear and probabilistic classifiers. Performance metricsincluding accuracy, sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV) are compared by implementing different classifiers with original feature sets (OFS) as well as reduced feature sets (RFS). The results show accuracy ranging from different sets.

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATIONS

- A software necessities requirement is a report of the advanced software system, which positions necessities that remain useful also non-functional as well as to narrate the communication between the operator besides software. It include the usagecircumstances.
- On the basis of resultant software product to do and what not to do and the contract is set between the consumers and vendors by the specification of software requirements and also rational foundation for approximating the prices, endanger and timetable given by the specification . This report gather adequate and appropriate specification that are need for the development of project. Once we have knowledge of the product to be developed then finally we get requirements.

3.1 FUNCTIONAL REQUIREMENTS

1) Collecting the required dataset.
2) Cleaning the collected data according to the needs.
3) Finding the best suited machine learning model for the pre-processed data.
4) Train the chosen ML model by using the train and the test dataset.
5) Test the accuracy to examine the efficiency of trained model.

3.2 NON- FUNCTIONAL REQUIREMENTS

This criterion is a benchmark for analysing and in some ways, defining a system's behaviour. The look of the system is defined by non-functional requirements, whereas the functionality of the system is defined by functional requirements. It is a non-functional requirement to have a software development environment.

- **Reliability**-the competence of a system to do and support its functions in monotonous circumstances as well as antagonistic or unforeseen circumstances.
- **Security**-concerning security or privacy issues nearby use of the creation or protection of the information used or sent by the creation. Describe any operator identity verification

requirement.

- **Usability**-The term usability also mentions to method for purifying ease of use dur ing the design procedure.
- **Interoperability**-that property stating to the ability of diverse system and administrate to gather in their work (inter-operate).

3.3 SYSTEM REQUIREMENTS

Every computer program requires the presence of particular hardware components or other software resources in order to function properly. This is referred to as (computer) requirement specification, and they are typically used as a recommendation or a rigid rule. Most software has two sets of minimum and recommended requirements. System designs tend to get more complex over time as the need for greater CPU power and resources in the most recent applications increases. According to experts in the field, this tendency will modernise existing computer systems more effectively than technological advancements. A second meaning of the term "system requirements" is an extension of the initial analysis and refers to the requirements that must be met when designing a system or subsystem. Usually, a firm starts with a list of operational needs before getting to the specifics of its IT infrastructure

3.3.1 HARDWARE REQUIREMENTS:

System: Pentium i3 Processor.

Hard Disk : 500 GB.

Monitor : 15’’ LED

Input Devices : Keyboard, Mouse

Ram : 4 GB

3.3.2 SOFTWARE REQUIREMENTS:

Operating system : Windows 10.

Coding Language : Python

Web Framework : Flask

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 Overview

System Analysis is the procedure of empathetic the problematic besides its domain. It is a exhaustive revision of the numerous processes achieved by a structure also their associations inside and external the scheme. Exercise, Knowledge besides shared intellect arecompulsory for gathering of applicable info desirable to progress a decent scheme. A decent investigation prototypical would afford the apparatus to recognize the unruly besides similarly the context of the explanation.

The key intentions of investigation is to seizure a comprehensive, uncertain also unswerving image of the requirements of the structure besides what the arrangement essentialdo to gratify the employer requirements as well as necessities. The stipulations are rehabilitated hooked on a "blue print" aimed at structure the structure throughout the structureenterprise stage, which is an collaborating progression. As well as it is the chief stage in affecting after the problematic domain to solution domain province. Altogether of the unambiguous necessities since the criticism style necessity be instigated, as fine as altogether of the disguised desires wanted by the operator.

Analysis consumes remained prepared for instance per the subsequent:

- Scrutinizing entirely the conceivable existing system.
- Firm the routines of the prevailing structure, curb of the existing scheme, obligation aimed on the new scheme.
- Recognized existing scheme explanation, performance then the inadequacy.
- Blue print of the proposed scheme. This embraces structure illustrations also structure collaboration figures.
- Equipped a gradient of reimbursements, which embraces together perceptible besides imperceptible welfares Quantifiable as well as Qualitative.

4.2 SYSTEM DESIGN

4.2.1 SYSTEM ARCHITECTURE

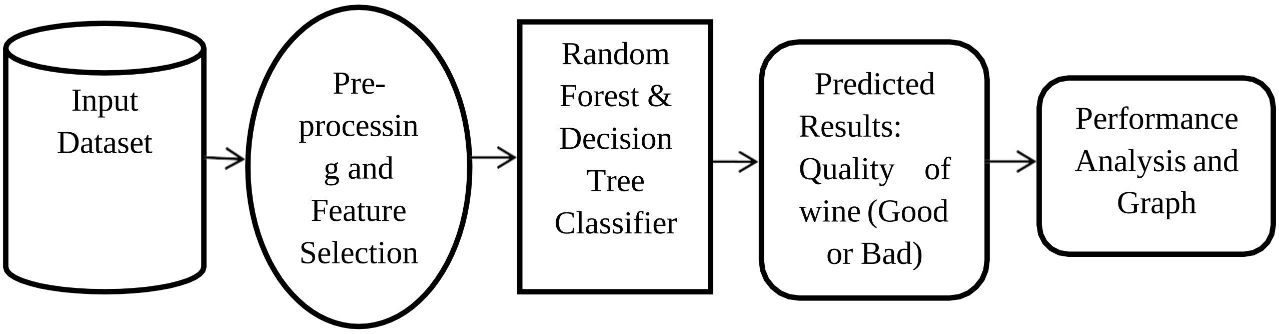

Input Dataset
Pre-processing and Feature Selection
Random Forest & Decision Tree Classifier
Predicted Results: Quality of wine (Good or Bad)
Performance Analysis and Graph

Fig. 4.2 System Architecture

Step 1: Data Collection

The dataset is from the UCI machine learning database repository [https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/]() The original data consist of variants of the Portuguese Vinho-Verde wine and has 1599 observations of Red wine and 4898 observations of White wine. For each, we have the wine quality (scored between 0 and 10) and eleven chemical attributes (quantitative), which are as follows: Fixed acidity, Volatile acidity, Citric acid, Residual sugar, Chlorides, Free sulfur dioxide, Total sulfur dioxide, Density, PH, Sulphates, and Alcohol

Fixed acidity - Most acids involved wine or fixed or nonvolatile.

Volatile acidity - The number of acetic acids in wine which at too high of levels can lead to an unpleasant, vinegar taste

Citric acid - Can be found in small quantities, add freshness and the flavor to the wine.

Residual sugar - The amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1g/L and wines with greater than 45g/L considered as sweet.

Chlorides - The amount of salt in the wine

Free sulfur dioxide - The free form of sulfur dioxide that is not bound to other molecules, and is used to calculate molecular sulfur dioxide

Total sulfur dioxide - The amount of free and bound forms of sulfur dioxide.

Density - The density of water is close to that of water depending on the percent of alcohol and the sugar

PH - Describe how acidic or basic a wine in on a scale from 0 to 14

Sulfates - A wine additive which can contribute to sulfur dioxide gas levels, which act as anantimicrobial and antioxidant

Alcohol - The percent alcohol content of the wine

Step 2: Data Pre-Processing

Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)

Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data

Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis

Split into training and evaluation sets

Step 3: Algorithm

In the model selection module, we used Random Forest Classifier machine learning algorithm. We got an accuracy of 100% on training set so we implemented this algorithm.

The Random Forest Algorithm

Let’s understand the algorithm in layman’s terms. Suppose you want to go on a trip and you would like to travel to a place which you will enjoy.

So what do you do to find a place that you will like? You can search online, read reviews on travel blogs and portals, or you can also ask your friends.

Let’s suppose you have decided to ask your friends, and talked with them about their past travel experience to various places. You will get some recommendations from every friend. Now you have to make a list of those recommended places. Then, you ask them to vote (or select one best place for the trip) from the list of recommended places you made. The place with the highest number of votes will be your final choice for the trip.

In the above decision process, there are two parts. First, asking your friends about their individual travel experience and getting one recommendation out of multiple places they have visited. This part is like using the decision tree algorithm. Here, each friend makes a selection of the places he or she has visited so far.

The second part, after collecting all the recommendations, is the voting procedure for selecting the best place in the list of recommendations. This whole process of getting recommendations from friends and voting on them to find the best place is known as the random forests algorithm.

