A

MINI PROJECT REPORT

on

WINE QUALITY PREDICTION BY USING MACHINE LEARNING ALGORITHMS

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

(MP)

Pinnapureddy Dileep : 177Y1A1209

Under the Guidance

of

Mr.P.RAMESH BABU

Assistant 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)

December 2020

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 Section2(f) & 12(B)of the UGC act,1956

CERTIFICATE

This is to certify that the project report titled "**Wine Quality Prediction by using Machine Learning Algorithms"** is being submitted by **Pinnapureddy Dileep (177Y1A1209)** in IV B.Tech II Semester **Information Technology** is a record bonafide work carried out by him. 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 Section2(f) & 12(B)of the UGC act,1956

DECLARATION

I hereby declare that the Mini Project Report entitled, "**Wine Quality Prediction by using Machine Learning Algorithms"** 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:

Pinnapureddy Dileep

(177Y1A1209)

ACKNOWLEDGEMENT

I am happy to express my deep sense of gratitude to the principal of the college **Dr. K. Venkateswara Reddy,** Professor, Department of Information Technology, Marri Laxman Reddy Institute of Technology & Management, for having provided me with adequate facilities to pursue my project.

I would like to thank **Mrs. Nagalakshmi ,** Assoc. Professor and Head, Department of Information Technology, 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.

I am very grateful to my project guide **Mr. P.Ramesh Babu,** Asst. Prof**.**, Department of Information Technology, Marri Laxman Reddy Institute of Technology & Management, for his extensive patience and guidance throughout my project work.

I sincerely thank my 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.

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

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

CONTENTS

ABSTRACT

Wine classification is a difficult task since taste is the least understood of the human senses. A good wine quality prediction can be very useful in the certification phase, since currently the sensory analysis is performed by human tasters, being clearly a subjective approach. An automatic predictive system can be integrated into a decision support system, helping the speed and quality of the performance. Furthermore, a feature selection process can help to analyze the impact of the analytical tests. If it is concluded that several input variables are highly relevant to predict the wine quality, since in the production process some variables can be controlled, this information can be used to improve the wine quality. The Wine quality prediction system sets up an automated prediction system based on the attributes that are listed in the dataset.The dataset mainly consists of the ingredient components that are present in the wine. Using machine learning algorithms the automated prediction system is designed. Classification models used here are

1) Random Forest

2) Stochastic Gradient Descent

3) SVC

4) Logistic Regression.

LIST OF FIGURES

SYMBOLS & ABBREVIATIONS

ML	**:**	Machine Learning
SVC	**:**	Support Vector Classifier
RF	**:**	Random Forest
SVM	**:**	Support Vector Machine
AI	**:**	Artificial Intelligence
UML	**:**	Unified Modeling Language

1.INTRODUCTION

The aim of this project is to predict the quality of wine on a scale of 0–10 given a set of features as inputs. The dataset used is Wine Quality Data set from UCI Machine Learning Repository. Input variables are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulphur dioxide, total sulphur dioxide, density, pH, sulphates, alcohol. And the output variable is quality (score between 0 and 10).We are dealing only with red wine. We have quality being one of these values: [3, 4, 5, 6, 7, 8]. The higher the value the better the quality. In this project we will treat each class of the wine separately and their aim is to be able and find decision boundaries that work well for new unseen data. These are the classifiers.

In this paper we are explaining the steps we followed to build our models for predicting the quality of red wine in a simple non-technical way. We are dealing only with red wine. We would follow similar process for white wine or we could even mix them together and include a binary attribute red/white, but our domain knowledge about wines suggests that we shouldn't. Classification is used to classify the wine as good or bad. Before examining the data it is often referred to as supervised learning because the classes are determined.

1.1 MOTIVATION

Red wine variant of the Portuguese "Vinho Verde" wine refers to Portuguese wine that originated in the historic Minho province in the far north of the country. The main goal of this problem is to find which features of these kinds of wine are the ones that provide most information about its quality. We will also try to make a prediction of a wine's quality and check if it matches with the real quality. Although this dataset can be viewed as a classification (multiclass classification) or a regression problem, we will solve it using regression techniques.

1.2 PROBLEM

In industries, understanding the demands of wine safety testing can be a complex task for the laboratory with numerous analytes and residues to monitor. But, our application's prediction, provide ideal solutions for the analysis of wine, which will make this whole process efficient and cheaper with less human interaction.

1.3SOLUTION

An automatic predictive system can be integrated into a decision support system, helping the speed and quality of the performance. Furthermore, a feature selection process can help to analyze the impact of the analytical tests. If it is concluded that several input variables are highly relevant to predict the wine quality, since in the production process some variables can be controlled, this information can be used to improve the wine quality. Classification models used here are 1) Random Forest 2) Stochastic Gradient Descent 3) SVC 4)Logistic Regression

1.4 SCOPE

The outcomes in this research are based on results that involve only sample datasets. It is necessary that additional datasets should be considered for the evaluation of different classification problems as the information growth in the recent technology is extending to heights beyond assumptions. Recent field of technology is growing and data are by nature dynamic. Hence, further classification of the entire system needs to be implemented right from the scratch since the results from the old process have become obsolete. The scope of future work can deal with Incremental learning, which stores the existing model and processes the new incoming data more efficiently.

1.5 PROBLEM DEFINITION

The Wine quality prediction system sets up an automated prediction system based on the attributes that are listed in the dataset.The dataset mainly consists of the ingredient components that are present in the wine.Using machine learning algorithms the automated prediction system is designed.

1.6 OBJECTIVE

- Our main objective is to predict the wine quality using machine learning through Python programming language.
- A large dataset is considered and wine quality is modelled to analyse the quality of wine through different parameters like fixed acidity, volatile acidity etc.
- All these parameters will be analysed through Machine Learning algorithms like, random forest classifier, Stochastic Gradient Descent, SVC, Logistic Regression algorithms which will helps to rate the wine on scale 1 - 10 or bad - good. Output obtained would further be checked for correctness and model will be optimized accordingly.
- It can support the wine expert evaluations and ultimately improve the production.

1.7 LIMITATIONS

The machine learning models used in this research aren't able to predict red wine quality with high accuracy, specificity and sensitivity. This is partially explained by the low prevalence of quality levels 3, 4 and 8 and the large distribution overlapping area stratified by quality.

Besides this, the lack of information about how the dataset was created may impact the prediction of quality using the physicochemical properties as predictors. Such examples are the composition of grape varieties in each wine, the mix of experts that evaluated wine quality, or the production year.

1.8 ORGANIZATION OF DOCUMENT

1 describes motivation, scope, objective and problem definition are discussed. 2 explains literature survey and basic concepts and terms. 3 discusses about user requirements, software and hardware requirements. 4 discuss about module design . In 5 we discuss about system implementation like method of implementation and experimental results. In 6 we discuss about testing and validation. In 7 we discuss about conclusion and future enhancements of the project and then references.

2. LITERATURE SURVEY

2.1 OVERVIEW

My automated predicted system provides accurate prediction values ,accurate training ,testing set values that impacts in the getting overall accurate predicted values of the wine.

2.2 EXISTING SYSTEM

The existing wine quality prediction implementations are showing not accurate values for predicting the wine quality .In such prediction implementation the use of machine learning algorithms are different .The prediction is done based on training and testing of dataset values ,in the old existing system the accuracy of training and testing data is low.

2.3 DISADVANTAGES OF EXISTING SYSTEM

- Existing system provide low prediction values.
- The accuracy of testing and training of data is low.
- The use of different machine learning algorithms.
- The separation of dependent and independent variables is not well formed.
- The data set prediction score is poor.

2.4 PROPOSED SYSTEM

To overcome the limitations of existing system, It gives insights of the dependency of target variables on independent variables using machine learning techniques to determine the quality of wine because it gives the best outcome for the assurance of quality of wine. . The dependent variable is “quality rating” whereas other variables i.e. alcohol ,sulphur etc. are assumed to be predictors or independent variables. .While hindering the effectiveness of the data model, various types of errors have occurred like over fitting, introduced from having too large of a training set and bias occur due to too small of a test set .To improve the prediction of wine quality with the help of data sets the different machine learning algorithms are used in this project.The use of machine learning algorithms like Random forest ,Support Vector Classifier ,Stochastic Gradient Descent ,Logistic Regression.

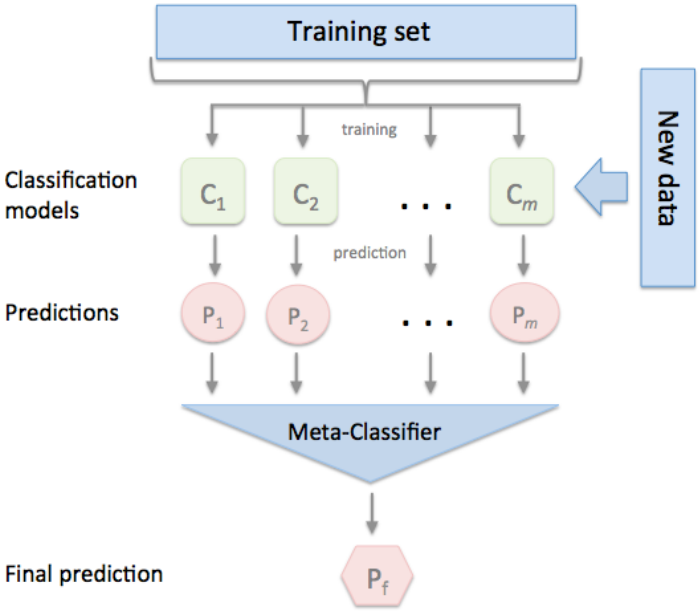


Fig: model to use different classifiers for prediction.

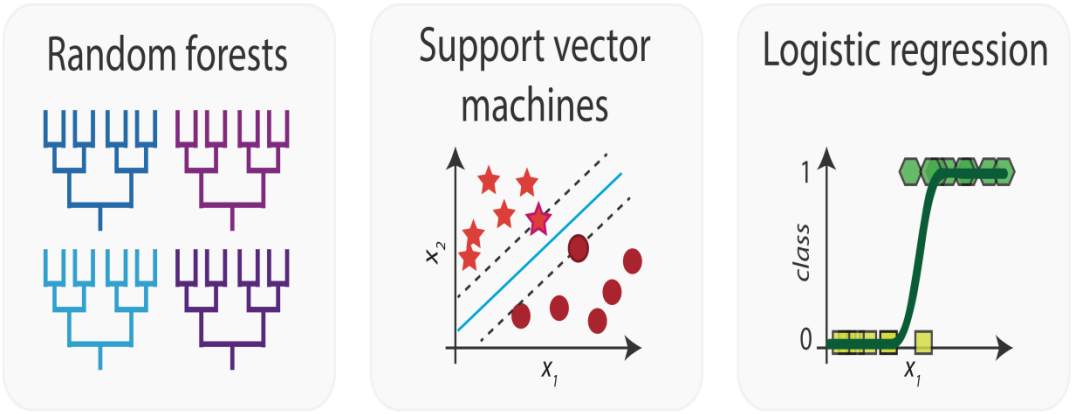


Fig: Different classifiers.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The quality of wine is predicted accurately by using this system and also it improves the accuracy of training and testing data of dataset. The quality is measured accurately by usage of the different algorithms and different classifiers in this model. It also proceeded with an automatic prediction system in it. The dependent and independent variables for the prediction is well-formed.

2.6 CONCLUSION

An automatic predictive system can be integrated into a decision support system, helping the speed and quality of the performance. Furthermore, a feature selection process can help to analyze the impact of the analytical tests. If it is concluded that several input variables are highly relevant to predict the wine quality, since in the production process some variables can be controlled, this information can be used to improve the wine quality. Classification models used here are

1) Random Forest

2) Stochastic Gradient Descent

3) SVC

4)Logistic Regression.

3. ANALYSIS

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. The technique has been applied in the study of mathematics and logic since before Aristotle, though analysis as a formal concept is a relatively recent development.

3.1 INTRODUCTION

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. The technique has been applied in the study of mathematics and logic since before Aristotle, though analysis as a formal concept is a relatively recent development.

