**Wine Quality Prediction using**

**Machine Learning Algorithms**

**Project ID: 26096**

*B.Tech. Project Report*

*submitted for fulfillment of*

*the requirements for the*

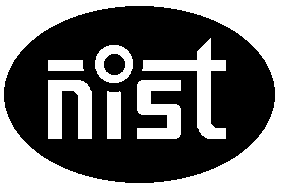
*Degree of Bachelor of Technology*

*Under Biju Pattnaik University of Technology*

*Submitted By*

**A. Tushar Kumar Patro Roll No. CSE201912524**

**Ashutosh Sahu Roll No. CSE201910344**



*2022– 2023*

*Under the guidance of*

**Dr. Sudhir Ranjan Pattanaik**

**NIST INSTITUTE OF SCIENCE & TECHNOLOGY (Autonomous)**

**Institute Park, Palur Hills, Berhampur, Odisha – 761008, India**

# ABSTRACT

The quality of a wine is important for the consumers as well as the wine industry. The traditional (expert) way of measuring wine quality is time-consuming. Nowadays, machine learning models are important tools to replace human tasks. In this case, there are several features to predict the wine quality but the entire features will not be relevant for better prediction. So, our thesis work is focusing on what wine features are important to get the promising result. For the purpose of classification model and evaluation of the relevant features, we used seven algorithms namely Logistic Regression, Random Forest, Decision Tree, SVC, XGBoost, KNN, Gaussian Naive Bayes. In this study, we used two wine quality datasets red wine and white wine. To evaluate the feature importance we used the Pearson coefficient correlation and performance measurement matrices such as accuracy, recall, precision, and f1 score for comparison of the machine learning algorithm.

Keywords— Classification, Logistic Regression, Random Forest, Decision Tree, SVC, XGBoost , KNN, Gaussian Naive Bayes.

# ACKNOWLEDGEMENT

We would like to take this opportunity to thank all those individuals whose invaluable contribution in a direct or indirect manner has gone into the making of this project a tremendous learning experience for me.

It is our proud privilege to epitomize my deepest sense of gratitude and indebtedness to my faculty guide, **Dr. Sudhir Ranjan Pattanaik** for his valuable guidance, keen and sustained interest, intuitive ideas and persistent endeavor. His guidance and inspirations enabled us to complete this report work successfully.

We give our sincere thanks to **Dr. Susmita Mahato** (Department Project Coordinator) for giving us the opportunity and motivating us to complete the project within stipulated period of time and providing a helping environment.

We acknowledge with immense pleasure the sustained interest, encouraging attitude and constant inspiration rendered by **Prof. (Dr.) Sukant K. Mohapatra (Chairman), Dr. Sudhir Ranjan Pattanaik (H.O.D of CSE) and Dr. Kunjabihari Swain (Department Project Coordinator).** Their continued drive for better quality in everything that happens at N.I.S.T. and selfless inspiration has always helped us to move ahead.

**A. Tushar Kumar Patro**

**Roll No. CSE201912524**

**Ashutosh Sahu**

**Roll No. CSE201910344**

# TABLE OF CONTENTS

[ABSTRACT i](#_Toc131428208)

[ACKNOWLEDGEMENT ii](#_Toc131428209)

[TABLE OF CONTENTS iii](#_Toc131428210)

[LIST OF FIGURES v](#_Toc131428211)

[1. INTRODUCTION 1](#_Toc131428212)

[2. PROBLEM STATEMENT 2](#_Toc131428213)

[3. WINE QUALITY PREDICTION: WORKFLOW 3](#_Toc131428214)

[4. DATASET DESCRIPTION 4](#_Toc131428215)

[5. METHODOLOGY 6](#_Toc131428216)

[5.1 Data Collection 6](#_Toc131428217)

[5.2 Data Preprocessing 7](#_Toc131428218)

[5.2.1 Data Quality Assessment 8](#_Toc131428219)

[5.2.2 Data Cleaning 8](#_Toc131428220)

[5.2.3 Data Transformation 9](#_Toc131428221)

[6. DATA VISUALIZATION 10](#_Toc131428222)

[6.1 Exploratory Data Analysis 11](#_Toc131428223)

[7. FEATURE SELECTION & FEATURE SCALING 15](#_Toc131428224)

[7.1 Feature Selection 15](#_Toc131428225)

[7.2 Feature Scaling 15](#_Toc131428226)

[8. DATA PARTITION 16](#_Toc131428227)

[9. MACHINE LEARNING ALGORITHMS 17](#_Toc131428228)

[9.1 Classification 17](#_Toc131428229)