It technically is an ensemble method (based on the divide-and-conquer approach) of decision trees generated on a randomly split dataset. This collection of decision tree classifiers is also known as the forest. The individual decision trees are generated using an attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and the most popular class is chosen as the final result. In the case of regression, the average of allthe tree outputs is considered as the final result. It is simpler and more powerful compared to the other non-linear classification algorithms.

How does the algorithm work?

<u>It works in four steps:</u>

Select random samples from a given dataset.

Construct a decision tree for each sample and get a prediction result from each decision tree.

Perform a vote for each predicted result.

Select the prediction result with the most votes as the final prediction.

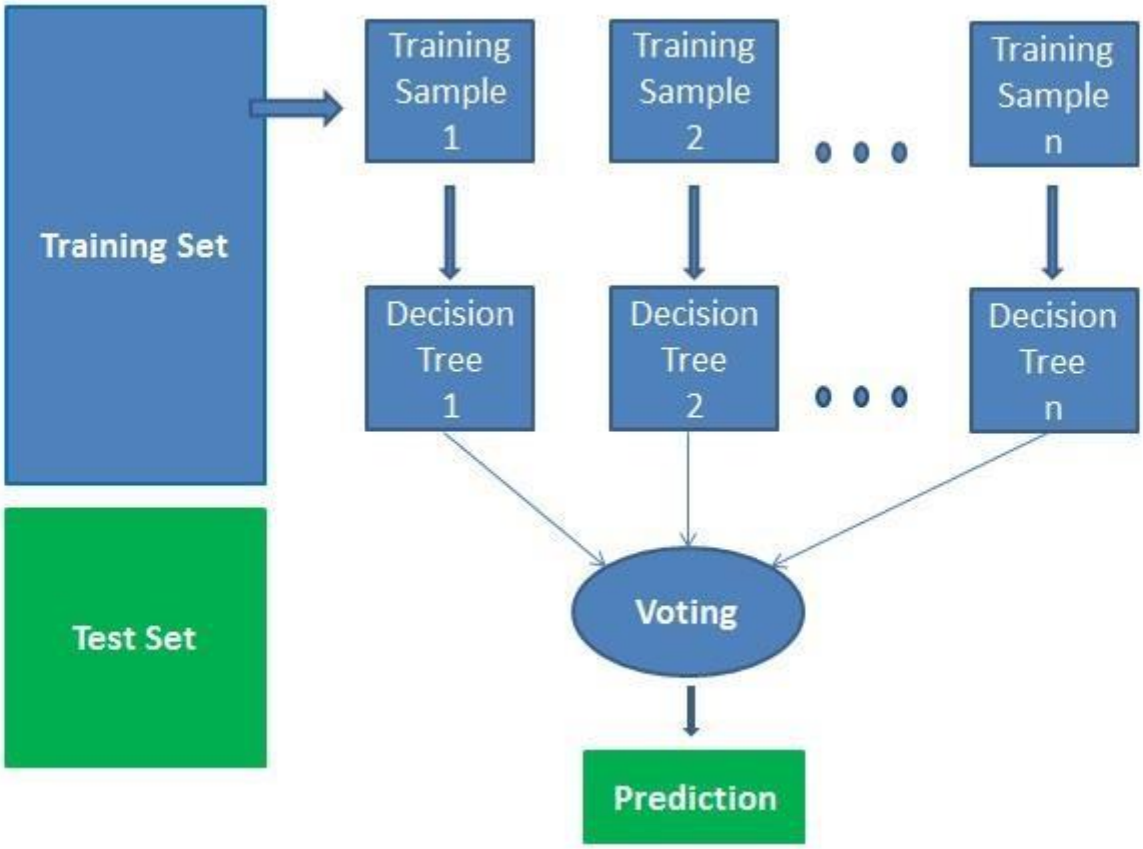


Fig. 4.2.1 Prediction analysis

Finding important features

Random forest also offers a good feature selection indicator. Scikit-learn provides an extra variable with the model, which shows the relative importance or contribution of each feature in the prediction. It automatically computes the relevance score of each feature in the training phase. Then it scales the relevance down so that the sum of all scores is 1.

This score will help you choose the most important features and drop the least important ones for model building.

Random forest uses gini importance or mean decrease in impurity (MDI) to calculate the importance of each feature. Gini importance is also known as the total decrease in node impurity. This is how much the model fit or accuracy decreases when you drop a variable. The larger the decrease, the more significant the variable is. Here, the mean decrease is a significant parameter for variable selection. The Gini index can describe the overall explanatory power of the variables.

Accuracy on test set:

We got an accuracy of 90.6% on test set.

Step 4: Saving the Trained Model

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle.

Make sure you have pickle installed in your environment.

Next, let’s import the module and dump the model into .pkl file.

4.2.2 DATA FLOW DIAGRAM

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is usedto model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system. The figure in the Fig 4.2 represents the DFD of Wine Prediction.

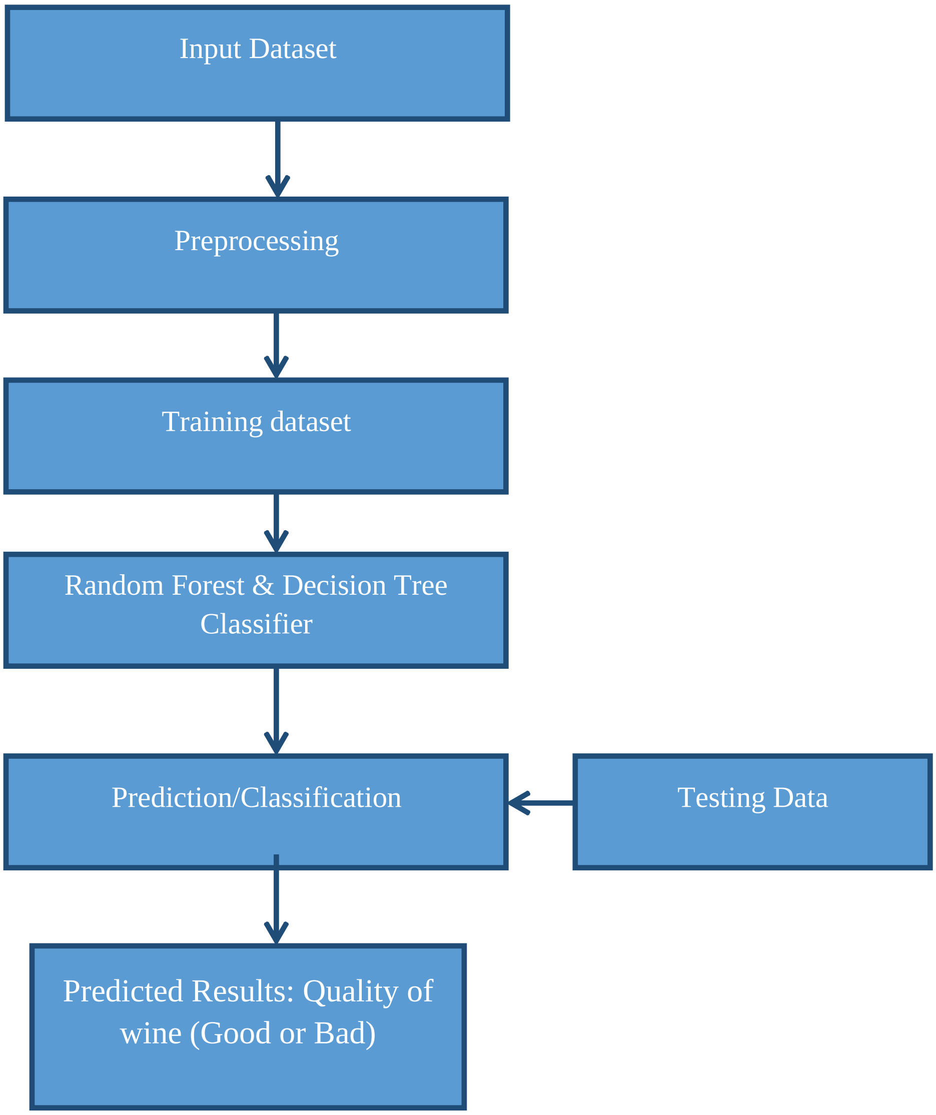

Input Dataset
Preprocessing
Training dataset
Random Forest & Decision Tree Classifier
Prediction/Classification
Testing Data
Predicted Results: Quality of wine (Good or Bad)

Fig. 4.2.2 Data Flow Diagram

- DFD shows how the information moves through the system and how it is modified bya series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

4.2.3 UML DIAGRAMS

- UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standardis managed, and was created by, the Object Management Group.
- The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have provensuccessful in the modeling of large and complex systems.
- The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks,patterns and components.