3.2 SOFTWARE REQUIREMENT SPECIFICATION

3.2.1 Functional Specifications

Customer Specifications:

In the prediction of wine quality,the customer required to collect the various factors that are included in the wine, suppose the ingredients in the wine are shows the basic properties of the wine. Customer data is taken as CSV file in which the factors of wine are included. The various properties are classified based on the dependent and independent factors. These factors affect the system in classifying them into different parameters for training data.

Management Specifications:

The wine quality prediction is done based on the customer specified dataset which hold certain properties in dataset.The prediction is done after the classification of independent and dependent variables.In the process of predicting the dataset must undergo training and testing of such dataset for different evaluations. The entire prediction is done based on the various classifiers which are used in the process of predicting of wine quality.The list of factors which are included in the dataset are evaluated based on the training and testing of dataset. The different classifier are implied on the dataset to get much more accuracy of quality of wine.

- Different classifiers are implied on the dataset to get much more accuracy.
- The training and testing of data is done with multiple factors and properties.
- The model which gives maximum score for the prediction is given primary consideration for prediction.

3.3 CONTENT DIAGRAM

During the initial phases of modelling requirements, developers must first understand the context where the system operates in a specific environment. The context model captures this perspective to allow decisions on project scope to be made by the stakeholders. The context model also shows system dependencies to its interacting environment. The external dependencies could be another automated system, manual processes, and functional peripheral or actors. They might produce data for system usage or consume the output of the system.

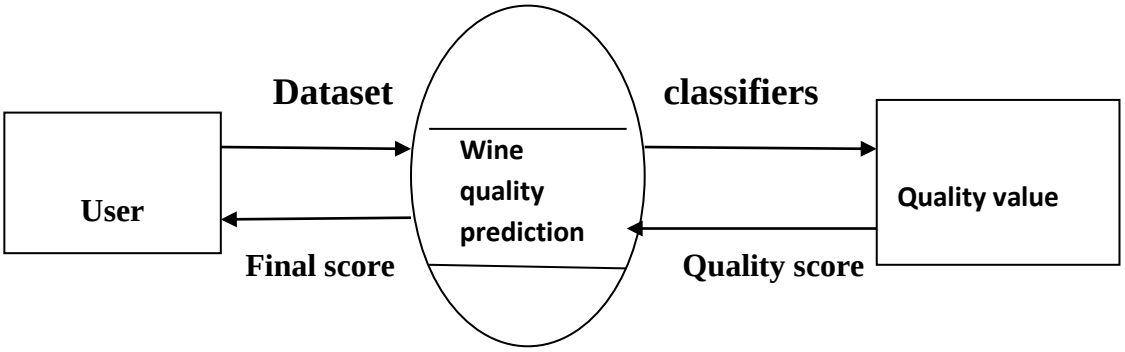


Fig: DFD Level 0

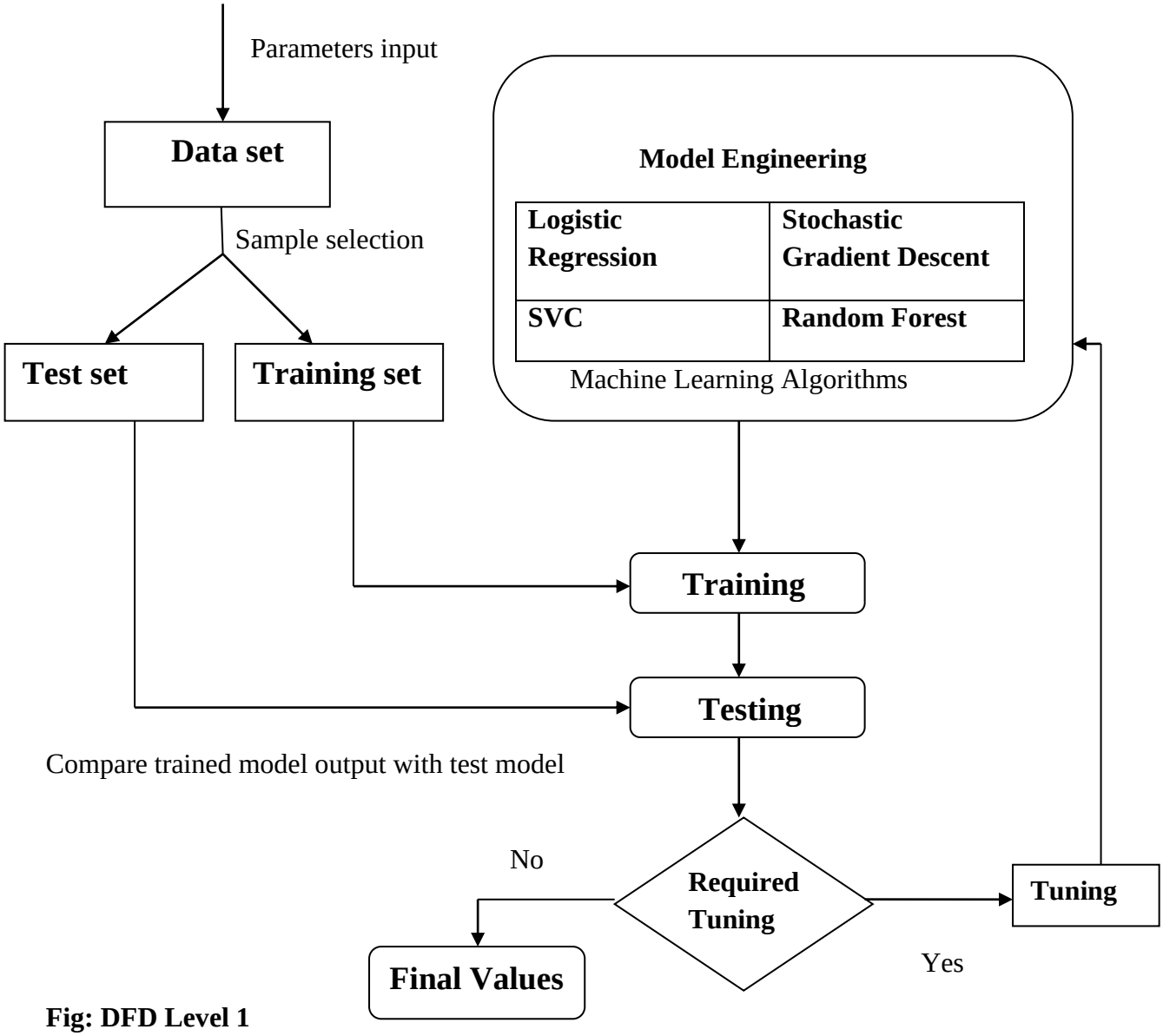


Fig: DFD Level 1

3.4 SYSTEM ARCHITECTURE

Very few systems use the available wine data for prediction purposes and even if they do, they are restricted by large number of association rules that apply. Disadvantages of using such systems are,

- Detection is not possible at an earlier stage.
- In the existing system, practical use of various, collected data is time consuming.
- This practice leads to errors and less accuracy rate.

The present model allows to predict the wine type and quality. The model is fed with various physiochemical properties of wine. The main goal of this system is to analyse the wine type and quality using machine learning algorithm such as Logistic regression, decision tree and Random forest. Wine data set is used and then preprocessed and transformed the data set. Then apply the decision tree algorithm on

the transformed data set. After applying the algorithm, wine type and quality is analysed and then the result obtained based on the prediction of whether the wine quality is low, high or medium.

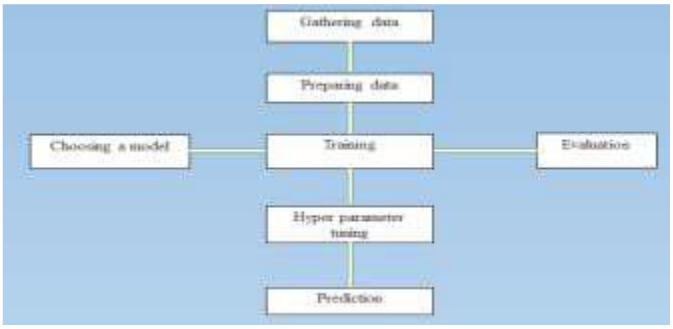


Fig : System Architecture

The above fig shows the steps in analysis of wine which is discussed below,
1. Gathering data
2. Preparing that data
3. Choosing a model
4. Training
5. Evaluation
6. Hyper parameter tuning
7. Prediction.

1. Gathering Data:
This step is very crucial as the quality and quantity of data gathered will directly determine how good the predictive model will turn out to be. The data collected is then tabulated and called as Training Data or Data Set. Theattribute values which determine the quality of and type of wine are given as data.

2. Data Preparation:
After the training data is gathered, next step is Data preparation, where the data is loaded into a suitable place and then prepared for use. Also, the data has to be split into two parts. The first part that is used in training our model,will be the majority of the dataset and the second will be used for the evaluation of the trained model's performance.

3. Choosing a model:
The next step that follows in the workflow is choosing a model.Choosing a model that is best suitable for analysis of wines. The model includes a Decision tree. It is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.
They provide a highly effective structure within which you can lay out options and investigate the possible outcomes of choosing those options. They also help you to form a balanced picture of the risks and rewards associated with each possible course of action.

4. Training:
The training process involves initializing some random values for model, predict the output with those values, then compare it with the model's prediction and then adjust

the values so that they match the predictions that were made previously.This process then repeats and each cycle of updating is called one training step.

5. Evaluation:
Evaluation allows the testing of the model against data that has never been seen and used for training. It is the representative of how the model will analysis in the real time.

6. Parameter Tuning:
Once the evaluation is over, any further improvement in training can be possible by tuning the parameters. There were a few parameters that were implicitly assumed when the training was done. Another parameter included is the learning rate that defines how far the line is shifted during each step, based on the information from the previous training step. These values all play a role in the accuracy of the training model, and how long the training will take.

7.Prediction: This is the point where the value of machine learning is realized. Previously predicted data will give the output.

3.5 CONCLUSION

The above architecture depicts the entire flow of the system which holds the different factors in the predicting of wine quality. The flow of system takes the sample values as user specified dataset. The input parameters split into multiple factors as training and testing of data, these data undergo into different classifiers for the predicting the accurate values for the wine quality.

4. DESIGN

4.1 INTRODUCTION

The wine data-frames and data sets are used here which are necessary for basic exploratory analysis and visualizations. The gathered data set will be used for data analysis and modeling through which output can be predicted. Data of different attributes and properties is considered for the analysis. The analysis of the data includes exploratory and predictive analysis.

4.2 SYSTEM MODELS

For making automated decisions on model selection we need to quantify the performance of our model and give it a score. For that reason, for the classifiers, we are using F1 score which combines two metrics: Precision which expresses how accurate the model was on predicting a certain class and Recall which expresses the inverse of the regret of missing out instances which are misclassified. Since we have multiple classes we have multiple F1 scores. We will be using the unweighted mean of the F1 scores for our final scoring. This is a business decision because we want our models to get optimized to classify instances that belong to the minority side, such as wine quality of 3 or 8 equally well with the rest of the qualities that are represented in a larger number. For the regression task we are scoring based on the coefficient of determination, which is basically a measurement of whether the predictions and the actual values are highly correlated. The larger this coefficient the better. For regressors we can also get F1 score if we first round our prediction.