[9.2 Regression 17](#_Toc131428230)

[9.1.1 Logistic Regression 18](#_Toc131428231)

[9.1.2 K-Nearest-Neighbour 18](#_Toc131428232)

[9.1.3 Support Vector Classifier 19](#_Toc131428233)

[9.1.4 Decision Tree 20](#_Toc131428234)

[9.1.5 Random Forest 22](#_Toc131428235)

[9.1.6 XGBoost 23](#_Toc131428236)

[9.1.7 Gaussian Naive Bayes 24](#_Toc131428237)

[10. EXPERIMENTAL RESULT, ANALYSIS & EVALUATION 25](#_Toc131428238)

[10.1 F1 Score 25](#_Toc131428239)

[10.2 Recall 25](#_Toc131428240)

[10.3 Precision 25](#_Toc131428241)

[10.4 Accuracy 26](#_Toc131428242)

[10.5 Confusion Matrix 26](#_Toc131428243)

[10.6 R²Score 27](#_Toc131428244)

[10.7 Mean Square Error 27](#_Toc131428245)

[10.8 Mean Absolute Error 28](#_Toc131428246)

[10.9 ROC AUC Score AND Curve 29](#_Toc131428247)

[11. CONCLUSION 31](#_Toc131428248)

[12. FUTURE WORK 32](#_Toc131428249)

[REFERENCES 33](#_Toc131428250)

# LIST OF FIGURES

[Figure 3.1 Wine Quality Prediction Workflow 3](#_Toc131428251)

[Figure 4.1: Dataset 5](#_Toc131428252)

[Figure 5.1. Flowchart of Wine Quality Prediction 6](#_Toc131428253)

[Figure 5.2 Types of Data 7](#_Toc131428254)

[Figure 6.1 Data Visualization 10](#_Toc131428255)

[Figure 6.2 Count plot of type 11](#_Toc131428256)

[Figure 6.3 Count plot of Quality 11](#_Toc131428257)

[Figure 6.4 Quality vs Density(KDEplot) 12](#_Toc131428258)

[Figure 6.5 Pair plot 12](#_Toc131428259)

[Figure 6.6 Heat Map 13](#_Toc131428260)

[Figure 6.7 Density Plot 14](#_Toc131428261)

[Figure 6.8 Histogram 14](#_Toc131428262)

[Figure 9.1 Decision Tree 21](#_Toc131428263)

[Figure 9.2 Random forest 23](#_Toc131428264)

[Figure 9.3 XG Boost 24](#_Toc131428265)

[Figure 10.1 R² Score 2](#_Toc131428266)7

[Figure 10.2 Mean Square Error 28](#_Toc131428267)

[Figure 10.3 Mean Absolute Error 29](#_Toc131428268)

[Figure 10.4 ROC AUC Score AND Curve 30](#_Toc131428269)