4.2.4 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphicaloverview of the functionality provided by a system in terms of actors, their goals (representedas use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted as shown in Fig. 4.3

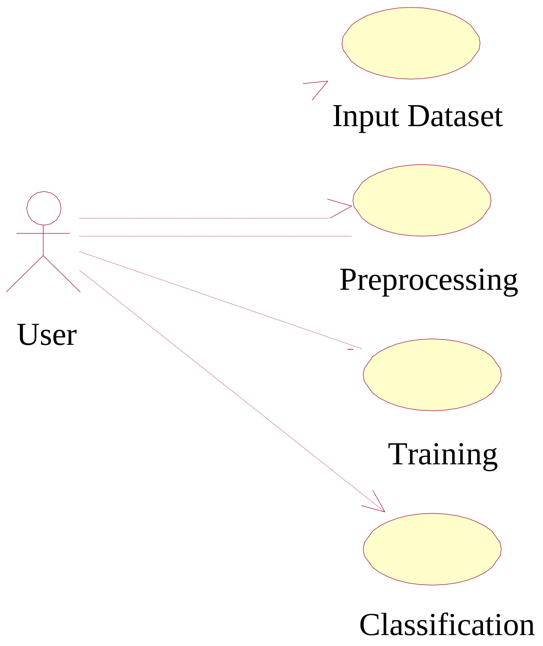


Fig. 4.2.4 Use Case Diagram

4.2.5 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

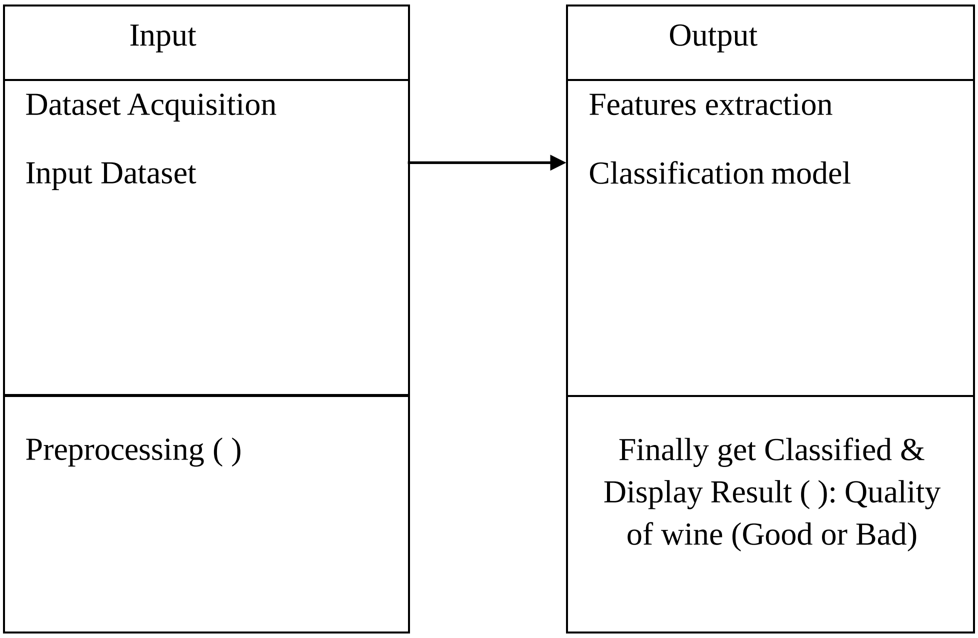


Fig 4.2.5 Class Diagram

4.2.6 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

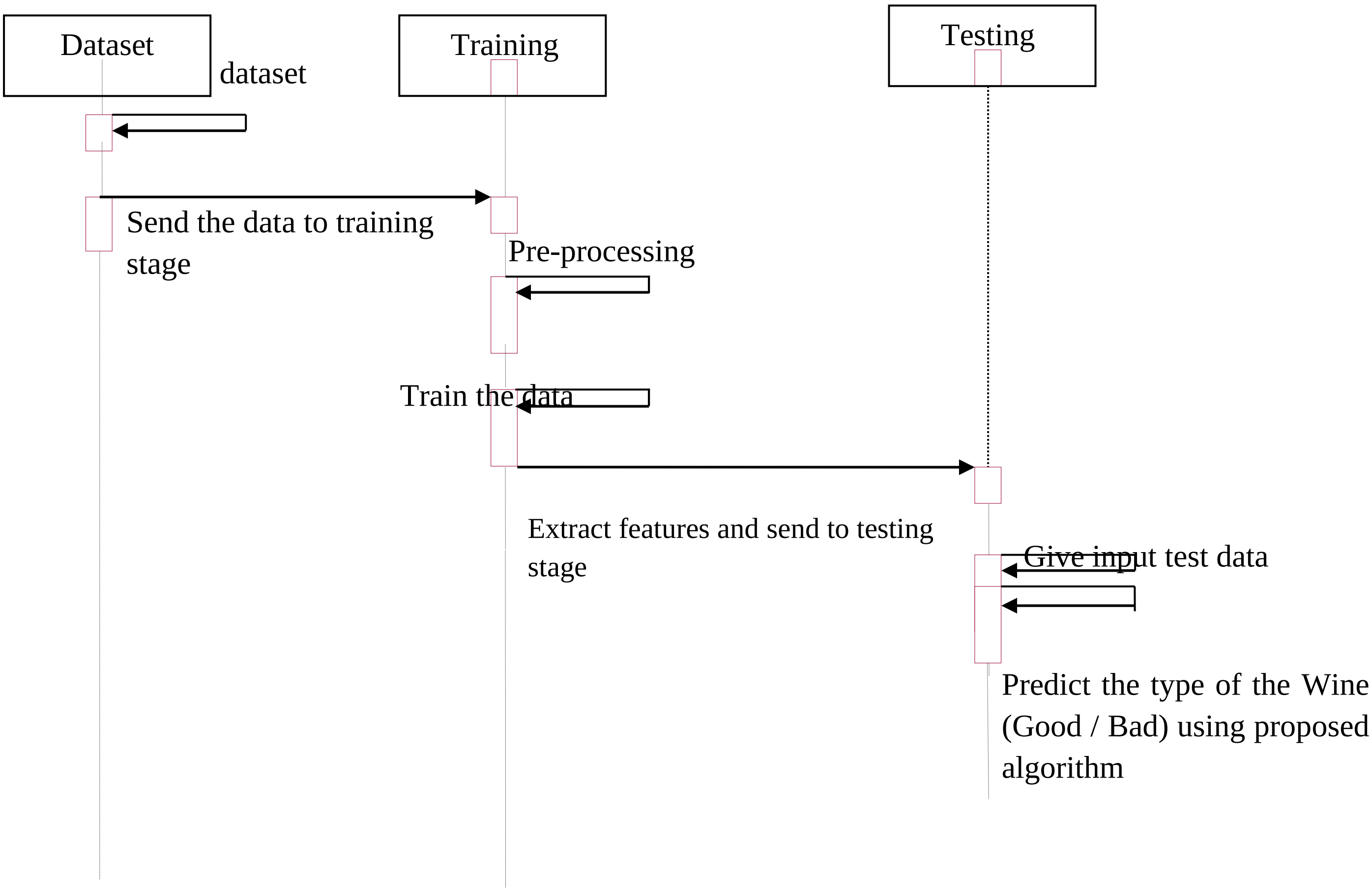

Dataset
Training
Testing
dataset
Send the data to training stage
Pre-processing
Train the data
Extract features and send to testing stage
Give input test data
Predict the type of the Wine (Good / Bad) using proposed algorithm

Fig.4.2.6 Sequence Diagram

4.2.7 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

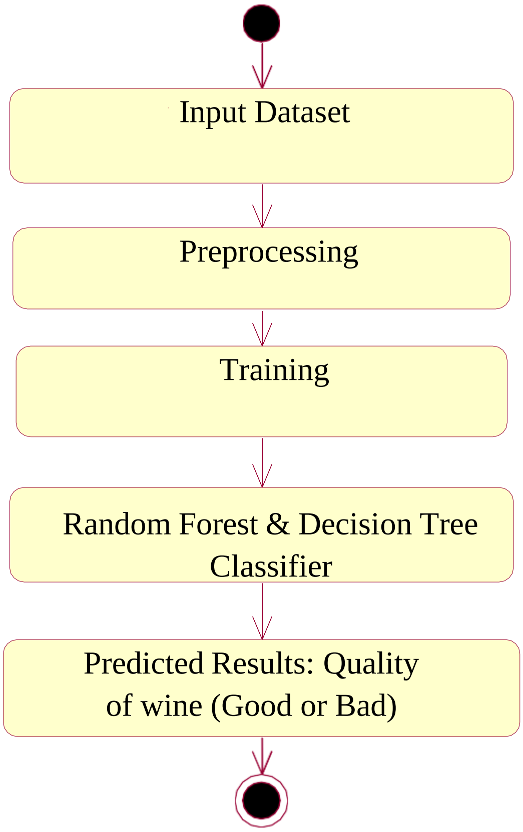

Input Dataset
Preprocessing
Training
Random Forest & Decision Tree Classifier
Predicted Results: Quality of wine (Good or Bad)

Fig.4.2.7 Activity Diagram

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Overview

The application phase of a venture is once the theoretic notion is distorted into a operative scheme, philanthropic operators faith that the novel structure container purpose professionally then efficiently. It involves careful research, study of the present structure then its application restraints, project of change-over approaches, then assessment of change-over approaches. Sideways after preparation, unique of the further most significant features of concocting for placement is operator teaching also exercise.