Splitting for Testing : We are keeping 20% of our dataset to treat it as unseen data and be able and test the performance of our models. We are splitting our dataset in a way such that all of the wine qualities are represented proportionally equally in both training and testing dataset.
Other than that the selection is being done randomly with uniform distribution. Various classification and regression algorithms are used to fit the model. The algorithms used in this paper are as follows:

For classification:

Random Forest Decision Trees classifier
Support Vector Machine classifier
Stochastic gradient descent
Logistic Regression classifier

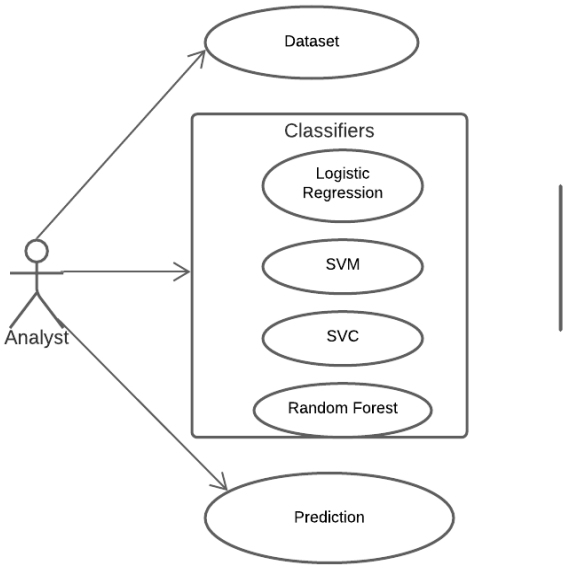


Fig: UseCase diagram

Preprocessing: Label Encoding is used to convert the labels into numeric form so as to convert it into the machine-readable form. It is an important pre-processing step for the structured dataset in supervised learning. We have used label encoding to label the quality of data as good or bad. Assigning 1 to good and 0 to bad.

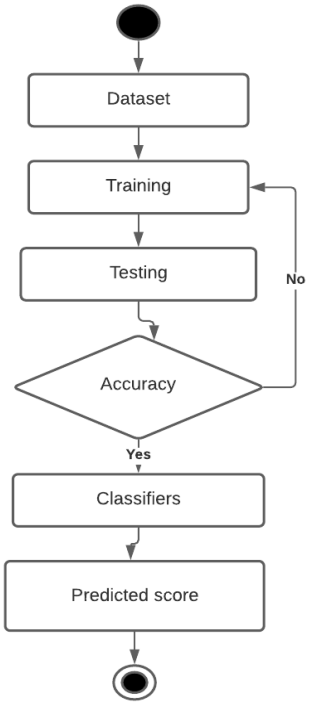


Fig: Activity diagram

Feature Selection:

As we can clearly see, volatile acidity and residual sugar are both not very impact full of the quality of wine. Hence we can eliminate these features. Though we are selecting these features, they will change according to the domain experts.

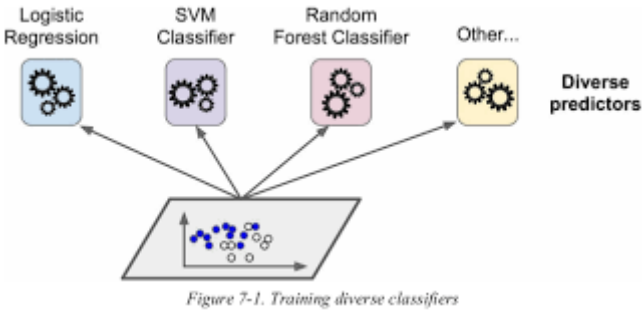


Fig: Model for processing the data.

4.3 MODULE DESIGN AND ORGANISATION

A. RANDOM FOREST

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest Algorithm

We can understand the working of Random Forest algorithm with the help of following steps −

- **Step 1** − First, start with the selection of random samples from a given dataset.
- **Step 2** − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3** − In this step, voting will be performed for every predicted result.
- **Step 4** − At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate its working **–**

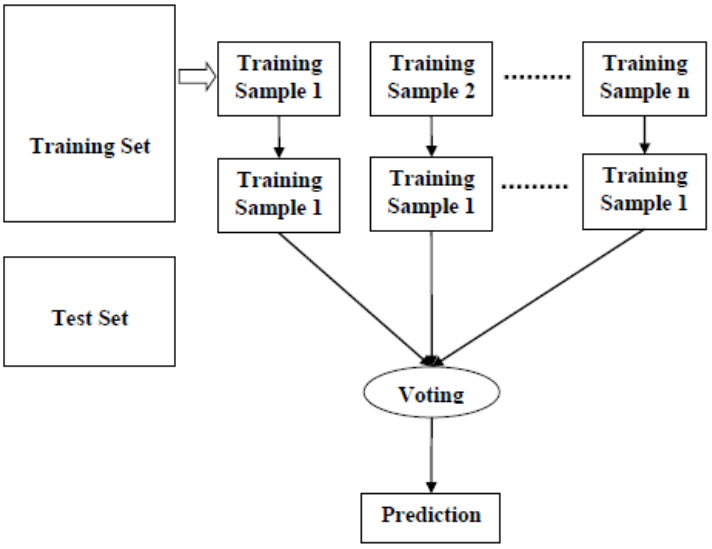


Fig: Random Forest Algorithm

B. LOGISTIC REGRESSION

Logistic regression is a classification algorithm, used when the value of the target variable is categorical in nature. Logistic regression is most commonly used when the data in question has binary output, so when it belongs to one class or another, or is either a 0 or 1.

It is important to understand that logistic regression should only be used when the target variables fall into discrete categories and that if there's a range of continuous values the target value might be, logistic regression should not be used. Examples of situations you might use logistic regression in include:

- Predicting if an email is spam or not spam
- Whether a tumor is malignant or benign
- Whether a mushroom is poisonous or edible.

When using logistic regression, a threshold is usually specified that indicates at what value the example will be put into one class vs. the other class. In the spam classification task, a threshold of 0.5 might be set, which would cause an email with a 50% or greater probability of being spam to be classified as "spam" and any email with probability less than 50% classified as "not spam".

Although logistic regression is best suited for instances of binary classification, it can be applied to multiclass classification problems, classification tasks with three or more classes. You accomplish this by applying a **"one vs. all**" strategy.

C. STOCHASTIC GRADIENT DESCENT

It is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for

a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features.

The advantages of Stochastic Gradient Descent are:

- Efficiency.
- Ease of implementation (lots of opportunities for code tuning).

The disadvantages of Stochastic Gradient Descent include:

- SGD requires a number of hyper parameters such as the regularization parameter and the number of iterations.
- SGD is sensitive to feature scaling.

D. SUPPORT VECTOR CLASSIFIER

- SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is known for its kernel trick to handle nonlinear input spaces. It is used in a variety of applications such as face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition.
- In this tutorial, you will be using scikit-learn in Python. If you would like to learn more about this Python package, I recommend you take a look at our Supervised Learning with scikit-learn course.
- SVM is an exciting algorithm and the concepts are relatively simple. The classifier separates data points using a hyperplane with the largest amount of margin. That's why an SVM classifier is also known as a discriminative classifier. SVM finds an optimal hyperplane which helps in classifying new data points.

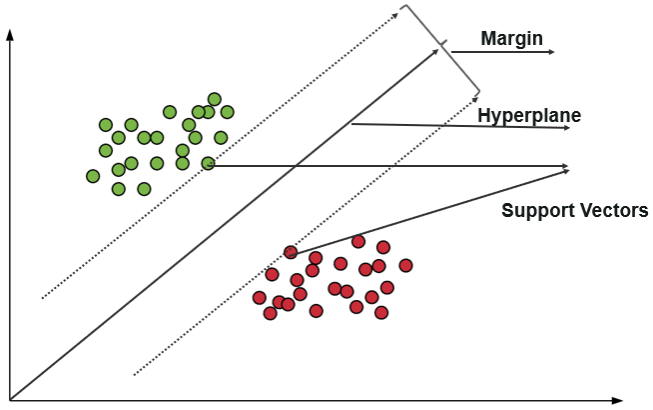


Fig: Support Vector Classifier

Support Vectors

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

Hyperplane

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Margin

A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

4.4 CONCLUSION

The specific objective of this study is to analyse how physicochemical properties like alcohol percentage, chlorides, sulphates contents etc., varies the quality of wine. This study analyse the wine types and quality with the various physicochemical variables. Analyse the wine type using logistic regression and quality by three machine learning algorithms such as decision tree, random forest and extreme gradient boosting. The results obtained are more accurate than previous techniques.

5. IMPLEMENTATION AND RESULT

5.1 INTRODUCTION

A major challenge faced by analysing wine type and quality is getting perfect accuracy within less time in the final result. This can be achieved by using some machine learning techniques.

5.1.1 Wine Attributes and Properties

Different attributes which are present in the wine which determine the wine types and its quality. Properties and attributes we have considered are:

- **Fixed acidity:** Acids are one of the fundamental properties of wine and contribute greatly to the taste of the wine. Reducing acids significantly might lead to wines tasting flat. This variable is usually expressed in *g(tartaric acid)/dm3* in the dataset.
- **Volatile acidity:** These acids are to be distilled out from the wine before completing the production process. It is primarily constituted of acetic acid.The volatile acidity is expressed in *g(acetic acid)/dm3* in the dataset.
- **Citric acid:** This is one of the fixed acids which gives a wine its freshness.Usually most of it is consumed during the fermentation process and sometimes it is added separately to give the wine more freshness. It's usually expressed in *g/dm3* in the dataset.
- **Residual sugar:** This typically refers to the natural sugar from grapes which remains after the fermentation process stops, or is stopped. It's usually expressed in *g/dm3* in the dataset.
- **Chlorides:** This is usually a major contributor to saltiness in wine. It's usually expressed in *g(sodium chloride)/dm3* in the dataset.
- **Free sulfur dioxide:** This is the part of the sulphur dioxide that when added to a wine is said to be free after the remaining part binds. Winemakers will always try to get the highest proportion of free sulphur to bind. This variable is expressed in *mg/dm3* in the dataset.
- **Total sulfur dioxide:** This is the sum total of the bound and the free sulfur dioxide (*SO2*). Here, it's expressed in *mg/dm3*..
- **Density:** This can be represented as a comparison of the weight of a specific volume of wine to an equivalent volume of water. It is generally used as a measure of the conversion of sugar to alcohol.Here, it's expressed in *g/cm3*.
- **pH:** Also known as the potential of hydrogen, this is a numeric scale to specify the acidity or basicity the wine. Fixed acidity contributes the most towards the pH of wines. You might know, solutions with a pH less than 7 are acidic, while solutions with a pH greater than 7 are basic. With a pH of 7, pure water is neutral. Most wines have a pH between 2.9 and 3.9 and are therefore acidic.
- **sulphates:** These are mineral salts containing sulfur. Sulphates are to wine as gluten is to food.They are a regular part of the winemaking aroundthe world and are considered essential. Here, it's expressed in *g(potassiumsulphate)/dm3* in the dataset.
- **Alcohol:** Wine is an alcoholic beverage. Alcohol is formed as a result of yeast

converting sugar during the fermentation process. The percentage of alcohol can vary from wine to wine. Hence it is not a surprise for this attribute to be a part of this dataset. It's usually measured in % vol or alcohol by volume (ABV).

- **Quality:** Wine experts graded the wine quality between 0 (very bad) and 10 (very excellent). The eventual quality score is the median of at least three evaluations made by the same wine experts.
- **Wine_type:** Since we originally had two datasets for red and white wine, we introduced this attribute in the final merged dataset which indicates the type of wine for each data point. A wine can either be a 'red' or a 'white' wine. One of the predictive models we will build would be such that we can predict the type of wine by looking at other wine attributes.
- **Quality_label:** This is a derived attribute from the quality attribute. We bucket or group wine quality scores into three qualitative buckets namely low,medium and high.

5.1.2 Machine Learning

Machine Learning is extensively used in analysing wine type and quality. Machine learning is an application of artificial intelligence(AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are often categorized as supervised, unsupervised and semi supervised algorithms:

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
- In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.
- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labeled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources

in order to train it/learn from it.Otherwise, acquiring unlabelled data generally doesn't require additional resources.

5.1.3 Introduction to Classifiers

The wine data-frames and data sets are used here which are necessary for basic exploratory analysis and visualizations. The gathered data set will be used for data analysis and modeling through which output can be predicted. Data of different attributes and properties is considered for the analysis. The analysis of the data includes exploratory and predictive analysis.

- **Exploratory data analysis**: Standard Machine Learning and analytics workflow recommend processing, cleaning, analysing, and visualizing your data before moving on toward modelling data. The required packages and necessary dependencies for analysis can be imported by using these code.

  ```
  import pandas as pd
  import matplotlib.pyplot as plt
  import matplotlib as mpl
  import numpy as np
  import seaborn as sns
  %matplotlib inline
  ```

  The datasets (red and white wine) and add some additional variables are used that is required to predict in future sections. The first variable to be added is wine_type, which would be either red or white wine based on the dataset and the wine sample. The second variable to be added is quality_label which is a qualitative measure of the quality of the wine samplebased on the quality variable score. The rules used for mapping quality to quality_label are described as follows.
  - Wine quality scores of 3, 4, and 5 are mapped to low quality wines under the quality_label attribute.
  - Wine quality scores of 8 and 9 are mapped to high• quality wines under the quality_label attribute.