# 1. INTRODUCTION

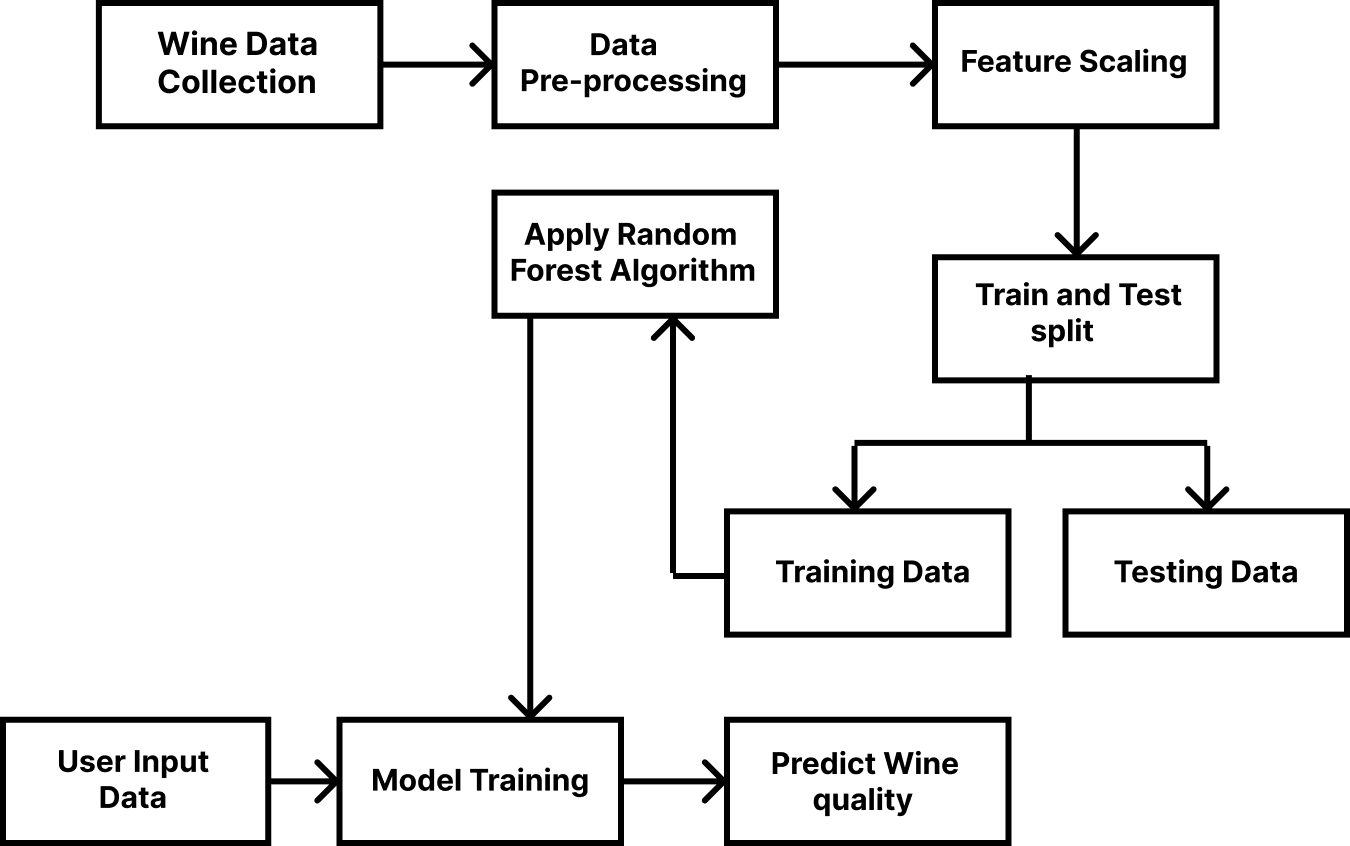
Wine is the most commonly used beverage globally, and its values are considered important in society. Quality of the wine is always important for its consumers, and mainly for producers in the present competitive market to raise the revenue. Historically, wine quality used to be determined by testing at the end of the production; to reach that level, one already spends lots of time and money. If the quality is not good, then the various procedure needs to be implemented from the beginning, which is very costly. Every person has their own opinion about the taste, so identifying a quality based on a person’s taste is challenging. With the development of technology, the manufacturers started to rely on various devices for testing in development phases. So, they can have a better idea about wine quality, which, of course, saves lots of money and time. In addition, this helped in accumulating lots of data with various parameters such as quantity of different chemicals and temperature used during the production, and the quality of the wine produced. These data are available in various databases (UCL Machine Learning Repository, and Kaggle). With the rise of ML techniques and their success in the past decade, there have been various efforts in determining wine quality by using the available data [1] [2] [3]. During this process, one can tune the parameters that directly control the wine quality. This gives the manufacturer a better idea to tune the wine quality by tuning different parameters in the development process. Besides, this may result in wines with multiple tastes, and at last, may result in a new brand. Hence, the analysis of the basic parameters that determine the wine quality is essential. In addition to humanitarian efforts, ML can be an alternative to identify the most important parameters that control the wine quality. In this work, we have shown how ML can be used to identify the best parameter on which the wine quality depends and in turn predict wine quality.

# 2. PROBLEM STATEMENT

* The first goal is the study of the importance of the features for the prediction of wine quality.
* The second goal is to test various classification systems to discover which one provides the best results

# 3. WINE QUALITY PREDICTION: WORKFLOW

The following figure shows **WORKFLOW** of Wine Quality Prediction



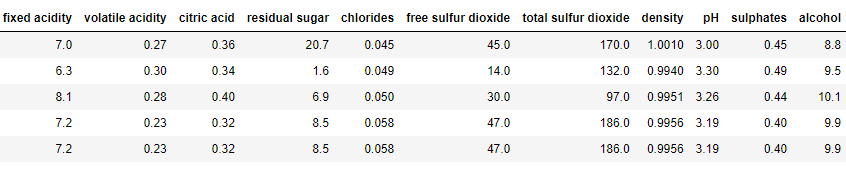
#### Figure 3.1 Wine Quality Prediction Workflow

# 4. DATASET DESCRIPTION

* The name of the data set used is: WINE QUALITY DATASET.
* Dataset that is used in our model is referred from Kaggle.com .
* This data will allow us to create different regression models to determine how different independent variables help predict our dependent variable, quality. Knowing how each variable will impact the wine quality will help producers, distributors, and businesses in the red wine industry better assess their production, distribution, and pricing strategy.
* The dataset contains a total of 12 variables, which were recorded for 14000 observations.