The additional complex the structure existence applied, the additional time then exertion would remain occupied for system examination as well as project fair to become it active then consecutively.

A direction commission aimed at enactment consumes remained bent, established on the plans of all administration. The grounding of a scheme application strategy is the chief stage in the application procedure. Rendering toward this strategy, calisthenics determination be approved obtainable, conferences around paraphernalia as thriving as capitals determination be detained, as well as supplementary paraphernalia determination be bought in directive to unite the novel system.

The absolute besides furthermost vital phase, the greatest grave phase in attaining a decent novel system then charitable operators faith, is application. It is probable that the novel structure determination be actual. Solitary afterward detailed trying has remained accomplished also it has remained strong-minded that the outline encounters the necessities willpower it be instigated.

System enactment is crafting the novel system attainable aimed at a crew of operators for priming, incessant lug then handling the system on a retro of period aimed at the implementation of maneuvers. In the previous phase, putting of the structure might basis bodily glitches aimed at that vital approaches essential to receipts to instill the punter aimedat the amenity of the structure. Afterward pledging that every then each one meaningful approximately the progression formerly lone lately changed scheme is to creating supplementary. Interpreting progressive scheme to retain scheme transmit then handling the waged of the structure, comprised in the system prominence.

Project productivity stays the pardon essential at attendance is liability is dependable, asylum then unquestionable, is the change amid each one Life series phases also system placement, in a homespun everywhere malfunctions ascend after scheme consume correspondence or notat all consequence on initiative procedure.

It comprises three stages

- **Creation of system execution**, anywhere each phase essential previous aimed at truthfully performing application region component achieved, by way of well as per research of every the assemblage atmosphere then to the backer societies.
- **Deploy System,** where the comprehensive ground work preparation is industrialised through Scheme chic then changed through subsequent phases of life cycle remains applied then confirmed.
- **Move towards activity group,** after collection, proceeds upkeep then gross concluded the utilization unit Area is loosened fragment inside commotion connotation.

5.2 MODULES:

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Accuracy on test set
- Saving the Trained Model

5.2.1 MODULES DESCSRIPTION:

Data Collection:

Data collection process is the first module of the project and this is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.

There are several techniques to collect the data, like web scraping, manual interventions. Our dataset is placed in the project and its located in the model folder. The dataset is referred from the popular standard dataset repository kaggle where all the researchers refer it. The following is the dataset link.

KaggleDatasetLink:

https://www.kaggle.com/datasets/jayaprakashpondy/wine-quality-dataset

Dataset:

The dataset consists of 6497 individual data. There are 13 columns in the dataset, which are described below.

fixed_acidity: Fixed acidity value in wine

volatile_acidity: volatile acidity value in wine

citric_acid: citric acid value in wine

residual_sugar : Residual sugar value in wine

chlorides: Chlorides value in wine

free_sulfur_dioxide : Free sulfur dioxide value in wine

total_sulfur_dioxide : Total sulfur dioxide value in wine

density: Density value in wine

pH: Ph value in wine
sulphates: sulphates value in wine

alcohol: alcohol value in wine

quality: Quality of wine(good or

bad)style: red or white

Data Preparation:

Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)

Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data

Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis

Split into training and evaluation sets

Model Selection:

In the model selection module, we used Random Forest Classifier machine learning algorithm. We got an accuracy of 100% on training set so we implemented this algorithm.

The Random Forest Algorithm

Let’s understand the algorithm in layman’s terms. Suppose you want to go on a trip and you would like to travel to a place which you will enjoy.

So what do you do to find a place that you will like? You can search online, read reviews on travel blogs and portals, or you can also ask your friends.

Let’s suppose you have decided to ask your friends, and talked with them about their past travel experience to various places. You will get some recommendations from every friend. Now you have to make a list of those recommended places. Then, you ask them to vote (or select one best place for the trip) from the list of recommended places you made. The place with the highest number of votes will be your final choice for the trip.

In the above decision process, there are two parts. First, asking your friends about their individual travel experience and getting one recommendation out of multiple places they have visited. This part is like using the decision tree algorithm. Here, each friend makes a selection of the places he or she has visited so far.

The second part, after collecting all the recommendations, is the voting procedure for selecting the best place in the list of recommendations. This whole process of getting recommendations from friends and voting on them to find the best place is known as the random forests algorithm.

It technically is an ensemble method (based on the divide-and-conquer approach) of decision trees generated on a randomly split dataset. This collection of decision tree classifiers is also known as the forest. The individual decision trees are generated using an attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and the most popular class is chosen as the final result. In the case of regression, the average of allthe tree outputs is considered as the final result. It is simpler and more powerful compared to the other non-linear classification algorithms.

How does the algorithm work?

It works in four steps:

Select random samples from a given dataset.

Construct a decision tree for each sample and get a prediction result from each decision tree.

Perform a vote for each predicted result.

Select the prediction result with the most votes as the final prediction.

The Decision Tree Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to aproblem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm, which stands for Classificationand Regression Tree algorithm.

A decision tree simply asks a question, and based on the answer (Yes/No), it furthersplit the tree into subtrees.

Below diagram explains the general structure of a decision tree:

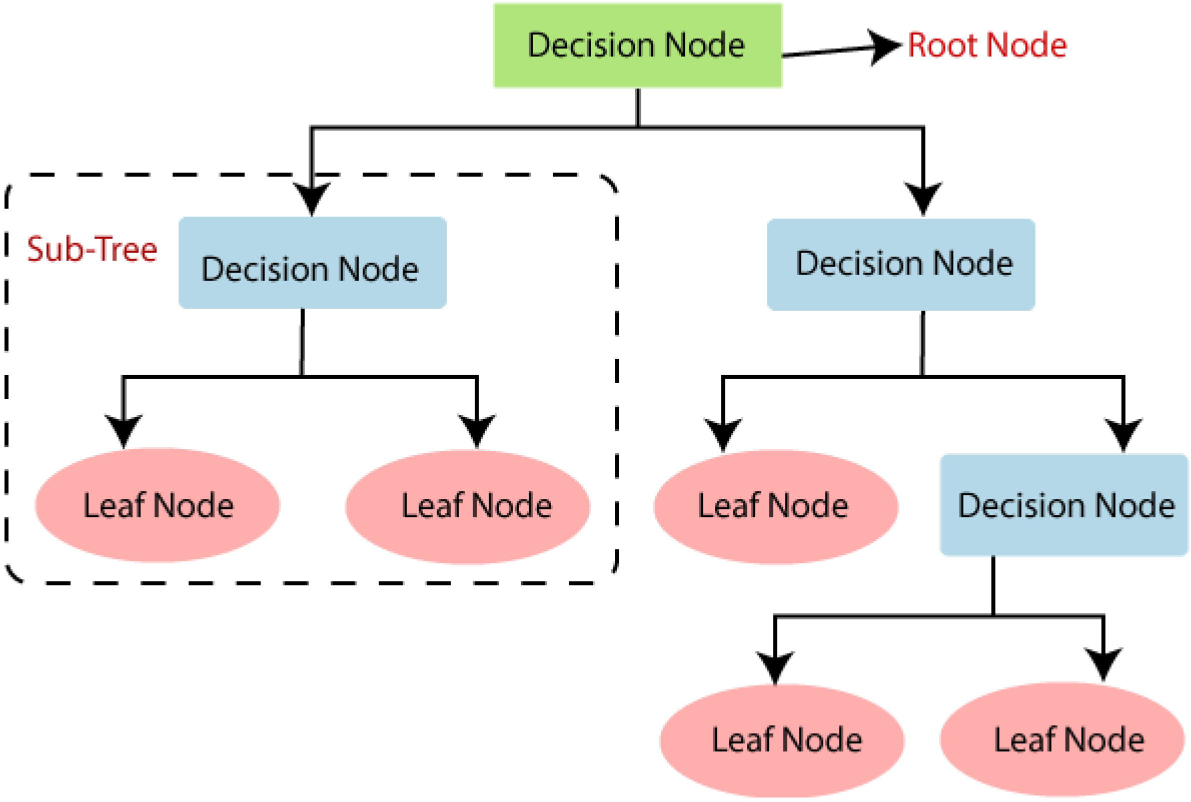


Fig. 5.2.1 Decision tree

Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.**Step-2**:

Find the best attribute in the dataset using Attribute Selection Measure (ASM).**Step-3**: Divide

the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step - 3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

5.3 Psuedo code:

```
import pandas as pd
import matplotlib pyplot as pltfrom
functools import reduce
df = pd.read_csv('wine_dataset.csv',encoding= 'unicode_escape')df
df.dtypes df.isnull().sum()
df.info() df['quality'].unique()
import seaborn as sns
plt.style.use('seaborn')
df['quality'].hist(bins=20)df
df['style'].value_counts()
plt.figure(figsize=(12,8))
sns.heatmap(df.corr(), annot=True, cmap='Dark2_r', linewidths = 2)
df.isnull().sum()
df['quality'] = ['Good' if x>=6 else 'Bad' for x in df['quality']]
df['quality'].unique()
df['quality'].value_counts()df
df['style'].unique() df.loc[df['style']=='red','style']
= 1
df.loc[df['style']=='white','style'] = 2df
df.dtypes
df['style'].astype(int)
df.dtypes
from sklearn.model_selection import train_test_splitfrom
sklearn.preprocessing import RobustScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

```
df['quality'].unique()
n = df['quality'].value_counts()['Good']
df_majority = df[df['quality']== 'Good']
df_minority = df[df['quality']=='Bad'] from
sklearn.utils import resample
df_minority_upsampled = resample(df_minority,replace=True,n_samples =
n,random_state=42)

df = pd.concat([df_majority,df_minority_upsampled])
df['quality'].value_counts()
X = df.drop(['quality'], axis=1).values

y = df['quality'].values
X.shape
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.20,
random_state=101)
rfc = RandomForestClassifier(n_estimators=60, random_state=23)
rfc.fit(X_train,y_train)
rfc.score(X_train,y_train)
from sklearn.metrics import accuracy_scorey_pred
= rfc.predict(X_test )
rfacc = accuracy_score(y_pred,y_test)
from sklearn.metrics import accuracy_scorey_pred
= rfc.predict(X_test )
rfacc = accuracy_score(y_pred,y_test)rfacc *
100
from sklearn import tree
dt = tree.DecisionTreeClassifier()dt =
dt.fit(X_train,y_train)
dt.score(X_train,y_train)
y_pred = dt.predict(X_test )
dtacc = accuracy_score(y_pred,y_test)dtacc *
100
plt.bar(['Random Forest', 'Decision Tree'],[rfacc,dtacc])
plt.xlabel("Algorithms")
plt.ylabel("Accuracy")
plt.show()
import sklearn.metrics
print(sklearn.metrics.classification_report(y_test, y_pred))y_pred
= rfc.predict(X_test )
y_true=y_test


from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_true,y_pred)
cm
import seaborn as sns
import matplotlib.pyplot as plt

f, ax=plt.subplots(figsize=(5,5))
sns.heatmap(cm,annot=True,linewidths=0.5,linecolor="red",fmt=".0f",ax=ax)
plt.xlabel("y_pred")
plt.ylabel("y_true")
plt.show()
import pickle