- **Descriptive statistics**: Some descriptive statistics of various features of interest in the dataset should be computed. This involves computing aggregation metrics like mean, median, standard deviation, and so on. The primary objectives is to build a model that can correctly predict if a wine is a red or white wine based on its attributes.

- **Inferential Statistics:** The general notion of inferential statistics is to draw inferences and propositions of a population using a data sample. The idea is to use statistical methods and models to draw statistical inferences from a given hypotheses. Each hypothesis consists of a null hypothesis and an alternative hypothesis. Based on statistical test results, if the result is statistically significant based on pre-set significance levels (e.g., if obtained p-value is less than 5% significance level), the null hypothesis is rejected in favor of the alternative hypothesis. Otherwise, if the results is not statistically significant,

concluded as that the null hypothesis was correct.

- **Univariate analysis:** Is the simplest form of analysing data. Uni ‖ means ―one‖, so in other words the data has only one variable. It doesn't deal with causes or relationships and its major purpose is to describe: it takes data, summarizes that data and finds patterns in the data. A variable in univariate analysis is just a condition or subset that the data falls into. It can be considered as a category. For example the analysis might look at a variable age or it might look at height or weight. Univariate descriptive statistics: some ways patterns found in univariate data include central tendency (mean, mode and median) and dispersion can be described: range, variance, maximum, minimum, quartiles, and standard deviation. Several options are there for describing data.

- **Multivariate analysis:** Is used to study more complex sets of data than what univariate analysis methods can handle. There are more than 20 different ways to perform multivariate analysis. This type of analysis is almost always performed with software as working with even the smallest of data sets can be overwhelming by hand.

5.2 PREDICTING WINE TYPES

In the wine quality dataset, there are two variants or types of wine—red and white wine. The main task of classification system is to predict the wine type based on other features.
To start with, necessary features should be selected and separate the prediction class labels and prepare train and test datasets. Prefix wtp_ is used in the variables to easily identify them as needed, where wtp depicts wine type prediction. The model is designed and implemented using
the logistic regression algorithm.

5.2.1 Logistic Regression

It is one of the most used Machine Learning algorithms for binary classification. It is a simple Algorithm and use as a performance baseline, it is easy to implement and it will do well enough in many tasks. The building block concepts of Logistic Regression can also be helpful in deep learning while building neural networks. Here it is used to predict the wine type whether it is red or white. There are three types of logistic regression algorithm:
Binary, Multi and ordinal. The flow chart of this algorithm can be return as

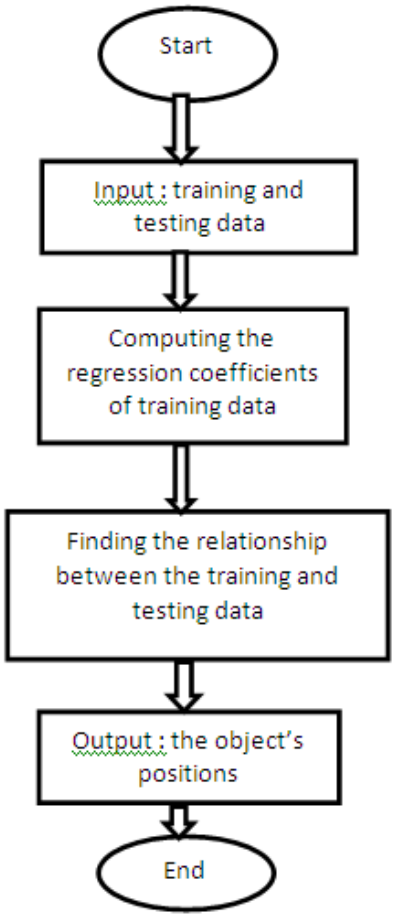


Fig : Flow chart of logistic regression

Predicting Wine Quality

In the wine quality dataset, there are several quality rating classes ranging from 3 to 9. Quality_label variable is focused that classifies wine into low, medium, and high ratings based on the underlying quality variable based on the mapping created in the ―Exploratory Data Analysis‖. This is done because several rating scores have very few wine samples and hence similar quality ratings were clubbed together into one quality class rating. Prefix wqp_is used for all variables and models involved in prediction of wine quality to distinguish it from other analysis. The prefix wqp stands for wine quality prediction .The following algorithms are used for classifying quality.

Decision tree algorithm

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification **problems** too. Here this algorithm is used to analyse the quality of wine. A decision tree is a flowchart-like structure in

which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

In decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated.
A decision tree consists of three types of nodes:
1. **Decision nodes** – The node where a there is a requirement set that determines the outcome,
typically represented by squares.
2. **Chance nodes** – A node were there isn't a set requirement to determine where to go in the split
but something that just has a probability of happening or not Ex: 50% chance of success or failure of a business ,typically represented by circles
3. **End nodes** – The end of a split, so something feeds into this but nothing comes out of it. Usually some result, ex: Business is successful , typically represented by triangles.

Algorithm :

INPUT : S, where S= set of classified instances
OUTPUT : Decision tree
Require : S=0, num_attributes>0
1: procedure BUILDTREE
2: repeat
3: maxGain←0
4: splitA←null
5: e← Entropy(Attributes)
6: for all Attributes a in S do
7: gain←InformationGain(a,e)
8: if gain>maxGain then
9: maxGain← gain
10: splitA← a
11: end if
12: end for
13: Partition(S,splitA)
14: until all partitions processed
15: end procedure

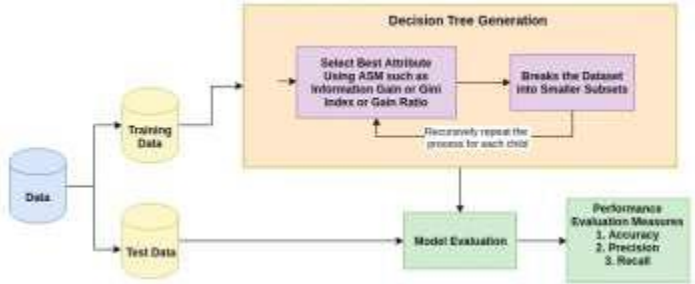


Fig : Workflow of decision tree algorithm

In this project decision tree algorithm is used to classify the wine data into its different quality and types. The leaf node will have the attributes with its value and represents its quality. One can easily understand by looking into the generated decision tree.

5.2.2 Random Forest Algorithm

Random Forest algorithm is a supervised classification algorithm. We can see it from its name, which is to create a forest by some way and make it random. There is a direct relationship between the number of trees in the forest and the results it can get: the larger the number of trees, the more accurate the result. But one thing to note is that creating the forest is not the same as constructing the decision with information gain or gain index approach.
The difference between Random Forest algorithm and the decision tree algorithm is that in Random Forest, the processes of finding the root node and splitting the feature nodes will run randomly. Some applications of using Random Forest algorithm: Banking, Medicine, Stock Market
and E-commerce:

- For the application in banking, Random Forest algorithm is used to find loyal customers, which means customers who can take out plenty of loans and pay interest to the bank properly, and fraud customers, which means customers who have bad records like failure to pay back a loan on time or have dangerous actions.
- For the application in medicine, Random Forest algorithm can be used to both identify the correct combination of components in medicine, and to identify diseases by analyzing the patient's medical records.
- For the application in the stock market, Random Forest algorithm can be used to identify a stock's behavior and the expected loss or profit.
- For the application in e-commerce, Random Forest algorithm can be used for predicting whether the customer will like the recommend products, based on the experience of similar customers.

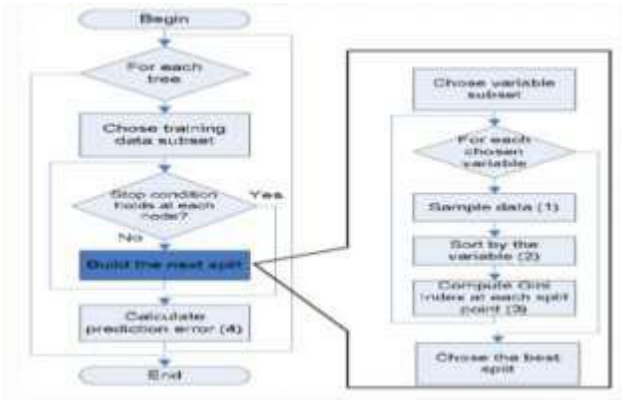


Fig : Flowchart of random forest algorithm

Table 1: List of attributes and values with its types and quality

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	wine_type	quality_label
0	7.0	0.17	0.74	12.8	0.045	24.0	126.0	0.99420	3.26	0.38	12.2	8	white	high
1	7.7	0.64	0.21	2.2	0.077	32.0	133.0	0.99560	3.27	0.45	9.9	5	red	low
2	6.8	0.39	0.34	7.4	0.020	38.0	133.0	0.99212	3.18	0.44	12.0	7	white	medium
3	6.3	0.28	0.47	11.2	0.040	61.0	183.0	0.99592	3.12	0.51	9.5	6	white	medium
4	7.4	0.35	0.20	13.9	0.054	63.0	229.0	0.99888	3.11	0.50	8.9	6	white	medium

Table 1 consists of the result of loading and merging the two datasets. Based on the physicochemical properties type of wine and quality will be predicted. It gives both type and quality of wine.

Table 2 List of attributes of wine types with statistical values

	Red Wine Statistics						White Wine Statistics					
	residual sugar	total sulfur dioxide	sulphates	alcohol	volatile acidity	quality	residual sugar	total sulfur dioxide	sulphates	alcohol	volatile acidity	quality
count	1599.00	1599.00	1599.00	1599.00	1599.00	1599.00	4898.00	4898.00	4898.00	4898.00	4898.00	4898.00
mean	2.54	46.47	0.66	10.42	0.53	5.64	6.39	138.36	0.49	10.51	0.28	5.88
std	1.41	32.90	0.17	1.07	0.18	0.81	5.07	42.50	0.11	1.23	0.10	0.89
min	0.90	6.00	0.33	8.40	0.12	3.00	0.60	9.00	0.22	8.00	0.08	3.00
25%	1.90	22.00	0.55	9.50	0.39	5.00	1.70	108.00	0.41	9.50	0.21	5.00
50%	2.20	38.00	0.62	10.20	0.52	6.00	5.20	134.00	0.47	10.40	0.26	6.00
75%	2.60	62.00	0.73	11.10	0.64	6.00	9.90	167.00	0.55	11.40	0.32	6.00
max	15.50	289.00	2.00	14.90	1.58	8.00	65.80	440.00	1.08	14.20	1.10	9.00