The quality of a wine is determined by 11 input variables:

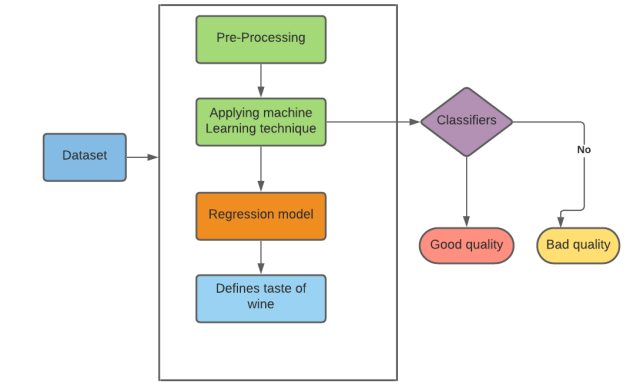
1. Alcohol: the amount of alcohol in wine
2. Volatile acidity: are high acetic acid in wine which leads to an unpleasant vinegar taste
3. Sulphates: a wine additive that contributes to SO2 levels and acts as an antimicrobial and antioxidant
4. Citric Acid: acts as a preservative to increase acidity (small 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)



#### Figure 4.1: Dataset

# 5. METHODOLOGY

The methodology consists of 5 stages to include; data collection, data preprocessing, use of machine learning algorithms& prediction of quality wine. Each stage is completed subsequently of one another.



#### Figure 5.1. Flowchart of Wine Quality Prediction

## 5.1 Data Collection

The process of gathering and analyzing accurate data from various sources to find answers to research problems, trends and probabilities, etc., to evaluate possible outcomes is Known as Data Collection.

Before an analyst begins collecting data, they must answer three questions first:

* What’s the goal or purpose of this research?
* What kinds of data are they planning on gathering?
* What methods and procedures will be used to collect, store, and process the information?

Additionally, we can break up data into qualitative and quantitative types.

* Qualitative data covers descriptions such as colour, size, quality, and appearance.
* Quantitative data, unsurprisingly, deals with numbers, such as statistics, poll numbers, percentages, etc.

Here, we have collected our data from WINE QUALITY DATA from Kaggle.com.

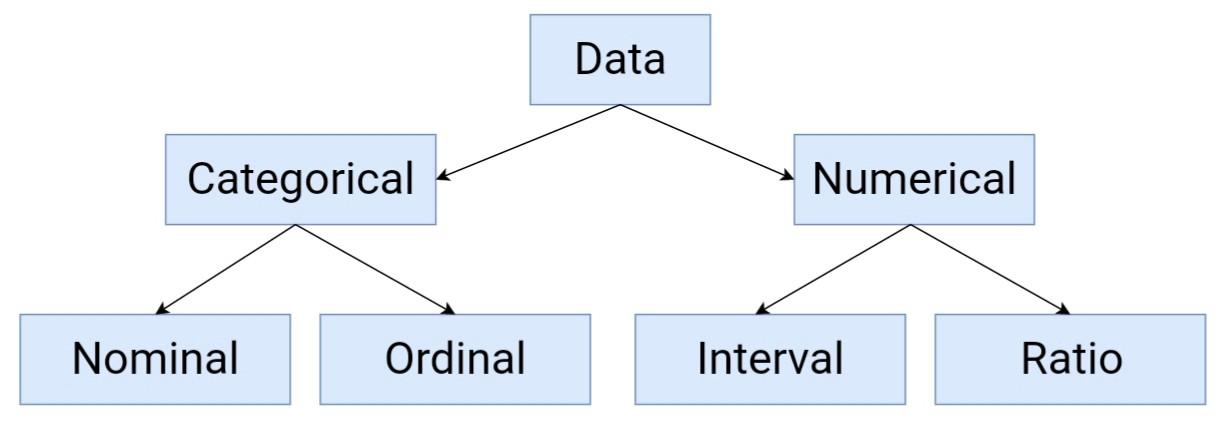
## 5.2 Data Preprocessing

Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning.

Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn’t have a regular, uniform design.

Machines like to process nice and tidy information – they read data as 1s and 0s. So, calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

Data Preprocessing however contains four steps through which preprocessing can be done.