pickle.dump(rfc,open('wine.pkl','wb')) wine =
pickle.load(open('wine.pkl','rb'))
```

CHAPTER 6

SYSTEM TESTING

6.1 Overview

Software analysis remains a vital chunk of software superiority assertion besides remains the last stage cutting-edge the requirement plan also coding procedure. It proposals a roadmap aimed at the designer, the excellence pledge activity, as well as the customer, a roadmap that summaries the phases to be occupied as portion of the challenging trail, when these phases are deliberate then formerly implemented, in addition in what way abundant exertion, time, also possessions are wanted. Testing frequently devours 30 to 40% of softwareexpansion project's inclusive exertion. Testing demonstrations that agenda purposes appear to be operative as predictable also that performance standards appear to devour existed meet. Also, documents met through testing stretches the strong sign of software-eminence. Itcanister solitary exhibit the attendance of software mistakes.

6.2 Types of Tests

Functional testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted astwo distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. Testing is event driven and is more concerned with the basic outcome of screens or fields. The task of the integration test is to check that components or software applications,
e.g. software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

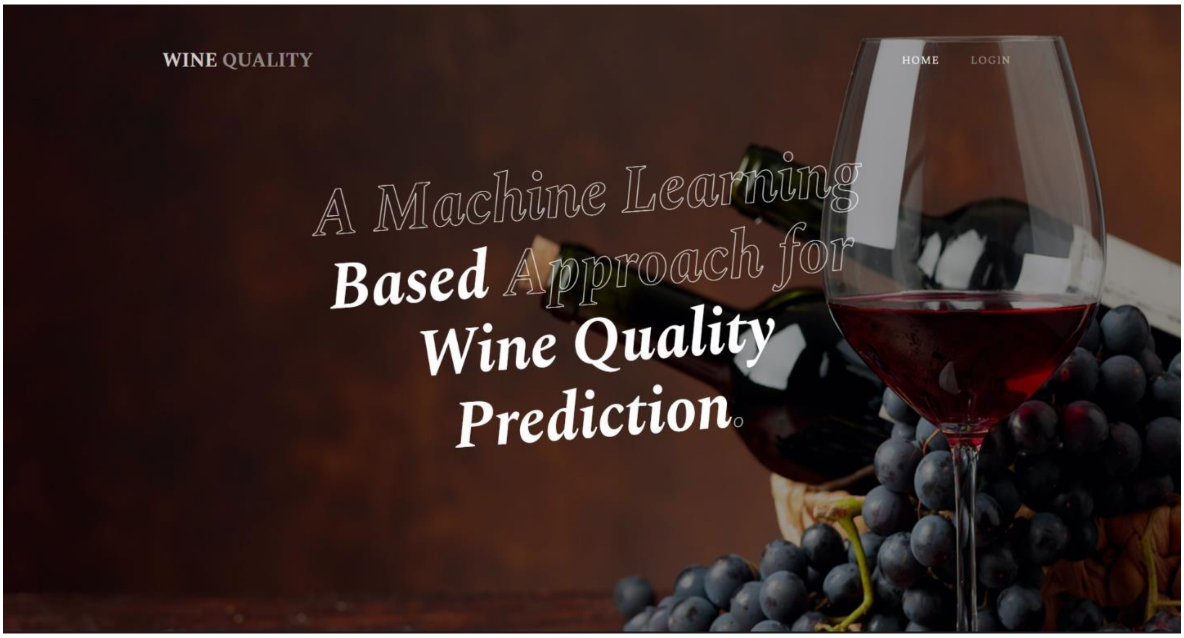
6.3 TEST CASES

Test Case	Input Description	Test Condition	Expected Outcome	Actual Result	Pass/ Fail
#01	Enter Wine Feature data and submit	Check for data	Uploaded Successfully	Data is uploaded successfully	Pass
#02	Enter Wine Feature data and submit	Check for data	Uploaded Successfully	Improper data input given shows error.	Fail
#03	Feature extraction is done for pre-processed data	Check for feature extraction	Feature extraction data is complete	As Expected	Pass
#04	Training Model using Random Forest & Decision Tree	Check for model loss and accuracy	Model trained Successfully	Model trained Successfully	Pass
#05	Detect the given input	Check for classification with accuracy	Predict result	Result will displayed	Pass

CHAPTER 7

RESULTS AND DISCUSSION

SNAPSHOTS



A Machine Learning Based Approach for Wine Quality Prediction.

Fig. 7.1 Home Page

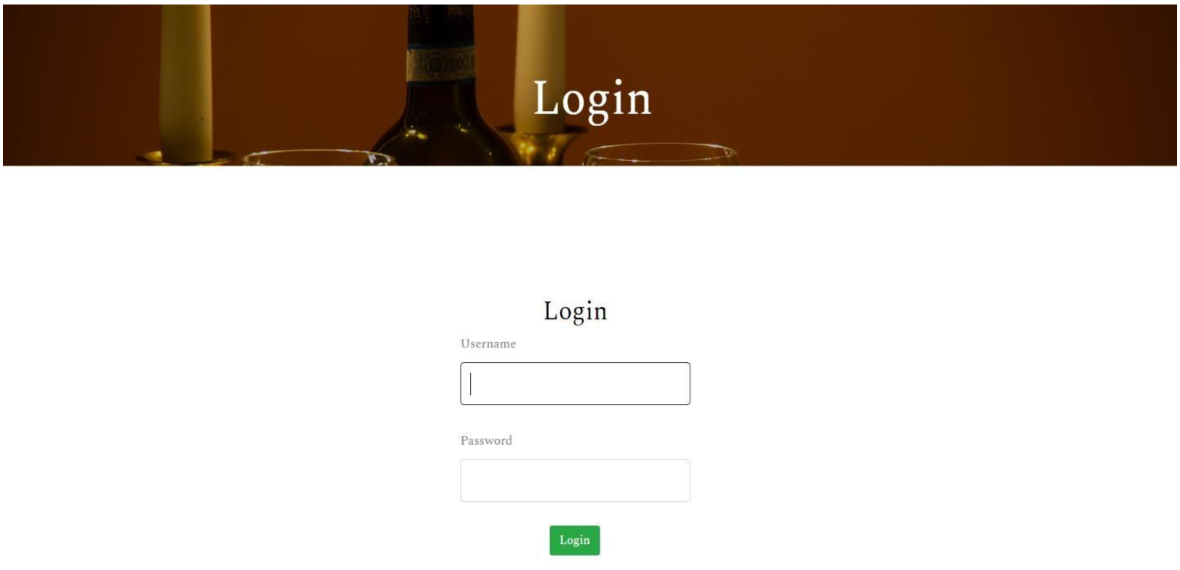
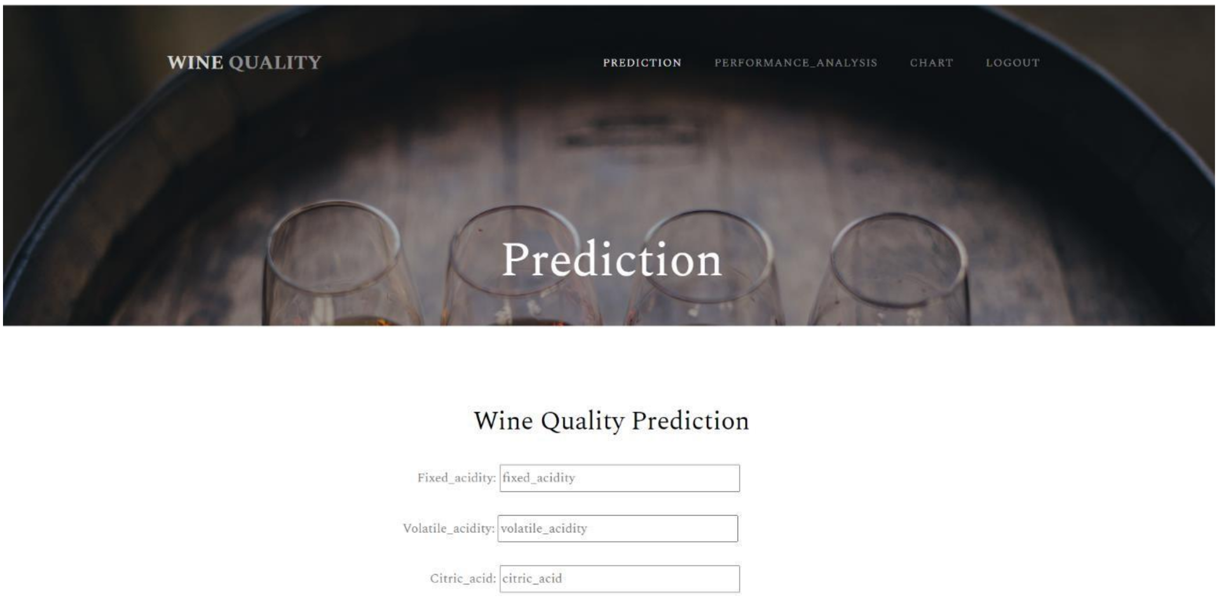


Fig. 7.2 Login Page



Prediction

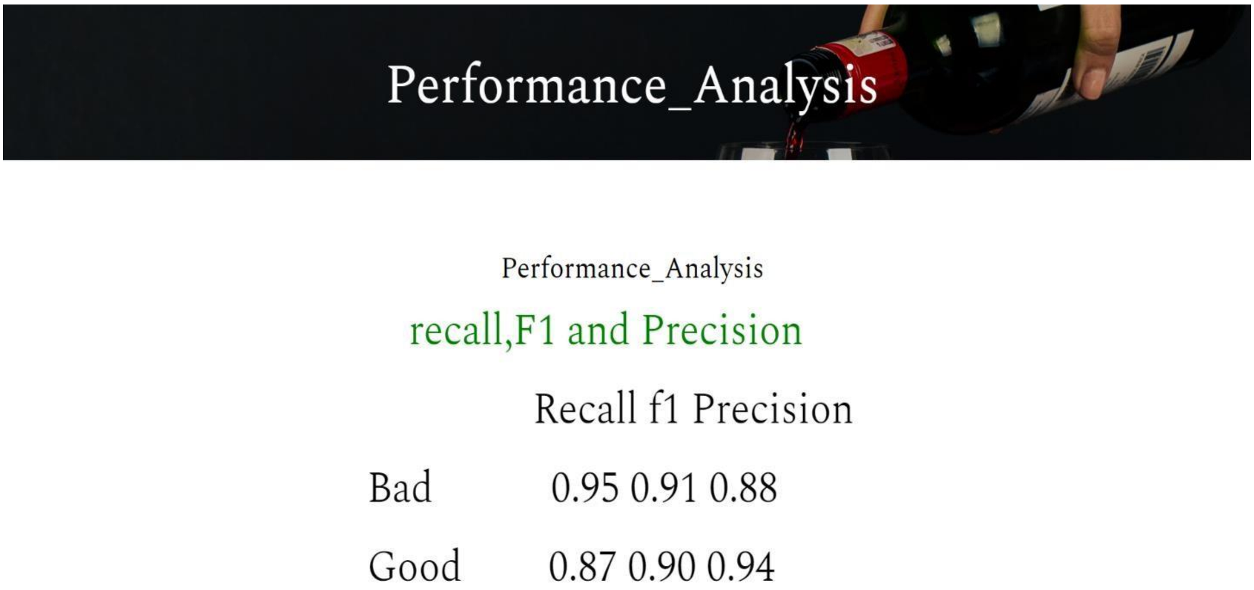
Wine Quality Prediction

Fixed_acidity: fixed_acidity

Volatile_acidity: volatile_acidity

Citric_acid: citric_acid

Fig. 7.3 Prediction Quality