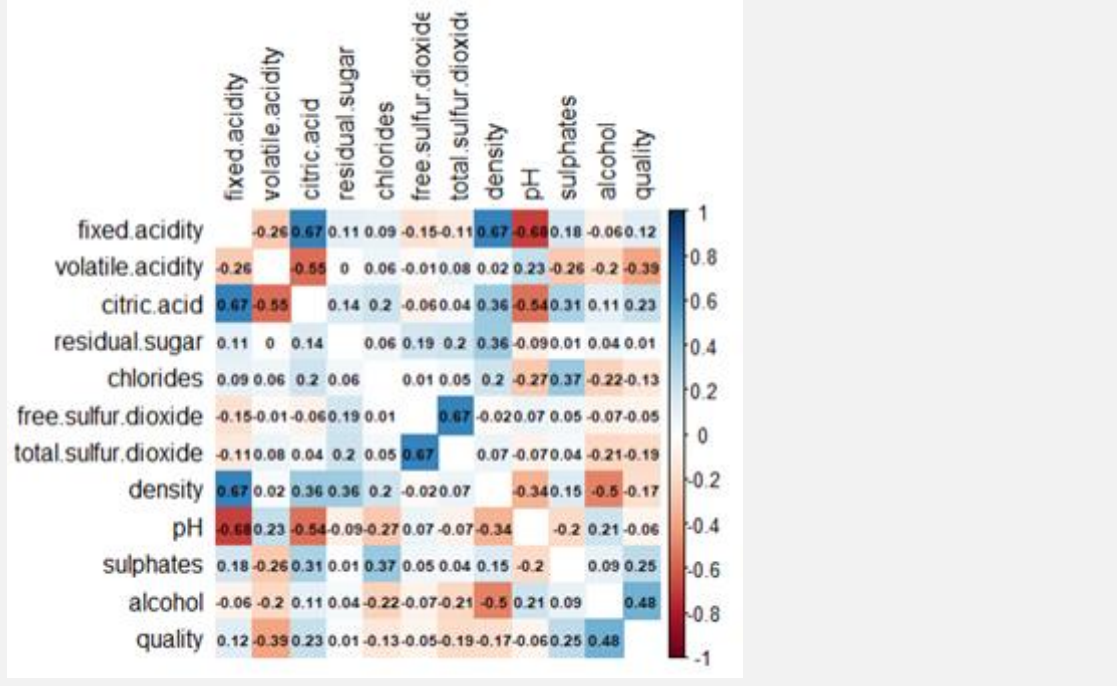
Table 3 List of attributes of wine quality with statistical values

	Low Quality Wine				Medium Quality Wine				High Quality Wine			
	alcohol	volatile acidity	pH	quality	alcohol	volatile acidity	pH	quality	alcohol	volatile acidity	pH	quality
count	2384.00	2384.00	2384.00	2384.00	3915.00	3915.00	3915.00	3915.00	198.00	198.00	198.00	198.00
mean	9.87	0.40	3.21	4.88	10.81	0.31	3.22	6.28	11.69	0.29	3.23	8.03
std	0.84	0.19	0.16	0.36	1.20	0.14	0.16	0.45	1.27	0.12	0.16	0.16
min	8.00	0.10	2.74	3.00	8.40	0.08	2.72	6.00	8.50	0.12	2.88	8.00
25%	9.30	0.26	3.11	5.00	9.80	0.21	3.11	6.00	11.00	0.21	3.13	8.00
50%	9.60	0.34	3.20	5.00	10.80	0.27	3.21	6.00	12.00	0.28	3.23	8.00
75%	10.40	0.50	3.31	5.00	11.70	0.36	3.33	7.00	12.60	0.35	3.33	8.00
max	14.90	1.58	3.90	5.00	14.20	1.04	4.01	7.00	14.00	0.85	3.72	9.00

Table .2 and .3 gives the statistical values of some of the attributes values from which its type and quality are determined. Table 5.2 has values for attributes such as residual sugar, total sulfur dioxide, sulphates and volatile acidity with the quality values. This analysis is made for two types of wine that is red and white. Table 5.3 has values for attributes such as alcohol, volatile acidity, ph. and quality values. Analysis is made for quality types that is low, medium and high.

Data Exploration and Transformation

To see which variables are likely to affect the quality of red wine the most, I ran a correlation analysis of our independent variables against our dependent variable, quality. This analysis ended up with a list of variables of interest that had the highest correlations with quality.



In order of highest correlation, these variables are:

1. **Alcohol**: the amount of alcohol in wine

2. **Volatile acidity**: are high acetic acid in wine which leads to an unpleasant vinegar taste

3. **Sulphates**: a wine additive that contributes to SO2 levels and acts as an antimicrobial and antioxidant

4. **Citric Acid**: acts as a preservative to increase acidity (small quantities add freshness and flavor to wines)

5. **Total Sulfur Dioxide:** is the amount of free + bound forms of SO2

6. **Density:** sweeter wines have a higher density

7. **Chlorides:** the amount of salt in the wine

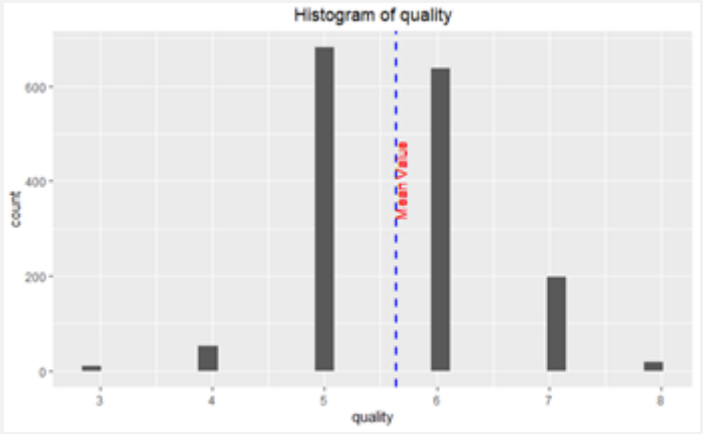
8. **Fixed acidity:** are non-volatile acids that do not evaporate readily

9. **pH:** the level of acidity

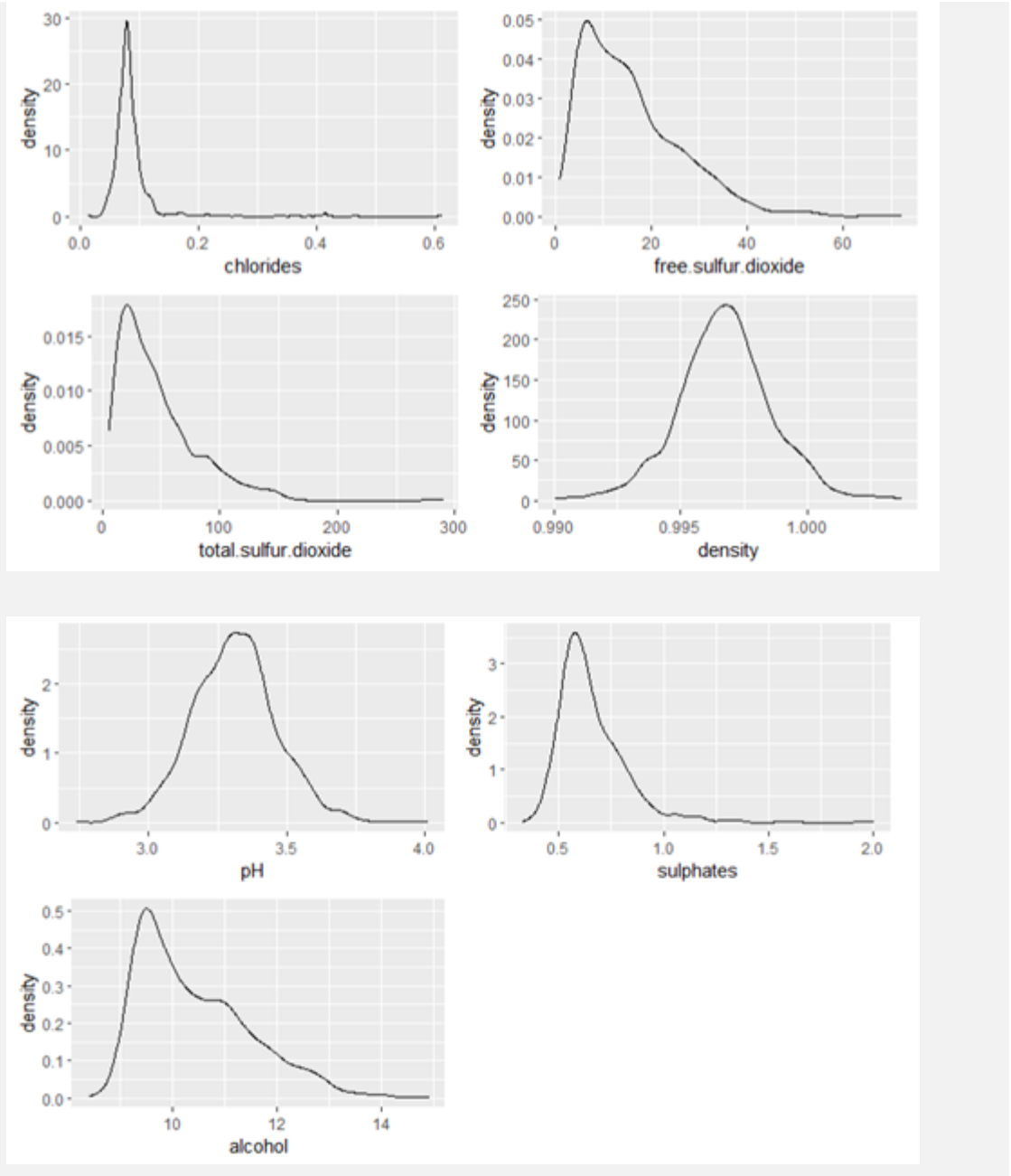
10. **Free Sulfur Dioxide**: it prevents microbial growth and the oxidation of wine

11. **Residual sugar**: is the amount of sugar remaining after fermentation stops. The key is to have a perfect balance between — sweetness and sourness (wines > 45g/ltrs are sweet)

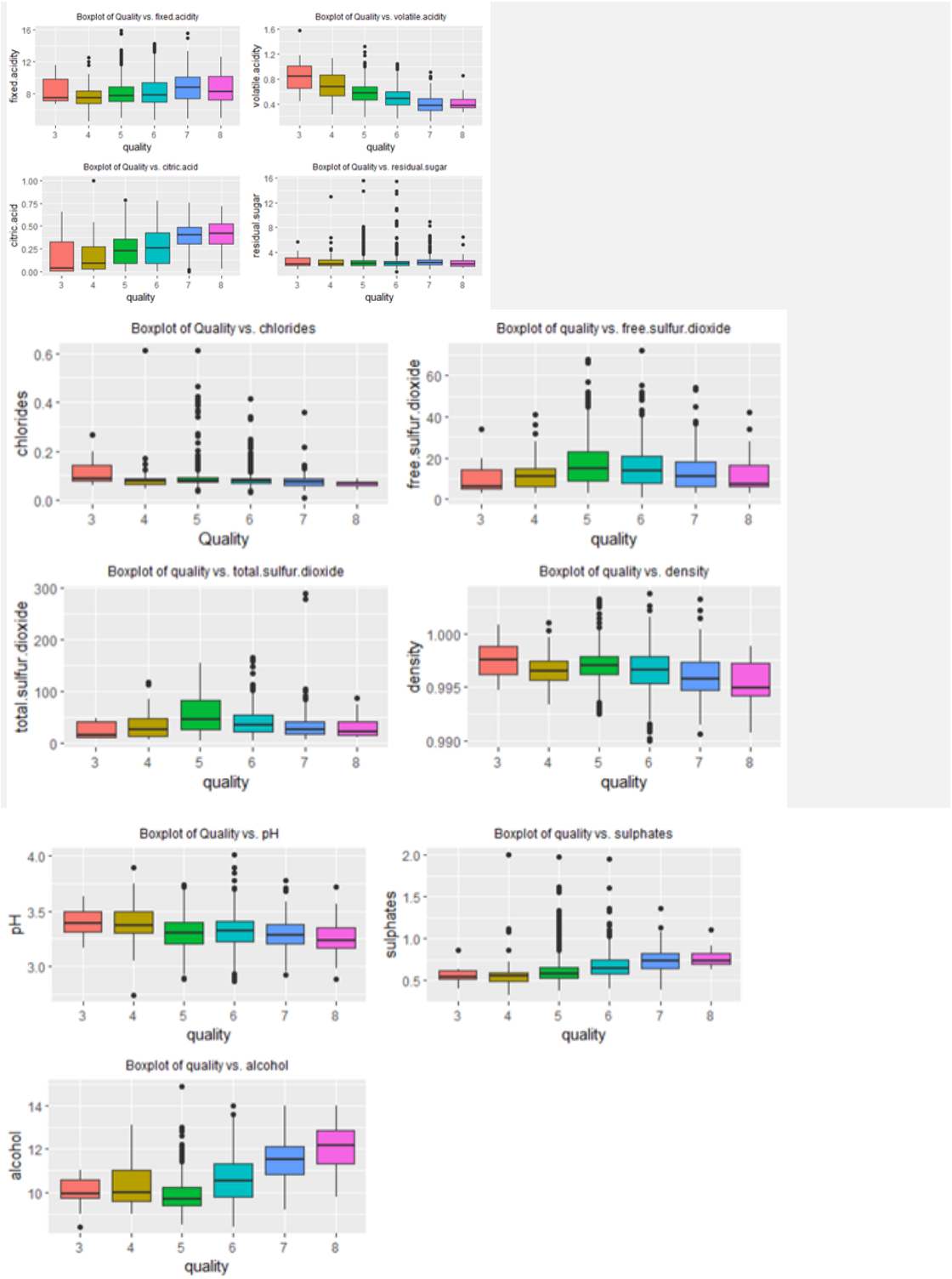
Starting with our dependent variable, quality, I found the popularity of the medium/average values of quality: 5 and 6. Considering the dependent variable's transformation, I found out that our data is normally distributed. This conclusion can be verified by running a QQ plot, which shows no need to transform our data.