#### Figure 5.2 Types of Data

### 5.2.1 Data Quality Assessment

Here we analyze the data and get a good idea of its overall quality, relevance to our project and consistency. There are a number of data anomalies and inherent problems to look out for in almost any data set, for example:

• **Mismatched data types:** When you collect data from many different sources, it may come to you in different formats. While the ultimate goal of this entire process is to reformat your data for machines, you still need to begin with similarly formatted data.

**• Data outliers:** Outliers can have a huge impact on data analysis results.

**• Missing data:** Take a look for missing data fields, blank spaces in text, or unanswered survey questions. This could be due to human error or incomplete data. To take care of missing data, you’ll have to perform data cleaning.

## 

### 5.2.2 Data Cleaning

It is the process of adding missing data and correcting, repairing, or removing incorrect or irrelevant data from a data set. Data cleaning is the most important step of preprocessing because it will ensure that your data is ready to go for your downstream needs.

It includes different steps for data cleaning:

Step1: Remove irrelevant data: Take a good look at your data and get an idea of what is relevant and what you may not need. Filter out data or observations that aren’t relevant to your downstream needs.

Step2: Duplicate our data: If we are collecting data from multiple sources or multiple departments, use scraped data for analysis, or have received multiple survey or client responses, you will often end up with data duplicates.

Duplicate records slow down analysis and require more storage. Even more importantly, however, if you train a machine learning model on a dataset with duplicate results, the model will likely give more weight to the duplicates, depending on how many times they’ve been duplicated. So, they need to be removed for well-balanced results.

Step3: Fix structural errors: Structural errors include things like misspellings, incongruent naming conventions, improper capitalization, incorrect word use, etc.

Step4: Deal with missing data: This could be due to incomplete data or human error. You’ll need to determine whether everything connected to this missing data – an entire column or row, a whole survey, etc.

Step5: Filter out data outliers: Outliers are data points that fall far outside of the norm and may skew your analysis too far in a certain direction.

Step6: Validate our data: Data validation is the final data cleaning technique used to authenticate your data and confirm that it’s high quality, consistent, and properly formatted for downstream processes.

### 5.2.3 Data Transformation

With data cleaning, we’ve already begun to modify our data, but data transformation will begin the process of turning the data into the proper format(s) you’ll need for analysis and other downstream processes.

This generally happens in one or more of the below:

• Aggregation: Data aggregation combines all of your data together in a uniform format.

• Normalization: Normalization scales your data into a regularized range so that you can compare it more accurately.

• Feature selection: Feature selection is the process of deciding which variables (features, characteristics, categories, etc.) are most important to your analysis.

• Discreditization: Discreditiization pools data into smaller intervals. It’s somewhat similar to binning, but usually happens after data has been cleaned.

• Concept hierarchy generation: Concept hierarchy generation can add a hierarchy within and between your features that wasn’t present in the original data.

# 6. DATA VISUALIZATION

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion.



#### Figure 6.1 Data Visualization

Here data visualization has been done using some different libraries as stated below. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

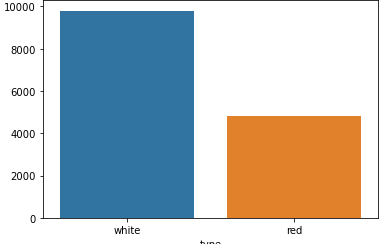
pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

Seaborn provides a variety of visualization patterns. It uses fewer syntax and has easily interesting default themes. It specializes in statistics visualization and is used if one has to summarize data in visualizations and also show the distribution in the data.

Here the different types of graphical representations are used such as line graph, displot, scatterplot etc for better visualization.

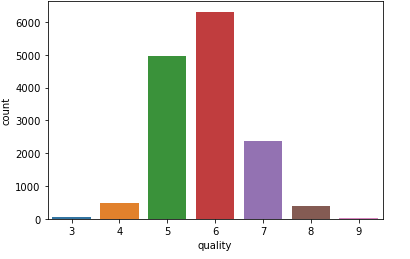
## 6.1 Exploratory Data Analysis

The below count plot shows the type of data which is white & red wine. We can see the data is classified with white wine count and red wine count.



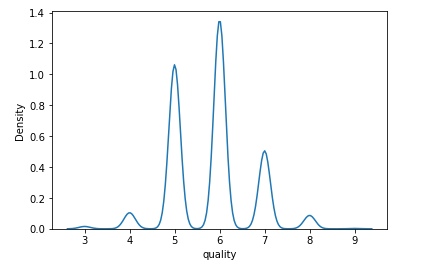
#### Figure 6.2 Count plot of type

On the basis of quality we build a bar plot.