Performance_Analysis

Performance_Analysis

recall,F1 and Precision

	Recall	f1	Precision
Bad	0.95	0.91	0.88
Good	0.87	0.90	0.94

Fig. 7.4 Performance Analysis

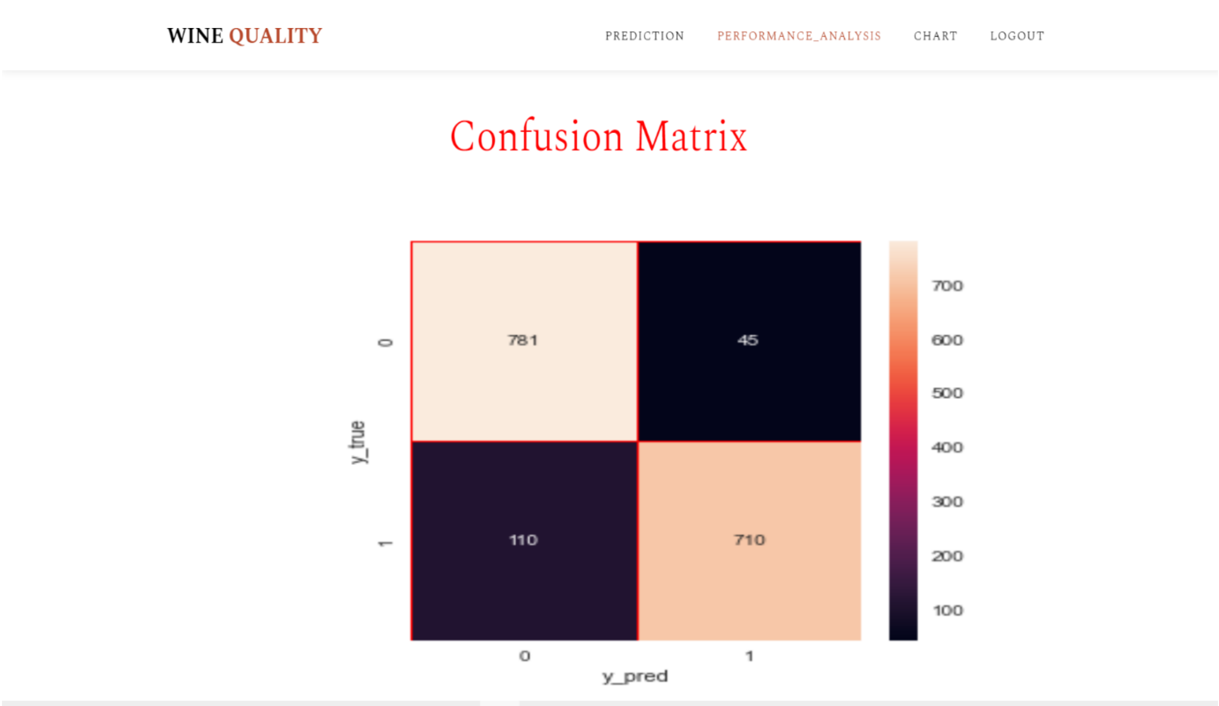


Fig. 7.5 Confusion Matrix

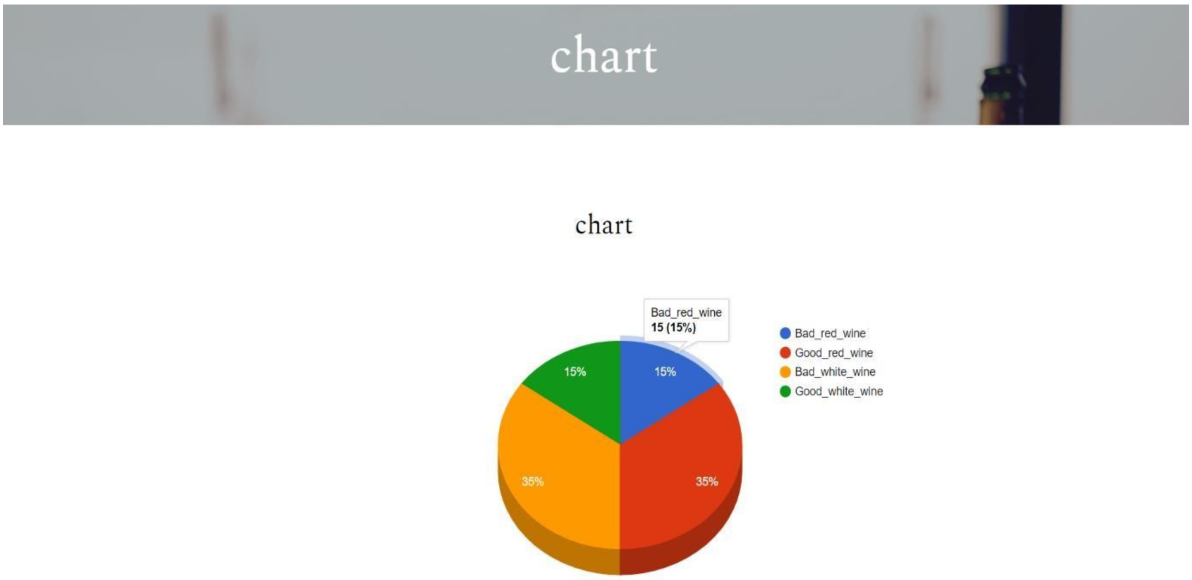
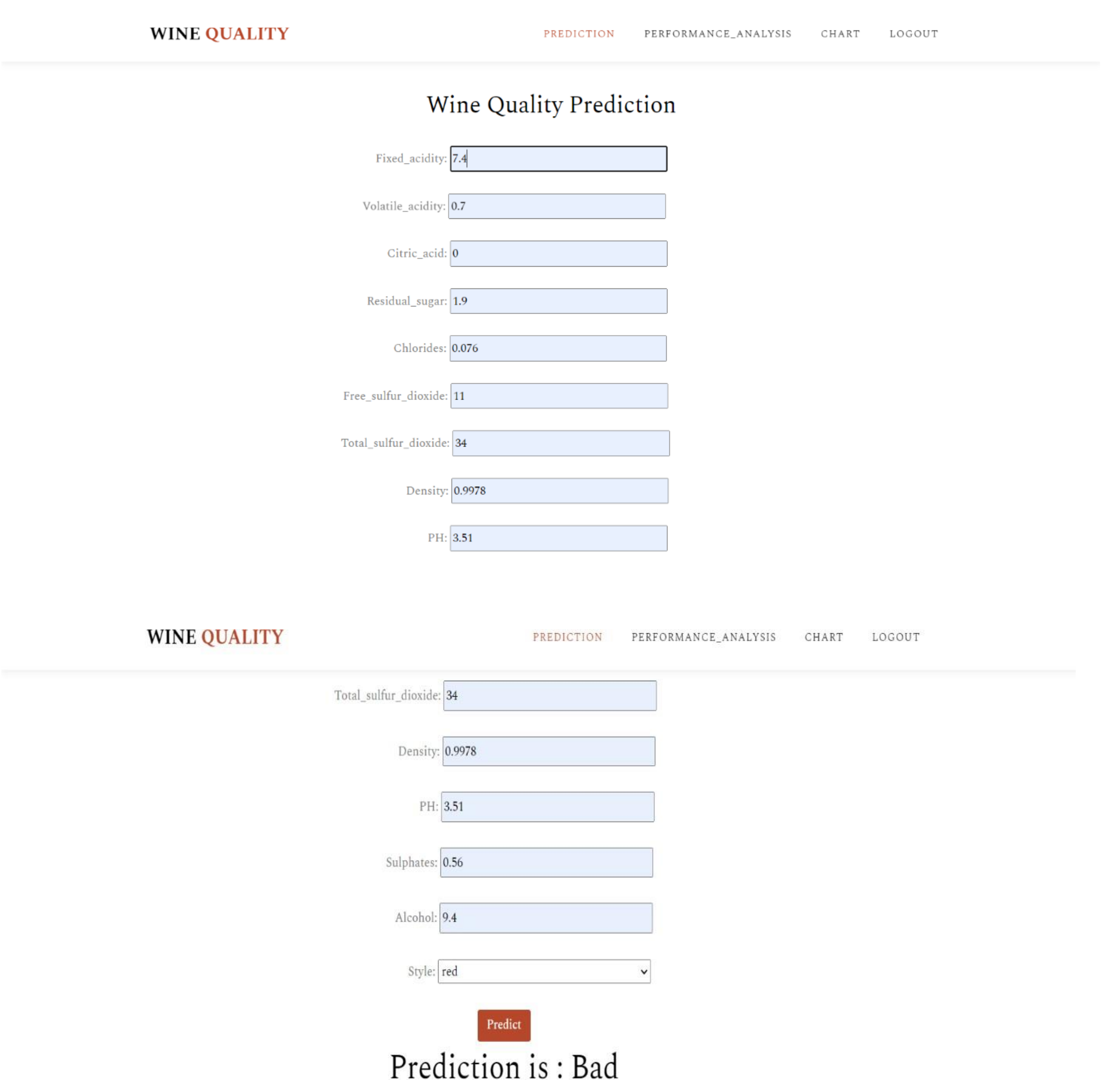


Fig. 7.6 Chart



Wine Quality Prediction

Fixed_acidity: 7.4

Volatile_acidity: 0.7

Citric_acid: 0

Residual_sugar: 1.9

Chlorides: 0.076

Free_sulfur_dioxide: 11

Total_sulfur_dioxide: 34

Density: 0.9978

PH: 3.51

Total_sulfur_dioxide: 34

Density: 0.9978

PH: 3.51

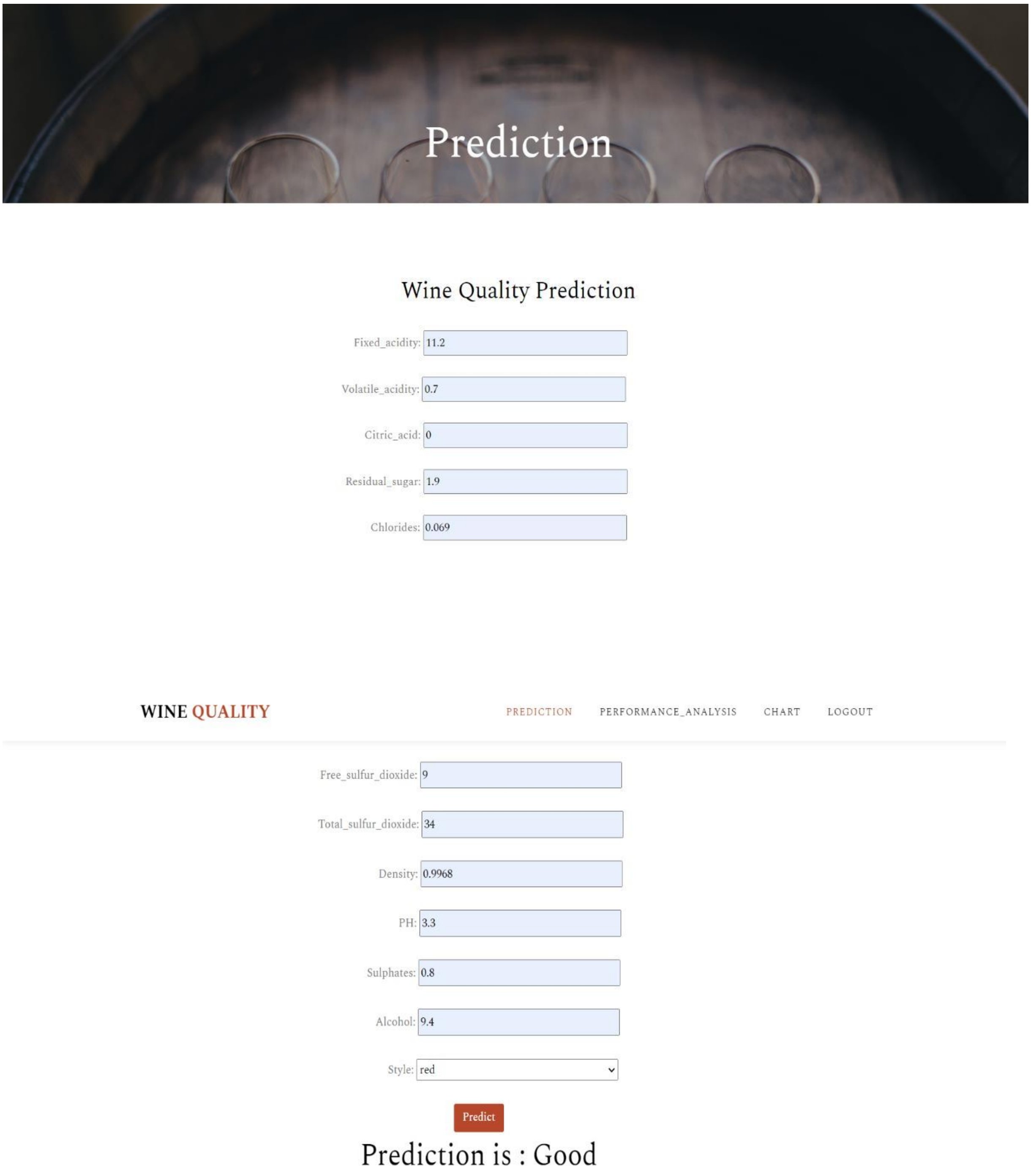
Sulphates: 0.56

Alcohol: 9.4

Style: red

Prediction is : Bad

Fig. 7.7 Quality analysis



Wine Quality Prediction

Fixed_acidity: 11.2

Volatile_acidity: 0.7

Citric_acid: 0

Residual_sugar: 1.9

Chlorides: 0.069

Free_sulfur_dioxide: 9

Total_sulfur_dioxide: 34

Density: 0.9968

PH: 3.3

Sulphates: 0.8

Alcohol: 9.4

Style: red

Prediction is : Good

Fig. 7.8 Prediction Style

CONCLUSION

The work carried out involves deployment of Random Forest Classifier for prediction of Wine quality. The Train Accuracy achieved by our proposed system is 100% and the test Accuracy achieved by our proposed system is 90% which is better as compared to the existing system. The experiment illustrates that the values of dependent variable such as volatile acidity, residual sugar, SO2 and citric acid can be predicted more accurately when only significant factors.

FUTURE ENHANCEMENTS

In the future, to improve the accuracy of the classifier, it is clear that the algorithm or the data must be adjusted. We recommend feature engineering, using potential relationships between wine quality, or applying the boosting algorithm on the more accurate method.

In addition, by applying the other performance measurement and other machine learning algorithms for the better comparison on results. This study will help the manufacturing industries to predict the quality of the different types of wines based on certain features, and also it will be helpful for them to make a good product.

REFERENCES

[1] Piyush Bhardwaj , Parul Tiwari , Kenneth Olejar Jr , Wendy Parr and Don Kulasiri, A machine learning application in wine quality prediction, Journal of Machine Learning with Applications,(2022), Vol. 08, PP 34-38.

[2] K. R. Dahal, J. N. Dahal, H. Banjade and S. Gaire, Prediction of Wine Quality Using Machine Learning Algorithm, in Open Journal of Statistics (2021),Vol.11 PP 278-289 .