Next, for independent numerical variables, the first step to further analyze the relationship with our dependent variable was to create density plots visualizing the spread of the data.

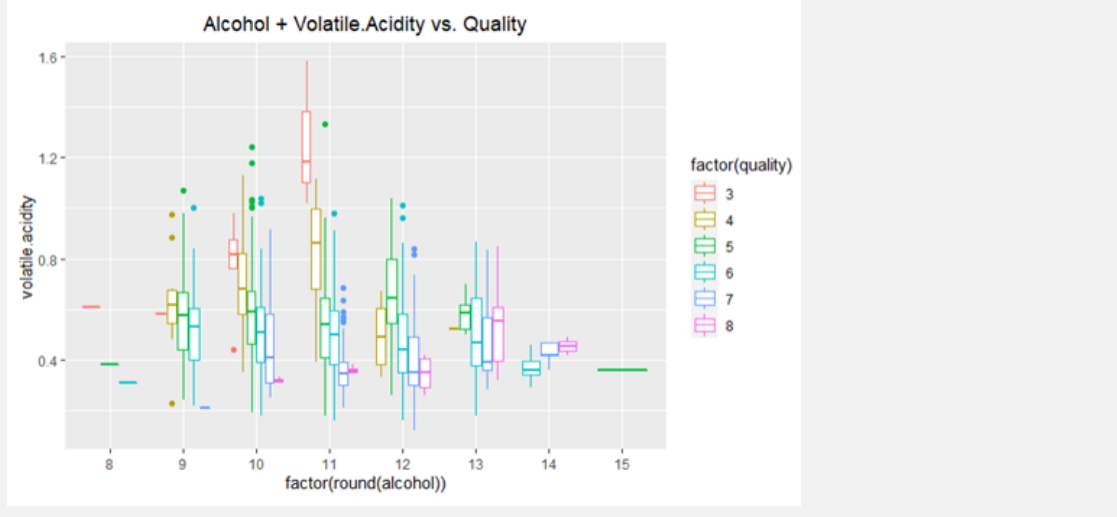


It can be seen that most red wines' pH levels are always between 3–4 and chlorides — the amount of salt is most prevalent at level 0.1. After analyzing the density plots, I plotted the interaction between our numeric variables of interest and our dependent variable of quality.

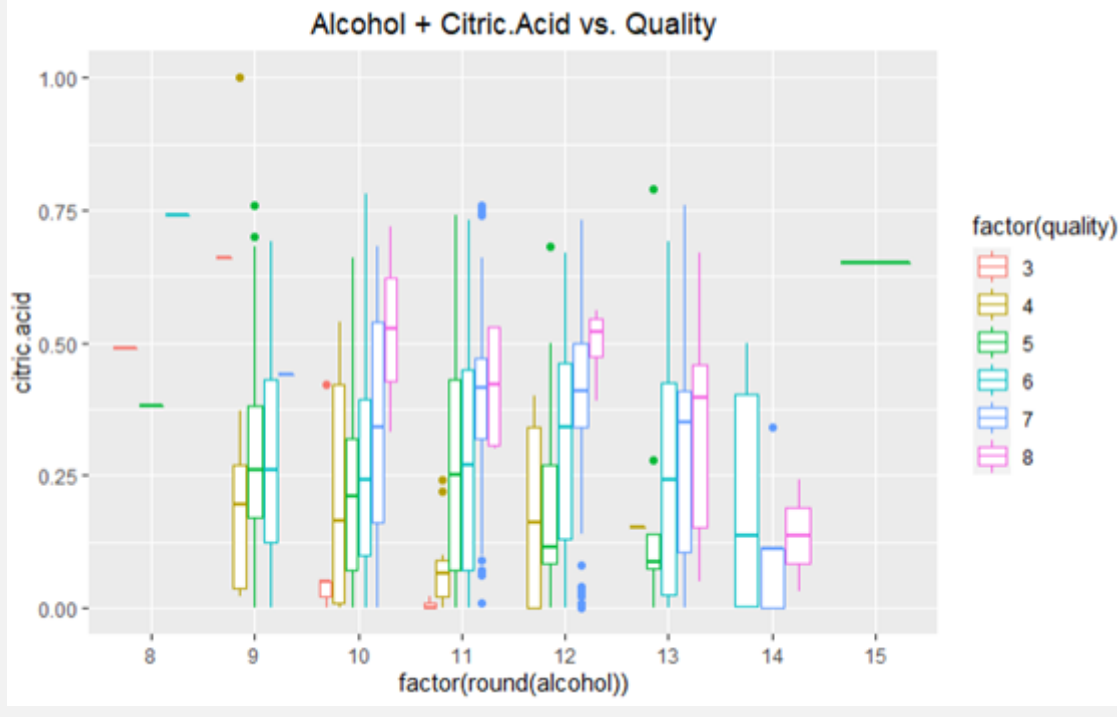


Three different patterns can be observed. First, there are positive relationships between quality and critic.acid, alcohol, and sulphates. Even though wines with a higher level of alcohol may make them less popular, they should be highly rated in quality. Second, there are negative relationships between quality and volatile.acidity, density, and pH. It is reasonable that less sweet wines and a lower level of acidity are favored in quality testings. Last, these independent variables show no significant relationship with quality: residual.sugar, chlorides, and total.sulfur.dioxide.

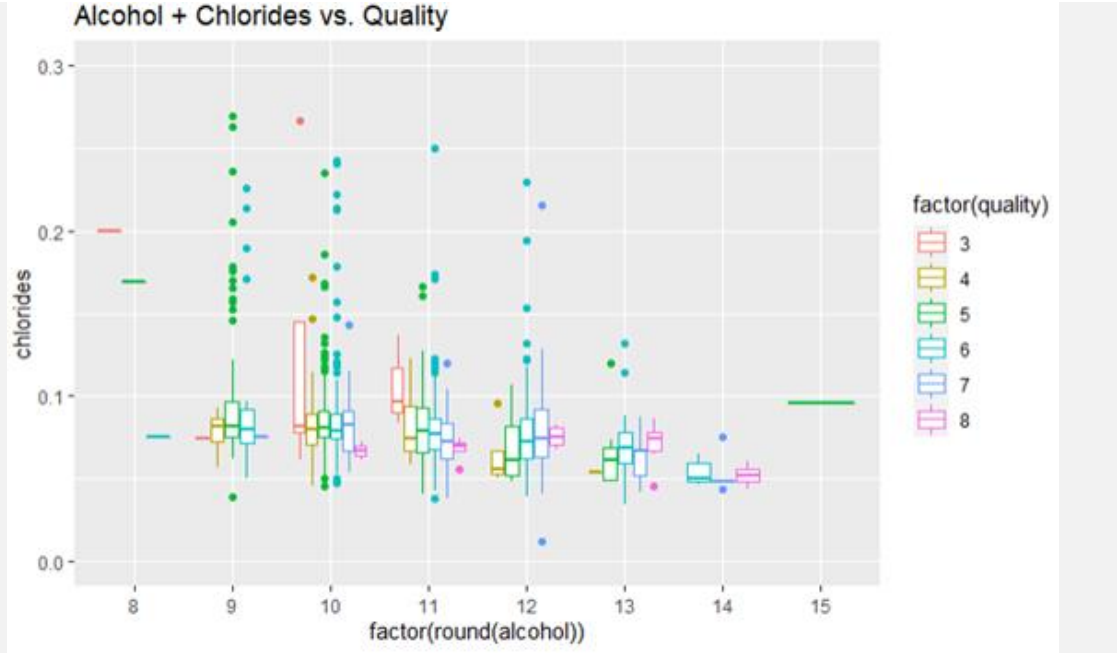
To dive deep into relationships within independent variables and with quality, I built different three-dimensional plots. When inspecting the two variables, alcohol and volatile.acidity with quality, we can see that with red wines' alcohol level between 9% to 12%, the level of volatile acidity decreases as the wines' alcohol level increases. For higher alcohol content (>12%), the pattern reverses, implying high-quality wines' popularity.



Keep researching the alcohol variable, I selected the citric.acid and visualized their interactions with quality. Interestingly, for wines with an alcohol percentage level below 14, as the level of citric acid increases, there is a rise in red wines' quality. The only exception was at alcohol 14%, where the citric acid level drops as the wine's quality increases.



Finally, an interaction analysis using chlorides in relationships with alcohol and quality shows that the wines' quality decreases when chloride level decreases at the alcohol before 12%. However, the quality of red wine increases as the chloride level increases at the alcohol level from 12%.



Last, we considered if the collinearity problem existed in our data. As a result of correlation analysis and VIF verification, we discovered some variables with slightly high correlations.

5.2.3 Stochastic Gradient Descent

It is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features.

The advantages of Stochastic Gradient Descent are:

- Efficiency.
- Ease of implementation (lots of opportunities for code tuning).

The disadvantages of Stochastic Gradient Descent include:

- SGD requires a number of hyper parameters such as the regularization parameter and the number of iterations.
- SGD is sensitive to feature scaling.

Stochastic Gradient Descent Classifier

In [45]:
```
from sklearn.linear_model import SGDClassifier

# creating the model
model = SGDClassifier(penalty=None)

# feeding the training model into the model
model.fit(x_train, y_train)

# predicting the values for the test set
y_pred = model.predict(x_test)

# classification report
print(classification_report(y_test, y_pred))
```

Fig: Stochastic Gradient Descent algorithm

In [30]:
```
print(classification_report(y_test, pred_svc))
```

```
              precision    recall  f1-score   support

           0       0.88      0.98      0.93       273
           1       0.71      0.26      0.37        47

    micro avg       0.88      0.88      0.88       320
    macro avg       0.80      0.62      0.65       320
weighted avg       0.86      0.88      0.85       320
```

5.2.4 Support Vector Classifier

- SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is known for its kernel trick to handle nonlinear input spaces. It is used in a variety of applications such as face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition.
- In this tutorial, you will be using scikit-learn in Python. If you would like to learn more about this Python package, I recommend you take a look at our Supervised Learning with scikit-learn course.
- SVM is an exciting algorithm and the concepts are relatively simple. The classifier separates data points using a hyperplane with the largest amount of margin. That's why an SVM classifier is also known as a discriminative classifier. SVM finds an optimal hyperplane which helps in classifying new data points.

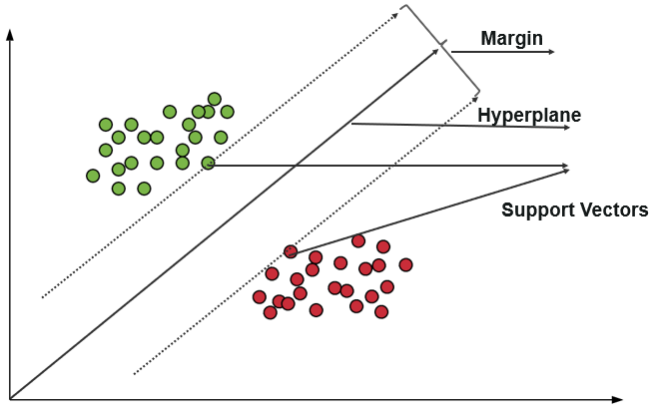


Fig: Support Vector Classifier

Support Vectors

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

Hyperplane

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Margin

A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

Support Vector Machine

In [46]:
```
from sklearn.svm import SVC

# creating the model
model = SVC()

# feeding the training set into the model
model.fit(x_train, y_train)

# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
```

Fig: SVM implementation

```
In [50]: # creating a new SVC model with these best parameters

         model2 = SVC(C = 1.4, gamma = 0.1, kernel = 'rbf')
         model2.fit(x_train, y_train)
         y_pred = model2.predict(x_test)

         print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.75      0.73      0.74       204
           1       0.72      0.74      0.73       196

   micro avg       0.73      0.73      0.73       400
   macro avg       0.74      0.74      0.73       400
weighted avg       0.74      0.73      0.74       400
```

Fig: SVM with different parameters.