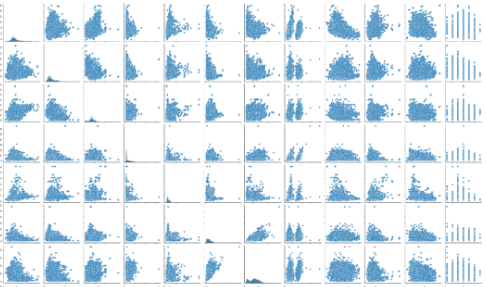
#### Figure 6.3 Count plot of Quality

Kdeplot is a Kernel Distribution Estimation Plot which depicts the probability density function of the continuous or non-parametric data variables



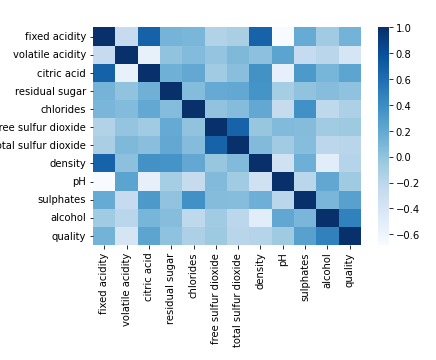
#### Figure 6.4 Quality vs Density(KDEplot)

We tried to build the scatter matrix of the wine dataset tovisualize the trends in our data.wth respect to each featurepresent in the dataset.



#### Figure 6.5 Pair plot

We build a graph which tells the correlation between the features of the dataset



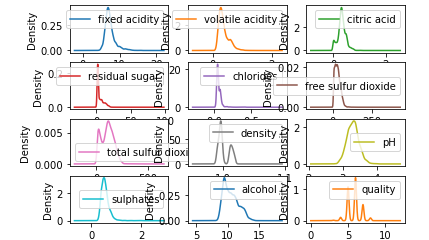
#### Figure 6.6 Heat Map

A heat map is extremely powerful way to visualize relationships between variables in high dimensional space. For example, in this case a correlation matrix with heat map colouring is shown below. A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable in the table is correlated with each of the other values in the

table. This allows us to see which pairs have the highest correlation.

The correlation count is between -1.0 to 1.0 The highest correlation is higher or nearly 1.0 and the lowest correlated features have correlation nearly -1.0.

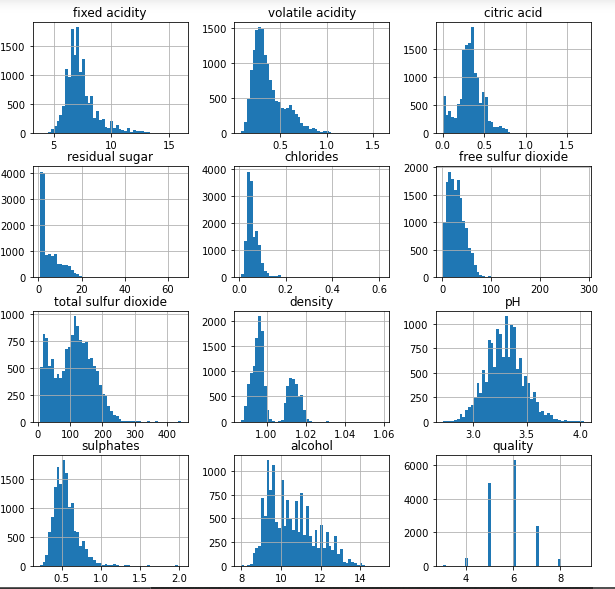
A density plot shows the distribution of a numerical variable. It takes only set of numeric values as input. It is really close to a histogram. To check the multiple distributions of data we visualized the density.



#### Figure 6.7 Density Plot

Histograms are good for showing general distributional features of dataset variables. You can see roughly where the peaks of the distribution are, whether the distribution is skewed or symmetric and if there are any outliers.

To check skewness and symmetry all the features are visualized with the help of Histogram Bins plot.



#### Figure 6.8 Histogram

# 7. FEATURE SELECTION & FEATURE SCALING

## 7.1 Feature Selection

In this section various feature variables in red white wine dataset are analyzed to detect the prevalent features to predict or assess the quality. Correlation tool is used select the dominant features based on the following facts. The variable fastened Acidity appears to possess virtually no impact on quality, Volatile Acidity shares a correlation with quality. The concentration of acid contributes the higher wine quality that direct correlation. An honest range against alcohol expected for quality wine. The lower concentration of Chloride appears to provide higher quality wines. Better wines used to be more acidic. And the residual sugar almost has no effect on wine quality. By analyzing the sample distribution in red and wine with nature of its independent variables the following feature are preferred to design the model. The important attributes are as follows acidity, sugar content, chlorides, sulfur, alcohol, pH and density.