[3] Shaw, B., Suman, A.K. and Chakraborty, B., Wine Quality Analysis Using Machine Learning, Book series of Emerging Technology in Modelling and Graphics. Advances in Intelligent Systems and Computing, Springer,Singapore 2019 ,Vol 937,PP 239-247

[4] Mahima, Ujjawal Gupta and Yatindra Patidar, Wine Quality Analysis Using Machine Learning Algorithms, Micro-Electronics and Telecommunication Engineering Proceedings of 3rd ICMETE 2019,Springer.,Vol 106, PP11- 18,Lecture Notes in Networks and Systems book series.

[5] Satyabrata Aich, Mangal Sain and Jin-Han Yoon, Prediction of Different Types of Wine Using Nonlinear and Probabilistic Classifiers. In Integrated Intelligent Computing, Communication and Security, (2019), Vol.771, PP 11- 19,Studies in Computational Intelligence book series SCI, Springer, Singapore.

[6] Yogesh Gupta, Selection of important features and predicting wine quality using machine learning techniques, Elsevier Science Direct, Procedia Computer Science, 2018, Vol. 125, PP 305-312.

[7] Alexan A. Khalafyan, Zaual A. Temerdashev, Vera A. Akin'shina and Yuri F. Yakuba, Data on the sensory evaluation of the dry red and white wines quality obtained by traditional technologies from European and hybrid grape varieties in the Krasnodar Territory, Russia, Journal of ELSEVIER, Data in Brief, Vol. 36, 2021, 106992, ISSN 2352-3409.

[8] Devika Pawar, Aakanksha Mahajan and Sachin Bhoithe, Wine QualityPrediction using Machine Learning Algorithms, Journal of Computer Applications Technology and Research, 2019, Vol. 08, –Issue 09, PP 385-388.

[9] María-Pilar Sáenz-Navajas , Jordi Ballester , Purification Fernández- Zurbano , Vicente Ferreira , Dominique Peyron and Dominique Valentin Wine Quality Perception: A Sensory Point of View, Journal of Wine Safety, ,Consumer Preference, and Human Health Springer,2016, PP 119–138

[10] Nikita Sharma ,Quality Prediction of Red Wine based on Different Feature Sets Using Machine Learning Techniques, International Journal of Science and Research (IJSR), July 2020 ,Volume 9, Issue 7, PP 1358-1366.

[11] Gaurang S Patkar and D. Balaganesh, Smart Agri Wine: An Artificial Intelligence Approach to Predict Wine Quality, Journal of Computer Science, Volume 17 No. 11,2021, PP 1099-1103.

[12] Anurag Sinha1 and Atul Kumar, Wine Quality and Taste Classification Using Machine Learning Model, International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE) 2020,Volume 4, Issue 4,PP 715- 721.

[13] B. Chen, C. Rhodes, A. Crawford, and L. Hambuchen, Wine informatics: applying data mining on wine sensory reviews processed by the computational wine wheel IEEE Conference on Data Mining Workshop,DEC 2014, PP 142- 149.