5.3 SOFTWARE TOOLS

5.3.1 OPEARTING SYSYTEM: WINDOWS 10

Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015 and broadly released for retail sale on July 29, 2015.
Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10 which are available to Windows Insiders. The latest stable build of Windows 10 is Version 1903 (May 2019 Update). Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

One of Windows 10's most notable features is its support for universal apps, an expansion of the Metro-style apps first introduced in Windows 8. Universal apps can be designed to run across multiple Microsoft product families with nearly identical code - available input devices - particularly on 2-in-1 PCs, both interfaces include an including PCs, tablets, smartphones, embedded systems, Xbox One, Surface Hub and Mixed Reality. The Windows user interface was revised to handle transitions between a mouse-oriented interface and a touchscreen-optimized interface based on updated Start menu which incorporates elements of Windows 7's traditional Start menu which incorporates elements of Windows 7's traditional start menu with the tiles of Windows 8. Windows 10 also introduced the Microsoft Edge web browser, a virtual desktop system, a window and desktop management feature called Task View, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX 12.

5.3.2 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages,

environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac os, and Linux.

To get Navigator, get the Navigator Cheat Sheet and install Anaconda. The Getting started with Navigator section shows how to start Navigator from the shortcuts or from a terminal window.

Why use Navigator?
In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.
The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments them - all inside Navigator.

What applications can be accessed?
The following applications are available by default in Navigator:

- Jupyter Lab
- Jupyter Notebook
- Spyder
- VSCode
- Glueviz
- Rstudio

Advanced conda users can also build their own Navigator applications.

5.3.3 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the "ipynb" extension.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell.

To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user. Jupyter Notebook provides a browser-based REPL built upon a number of popular open source libraries:

- Python
- ØMQ
- Tornado (web server)
- jQuery

5.3.4 PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library Most Python implementations (including CPython) include a read-eval-print loop (REPL), permitting them to function as a command line Interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting, As well as standard desktop integrated development environments, there are Web browser-based IDEs; SageMath (intended for developing science and math-related Python programs); Python Anywhere, a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing scientific computing.

5.4 EXECUTION

Code for entire classification and prediction is:

```
#!/usr/bin/env python

# coding: utf-8

# **Installing Libraries**

# In[15]:

import numpy as np
```

```
import pandas as pd

import matplotlib.pyplot as plt

# **Importing the Dataset**

# In[16]:


data=pd.read_csv(r"C:\Users\lenovo\Desktop\WineQuality
Predictions/winequality-red.csv")

data.head()

# In[17]:

data.columns

# In[18]:

data.describe()

# In[19]:

data.info()

# In[20]:

data.isnull().sum()

# In[21]:

data['quality'].value_counts()

# **Data Visualization**

# **Bivariate Analysis**

# In[22]:

# checking the variation of fixed acidity in the different qualities of wine


plt.scatter(data['quality'], data['fixed acidity'], color = 'green')

plt.title('relation of fixed acidity with wine')

plt.xlabel('quality')

plt.ylabel('fixed acidity')

plt.legend()

plt.show()
```

```
# In[23]:
# checking the variation of fixed acidity in the different qualities of wine.

plt.bar(data['quality'], data['alcohol'], color = 'maroon')
plt.title('relation of alcohol with wine')
plt.xlabel('quality')
plt.ylabel('alcohol')
plt.legend()
plt.show()

# In[24]:
# Composition of citric acid go higher as we go higher in the quality of the wine
import seaborn as sns
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'citric acid', data = data)

# In[25]:
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'residual sugar', data = data)
# In[26]:
#Composition of chloride also go down as we go higher in the quality of the wine
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'chlorides', data = data)
# In[27]:
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'free sulfur dioxide', data = data)

# In[28]:
#Sulphates level goes higher with the quality of wine
```

```
fig = plt.figure(figsize = (10,6))

sns.barplot(x = 'quality', y = 'sulphates', data = data)

# In[29]:

#Sulphates level goes higher with the quality of wine

fig = plt.figure(figsize = (10,6))

sns.barplot(x = 'quality', y = 'sulphates', data = data)
```

As we can see that like the above two items do not have very strong relation to the dependent variable we have to showcase a correlation plot to check which of the items are more related to the dependent variable and which items are less related to the dependent variables.

```
# In[30]:

f, ax = plt.subplots(figsize=(10, 8))

corr = data.corr()

sns.heatmap(corr,mask=np.zeros_like(corr,dtype=np.bool),
cmap=sns.diverging_palette(220, 10, as_cmap=True), square=True, ax=ax)
```

From the above correlation plot for the given dataset for wine quality prediction, we can easily see which items are related strongly with each other and which items are related weekly with each other.

For Example,

The strongly correlated items are :

#

1.fixed acidity and citric acid.

2.free sulphur dioxide and total sulphor dioxide.

3.fixed acidity and density.

4. alcohol and quality.

so, from above points there is a clear inference that alcohol is the most important characteristic to determine the quality of wine.

The weekly correlated items are :

1.citric acid and volatile acidity.

2.fixed acidity and ph.

```
# 3.density and alcohol.

# These are some relations which do not depend on each other at all.

# In[31]:

sns.pairplot(data)


# **Data pre-processing**

# In[32]:

# Removing Unnecassary columns from the dataset

# As we saw that volatile acidity, total sulphor dioxide, chlorides, density are very less related to the dependent variable

# quality so even if we remove these columns the accuracy won't be affected that much.

#data = data.drop(['volatile acidity', 'total sulfur dioxide', 'chlorides', 'density'], axis = 1)

# checking the shape of the dataset

#print(data.shape)

# In[33]:


data.columns


# In[34]:

# converting the response variables(3-7) as binary response variables that is either good or bad

#names = ['bad', 'good']

#bins = (2, 6.5, 8)

#data['quality'] = pd.cut(data['quality'], bins = bins, labels = names)


data['quality'] = data['quality'].map({3 : 'bad', 4 :'bad', 5: 'bad',
                                       6: 'good', 7: 'good', 8: 'good'})


# In[35]:
```

```
# analyzing the different values present in the dependent variable(quality column)
data['quality'].value_counts()
# In[36]:
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['quality'] = le.fit_transform(data['quality'])
data['quality'].value_counts


# In[37]:
sns.countplot(data['quality'])
# In[38]:
# dividing the dataset into dependent and independent variables
x = data.iloc[:,:11]
y = data.iloc[:,11]
# determining the shape of x and y.
print(x.shape)
print(y.shape)


# In[40]:
# dividing the dataset in training and testing set


from sklearn.model_selection import train_test_split x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 44)


# determining the shapes of training and testing sets
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
# In[52]:
```

```
# standard scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)

# **Modelling**
# **Logistic Regression**
# In[44]:


from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import GridSearchCV, cross_val_score

# creating the model
model = LogisticRegression()
# feeding the training set into the model
model.fit(x_train, y_train)
# predicting the results for the test set
y_pred = model.predict(x_test)
# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))

# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
```

```
print(confusion_matrix(y_test, y_pred))

# **Stochastic Gradient Descent Classifier**
# In[45]:

from sklearn.linear_model import SGDClassifier

# creating the model
model = SGDClassifier(penalty=None)

# feeding the training model into the model
model.fit(x_train, y_train)

# predicting the values for the test set
y_pred = model.predict(x_test)

# classification report
print(classification_report(y_test, y_pred))

# **Support Vector Machine**
# In[46]:


from sklearn.svm import SVC
# creating the model
model = SVC()

# feeding the training set into the model
model.fit(x_train, y_train)
```

```
# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))



# In[47]:
# finding the best parameters for the SVC model

param = {
        'C': [0.8,0.9,1,1.1,1.2,1.3,1.4],
        'kernel':['linear', 'rbf'],
        'gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
grid_svc = GridSearchCV(model, param_grid = param, scoring = 'accuracy', cv = 10)

# In[48]:

grid_svc.fit(x_train, y_train)



# In[49]:
grid_svc.best_params_
# In[50]:
# creating a new SVC model with these best parameters
```

```
model2 = SVC(C = 1.4, gamma = 0.1, kernel = 'rbf')
model2.fit(x_train, y_train)
y_pred = model2.predict(x_test)
print(classification_report(y_test, y_pred))



# **Random Forest**
# In[51]:

from sklearn.ensemble import RandomForestClassifier
# creating the model
model = RandomForestClassifier(n_estimators = 200)

# feeding the training set into the model
model.fit(x_train, y_train)

# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))



# In[59]:
# classification report
print(classification_report(y_test, y_pred))
# confusion matrix
print(confusion_matrix(y_test, y_pred))
```

In[60]:

#Now lets try to do some evaluation for random forest model using cross validation.

model_eval = cross_val_score(estimator = model, X = x_train, y = y_train, cv = 10)

model_eval.mean()

5.5 CONCLUSION

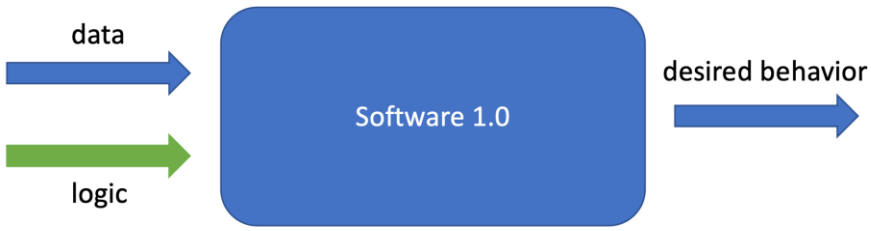
These days, machine learning techniques are being widely used to solve real-world problems by storing, manipulating, extracting and retrieving data from large sources. Supervised machine learning techniques have been widely adopted however these techniques prove to be very expensive when the systems are implemented over wide range of data. This is due to the fact that significant amount of effort and cost is involved because of obtaining large labeled data sets. Thus active learning provides a way to reduce the labeling costs by labeling only the most useful instances for learning.

When coming from a complex model to the more simplification of the truth model', the part of accuracy will be sacrificed. The research in ML application must have analysis about the trade-off between speed and accuracy.

6. TESTING AND VALIDATION

6.1 INTRODUCTION

In traditional software systems, humans write the logic which interacts with data to produce a desired behavior. Our software tests help ensure that this **written logic** aligns with the actual expected behavior.



However, in machine learning systems, humans provide desired behavior as examples during training and the model optimization process produces the logic of the system. How do we ensure this **learned logic** is going to consistently produce our desired behavior?

Let's start by looking at the best practices for testing traditional software systems and developing high-quality software.

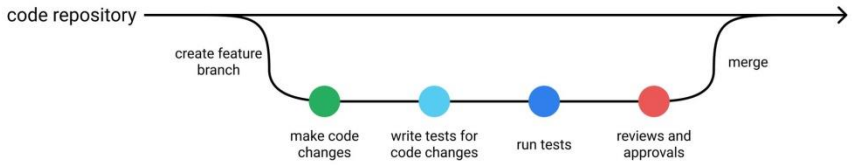
A typical software testing suite will include:

- **unit tests** which operate on atomic pieces of the codebase and can be run quickly during development,
- **regression tests** replicate bugs that we've previously encountered and fixed,
- **integration tests** which are typically longer-running tests that observe higher-level behaviors that leverage multiple components in the codebase, and follow conventions such as:

➢ don't merge code unless all tests are passing,

➢ always write tests for newly introduced logic when contributing code,

➢ when contributing a bug fix, be sure to write a test to capture the bug and prevent future regressions.