## 7.2 Feature Scaling

For instance, the values of total Sulphur dioxide are extremely large compared to the chlorides. Such a large value of one variable can have dominance over other quantities during the training process in ML models. For instance, while doing K-nearest neighbor KNN [9], or SVM if one does not standardize the non uniform data, the datapoints with high distance will dominate the performance of the KNN or SVM model. So, feature scaling is a very important step one need to take care of, before training any ML model. There are many feature scaling methods. The most common and popular techniques that have been using in the ML community are standardization and normalization. There is not theoretical evidence of claiming which method work best. To scale the features of the dataset, normalization has been used. The formulas used to calculate the normalization is as follows:

Xn = (X - Xminimum) / ( Xmaximum - Xminimum)

Xn = Value of Normalization

Xmaximum = Maximum value of a feature

Xminimum = Minimum value of a feature

# 8. DATA PARTITION

Machine learning techniques are employed to predict wine quality in this study. The processes in the suggested methodology is depicted Pre-processing is done on the first wine dataset. The data is further divided into training (80%) and testing (20%) sets, with the training set being utilized to train the model utilizing Logistic Regression, Decision Tree, Random Forest, XGBoost, KNN, Gaussian Naïve Bayes and Support Vector Classifier algorithms. We train data and is used to find the relationship between target and predictor variables. The main purpose of the splitting data is to avoid overfitting. If overfitting occurs, the machine learning algorithm could perform exceptionally in the training dataset, but perform poorly in the testing dataset

The testing set is used to determine the accuracy of several models, and then conclusions are generated to choose the optimal model for predicting wine quality. The trained model is used to determine the testing set's correctness. The accuracy of various algorithms is assessed and compared in order to determine the optimum algorithm for predicting wine quality.

# 9. MACHINE LEARNING ALGORITHMS

## 9.1 Classification

Cataloguing is an information mining highlight that relegates objects to target classifications or classes inside a set. The arrangement objective is to anticipate the objective class precisely in the information for every function. A grouping model might be utilized, for instance, to order advance candidates as little, medium, or high credit chances. Arrangement errands start with an informational collection that knows the class tasks. Characterization is discrete and doesn't infer request. Nonstop, skimming point esteems will suggest an objective number rather than a clear cut one. A prescient model that has a mathematical objective uses a relapse calculation, not a calculation for order. The clearest kind of issue with order is a double grouping. The objective quality in paired characterization has just two potential qualities: high praise score or low praise assessment, for instance. Multiclass targets have multiple qualities: low, medium, high, and obscure FICO ratings, for instance. In the model build strategy (preparing), an arrangement calculation discovers connections between the indicator esteems and the objective qualities. Various calculations for the arrangement utilize explicit strategies to recognize connections. These connections are plot in a model that would then be able to be applied to another arrangement of information in which the class tasks are obscure.

Consequently, the goal of the proposed chapter is to predict the quality of the wine based on physicochemical tests through machine learning models. The upcoming sections precisely narrate the classification steps adopted by them in prediction.

## 9.2 Regression

Regression is a method for understanding the relationship between independent variables or features and a dependent variable or outcome. Outcomes can then be predicted once the relationship between independent and dependent variables has been estimated. Regression is a field of study in statistics which forms a key part of forecast models in machine learning. It’s used as an approach to predict continuous outcomes in predictive modelling, so has utility in forecasting and predicting outcomes from data. Machine learning regression generally involves plotting a line of best fit through the data points. The distance between each point and the line is minimized to achieve the best fit line.

Regression analysis is used to understand the relationship between different independent variables and a dependent variable or outcome. Models that are trained to forecast or predict trends and outcomes will be trained using regression techniques. These models will learn the relationship between input and output data from labelled training data. It can then forecast future trends or predict outcomes from unseen input data, or be used to understand gaps in historic data.

### 9.1.1 Logistic Regression

Logistic regression is one of the types of regression model where the regression analysis is executed when the dependent variable is binary. This regression method is used to explain the data and the relationship between the independent binary variable and one or more nominal, ratio-level independent variables.

The logistic regression analyses is widely used to determine the behavioral pattern of the individuals where they are able to predict an outcome. Most of the time the outcome can be ether yes or no. This regression analysis is widely used in each and every organization and based on the prediction made the decisions vary. Thus impacting the overall execution of the day to day process in an organization. It can also be used as a risk analysis tool.