A typical workflow for software development.

When we run our testing suite against the new code, we'll get a report of the specific behaviors that we've written tests around and verify that our code changes don't affect the expected behavior of the system. If a test fails, we'll know which specific behavior is no longer aligned with our expected output. We can also look at this testing report to get an understanding of how extensive our tests are by looking at metrics such as **code coverage**.

Let's contrast this with a typical workflow for developing machine learning systems. After training a new model, we'll typically produce an evaluation report including:

- performance of an established metric on a validation dataset,
- plots such as precision-recall curves,
- operational statistics such as inference speed,
- examples where the model was most confidently incorrect,and follow conventions such as:
- save all of the hyper-parameters used to train the model,
- only promote models which offer an improvement over the existing model (or baseline) when evaluated on the same dataset.

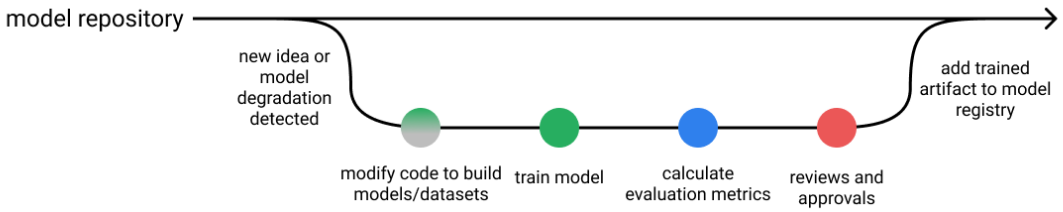


Fig:A typical workflow for model development.

When reviewing a new machine learning model, we'll inspect metrics and plots which summarize model performance over a validation dataset. We're able to compare performance between multiple models and make relative judgements, but we're not immediately able to characterize specific model behaviors. For example, figuring out *where* the model is failing usually requires additional investigative work; one common practice here is to look through a list of the top most egregious model errors on the validation dataset and manually categorize these failure modes.

Assuming we write behavioral tests for our models (discussed below), there's also the question of whether or not we have enough tests! While traditional software tests have metrics such as the lines of code covered when running tests, this becomes harder to quantify when you shift your application logic from lines of code to parameters of a

machine learning model. Do we want to quantify our test coverage with respect to the input data distribution? Or perhaps the possible activations inside the model?

Odena et al. introduce one possible metric for coverage where we track the model logits for all of the test examples and quantify the area covered by radial neighborhoods around these activation vectors. However, my perception is that as an industry we don't have a well-established convention here. In fact, it feels like that testing for machine learning systems is in such early days that this question of test coverage isn't really being asked by many people.

6.2 DESIGN OF TEST CASES SCENARIOS & VALIDATION

In my opinion, there's two general classes of model tests that we'll want to write.

- **Pre-train tests** allow us to identify some bugs early on and short-circuit a training job.
- **Post-train tests** use the trained model artifact to inspect behaviors for a variety of important scenarios that we define.

Pre-train tests

There's some tests that we can run without needing trained parameters. These tests include:

- check the shape of your model output and ensure it aligns with the labels in your dataset
- check the output ranges and ensure it aligns with our expectations (eg. the output of a classification model should be a distribution with class probabilities that sum to 1)
- make sure a single gradient step on a batch of data yields a decrease in your loss make assertions about your datasets
- check for label leakage between your training and validation datasets

The main goal here is to identify some errors early so we can avoid a wasted training job.

Post-train tests

However, in order for us to be able to understand model behaviors we'll need to test against trained model artifacts. These tests aim to interrogate the logic learned during training and provide us with a behavioral report of model performance.

Importing the Dataset

```
In [16]: data = pd.read_csv(r"C:\Users\lenovo\Desktop\Wine Quality Predictions/winequality-red.csv")

         data.head()
```

Out[16]:

	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

Fig: Passing data sets to model

To check various parameters that affecting the model for prediction is analyzed by bi variate analysis are

Bivariate Analysis

```
In [22]: # checking the variation of fixed acidity in the different qualities of wine

         plt.scatter(data['quality'], data['fixed acidity'], color = 'green')
         plt.title('relation of fixed acidity with wine')
         plt.xlabel('quality')
         plt.ylabel('fixed acidity')
         plt.legend()
         plt.show()
```

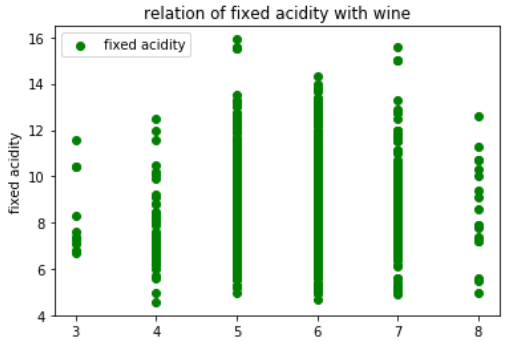


Fig: Bivariate Analysis

We have used, train_test_split() function that we imported from sklearn to split the data. Notice we have used test_size=0.2 to make the test data 20% of the original data. The rest 80% is used for training.

Now let's print and see the first five elements of data we have split using head() function.

```
print(X_train.head())
```

Training Data Using head()

After we obtained the data we will be using, the next step is data normalization. It is part of pre-processing in which data is converted to fit in a range of -1 and 1. These are simply, the values which are understood by a machine learning algorithm easily.

We can observe, that now the values of all the train attributes are in the range of -1 and 1 and that is exactly what we were aiming for.

Time has now come for the most exciting step, training our algorithm so that it can predict the wine quality. We do so by importing a DecisionTreeClassifier() and using fit() to train it.

```
#Now separate the dataset as response variable and feature variabes
X = wine.drop('quality', axis = 1)
y = wine['quality']

#Train and Test splitting of data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 42)

#Applying Standard scaling to get optimized result
sc = StandardScaler()

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

svm

Fig: training and testing of data

6.3 CONCLUSION

Machine learning systems are trickier to test due to the fact that we're not explicitly writing the logic of the system. However, automated testing is still an important tool for the development of high-quality software systems. These tests can provide us with a behavioral report of trained models, which can serve as a systematic approach towards error analysis. Throughout this blog post, I've presented "traditional software development" and "machine learning model development" as two separate concepts. This simplification made it easier to discuss the unique challenges associated with testing machine learning systems; unfortunately, the real world is messier. Developing machine learning models also relies on a large amount of "traditional software development" in order to process data inputs, create feature representations, perform data augmentation, orchestrate model training, expose interfaces to external systems, and much more. Thus, effective testing for machine learning systems requires both a traditional software testing suite (for model development infrastructure) and a model testing suite (for trained models).

7. CONCLUSION AND FUTURE ENHANCEMENTS

Algorithms used for classification are:
1) Logistic Regression
2) Stochastic gradient descent
3) Support Vector Classifier
4) Random Forest

- Logistic Regression gave us an accuracy of 86% Performance matrix of Logistic Regression

Precision	Recall	F1-Score	Support
0	0.88	0.98	0.93	273
1	0.71	0.26	0.37	47

- Stochastic gradient descent was able to give an average accuracy of 81%. Performance matrix of SGD:

Precision	Recall	F1-Score	Support
0	0.89	0.93	0.91	273
1	0.42	0.30	0.35	47

- Support Vector Classifier has given an accuracy of 85%. Performance matrix of SVC:

Precison	Recall	F1-Score	Support
0	0.89	0.93	0.91	273
1	0.71	0.26	0.37	47

- Random Forest gave us an accuracy of 87.33%

Precision	Recall	F1-Score	Support
0	0.90	0.97	0.93	273
1	0.68	0.40	0.51	47

Based on the bar plots plotted we come to an conclusion that not all input features are essential and affect the data, for example from the bar plot against quality and residual sugar we see that as the quality increases residual sugar is moderate and does not have change drastically. So this feature is not so essential as compared to others like alcohol and citric acid, so we can drop this feature while feature selection.
For classifying the wine quality, we have implemented multiple algorithms, namely
1) Logistic Regression
2) Stochastic gradient descent
3) Support Vector Classifier
4) Random Forest
We were able to achieve maximum accuracy using random forest of 88%. Stochastic gradient descent giving an accuracy of 81% .SVC has an accuracy of 85% and logistic regression of 86%.

FUTURE ENHANCEMENTS

The outcomes in this research are based on results that involve only sample datasets. It is necessary that additional datasets should be considered for the evaluation of different classification problems as the information growth in the recent technology is extending to heights beyond assumptions. Recent field of technology is growing and data are by nature dynamic. Hence, further classification of the entire system needs to be implemented right from the scratch since the results from the old process have become obsolete. The scope of future work can deal with Incremental learning, which the existing model and processes the new incoming data more efficiently.

8. REFERENCES

1. P.Appalasamy and A. Mustapha,(2012) "Classificationbased Data Mining Approach for Quality Control in Wine Production", Volume 12 (6), Journal of Applied Sciences.
2. Shen Yin,(2013) "Research Article: Quality Evaluation Based on Multivariate Statistical Methods", Article ID 639652, 10 pages.
3. DimitrijaAngelkov,(2016) "Data mining of wine quality",Volume 45, August 31.
4. Y. Subba Reddy and Prof. P. Govindarajulu,(2017),"An Efficient User Centric Clustering Approach for Product Recommendation Based on Majority Voting: A Case Study on Wine Data Set", IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.10.
5. Bernad Chen,(2016) ―Wine Informatics-Association rule based classification‖, 11th December 2016. Miami, FL,USA.
6. Yesim Er and AytenAtasoy,(2016) "The Classification of White Wine and Red Wine According to Their Physicochemical Qualities".
7. Zhang Lingfeng and Feng Feng,(2017) "Wine quality identification based on data mining research",12th International Conference on Computer Science and Education (ICCSE).

APPENDIX-A

Structured/Unstructured data: Structured Data in CSV format.

Various researches and students have published related work in national and international research papers, thesis to understand the objective, types of algorithm they have used and various techniques for pre-processing.
College of Intelligent Science and Engineering, China has written a paper on Evaluation and Analysis Model of Wine Quality Based on Mathematical Model.They have used various mathematical test to predict the quality of wine.The Mann-Whitney U test is used to analyze the wine evaluation results of the two wine tasters, and it is found that the significant difference between the two is small.

APPENDIX-B

Source Code:

1. **Logistic Regression**

```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report, confusion_matrix

from sklearn.model_selection import GridSearchCV, cross_val_score

# creating the model

model = LogisticRegression()

# feeding the training set into the model

model.fit(x_train, y_train)

# predicting the results for the test set

y_pred = model.predict(x_test)

# calculating the training and testing accuracies

print("Training accuracy :", model.score(x_train, y_train))

print("Testing accuracy :", model.score(x_test, y_test))

# classification report

print(classification_report(y_test, y_pred))

# confusion matrix

print(confusion_matrix(y_test, y_pred))
```

2. **Stochastic Gradient Descent Classifier**

```
from sklearn.linear_model import SGDClassifier
```

```
# creating the model

model = SGDClassifier(penalty=None)

# feeding the training model into the model

model.fit(x_train, y_train)

# predicting the values for the test set

y_pred = model.predict(x_test)

# classification report

print(classification_report(y_test, y_pred))
```

3. Support Vector Machine

```
from sklearn.svm import SVC

# creating the model

model = SVC()

# feeding the training set into the model

model.fit(x_train, y_train)

# predicting the results for the test set

y_pred = model.predict(x_test)

# calculating the training and testing accuracies

print("Training accuracy :", model.score(x_train, y_train))

print("Testing accuracy :", model.score(x_test, y_test))
```

4. Random Forest

```
from sklearn.ensemble import RandomForestClassifier

# creating the model
```

```
model = RandomForestClassifier(n_estimators = 200)

# feeding the training set into the model

model.fit(x_train, y_train)

# predicting the results for the test set

y_pred = model.predict(x_test)

# calculating the training and testing accuracies

print("Training accuracy :", model.score(x_train, y_train))

print("Testing accuracy :", model.score(x_test, y_test))

# classification report

print(classification_report(y_test, y_pred))

# confusion matrix

print(confusion_matrix(y_test, y_pred))
```