### 9.1.2 K-Nearest-Neighbour

K-nearest neighbours (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems.

However, it is manly used for classification predictive problems in industry. The following two properties would define KNN well −

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

**Working of KNN Algorithm:**

K-nearest neighbours (KNN) algorithm uses „feature similarity‟ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following Calculate the distance between test data and each row of training data with the help of any of the method namely:

Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean. Now, based on the distance value, sort them in ascending order.

3.3 − Next, it will choose the top K rows from the sorted array.

3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 – End

### 9.1.3 Support Vector Classifier

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using LinearSVC or SGDClassifier instead, possibly after a Nystroem transformer or other Kernel Approximation.

The multiclass support is handled according to a one-vs-one scheme.

The advantages of support vector machines are:

* Effective in high dimensional spaces.
* Still effective in cases where number of dimensions is greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

* If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
* SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

The support vector machines in scikit-learn support both dense (numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. However, to use an SVM to make predictions for sparse data, it must have been fit on such data. For optimal performance, use C-ordered numpy.ndarray (dense) or scipy.sparse.csr\_matrix (sparse) with dtype=float64.

### 9.1.4 Decision Tree

Decision Tree is a supervised learning technique that can be used to solve problems in classification and regression. It is, however, mostly employed to solve categorization difficulties. It's a tree-structured classifier, with leaf nodes representing outcomes, interior nodes representing dataset attributes, and branches representing decision rules. Decision trees need less effort for data preparation during pre-processing than other methods. Data normalisation and scaling are not required when using a decision tree. In addition, missing values in the data have little impact on the decision tree-building process. A slight change in the data, on the other hand, can result in a significant change in the structure of the optimal decision tree. Calculations can become quite complicated, especially when multiple values are uncertain and/or multiple outcomes are related. The biggest problem with implementing a decision tree is figuring out how to make it work.

Information Gain= Entropy(S)-[(WeightedAvg) \*Entropy(each feature)

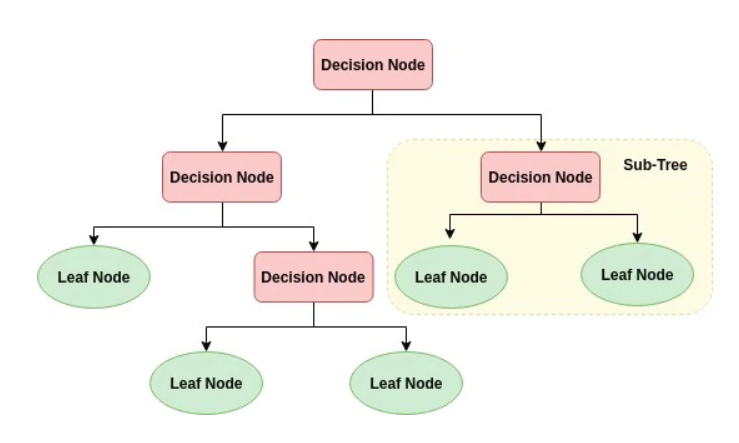
Entropy is a metric for determining the degree of impurity in a particular property. It denotes the randomness of data.

The Gini Index equation is as follows .

Gini Index= 1- ∑jPj²

The Gini index is a measure of impurity or purity used in the Classification and Regression Tree (CART) technique to create a decision tree.

Figure below depicts a decision tree, with internal nodes representing dataset attributes and branches representing decision rules.



#### Figure 9.1 Decision Tree

### 9.1.5 Random Forest

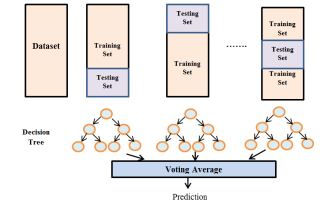
Random forest is a supervised learning method that can be used to solve problems in classification and regression. It creates a "forest" out of an ensemble of decision trees, which are commonly trained using the "bagging" method. The bagging method combines several learning models to improve the final outcome. Random Forest is a learning method that employs the construction of many decision trees to achieve its results. The random forest makes the final selection, which is based on the majority of the trees. Random Forest produces n number of decision trees by randomly selecting records from a dataset, as shown in Figure below. Instead than relying on a single decision tree, the random forest collects the forecasts from each tree and predicts the final output based on the majority votes of predictions. The greater the number of trees in the forest, the higher the accuracy and the less chance of overfitting.

The Random Forest algorithm has two stages: the first is to generate the random forest, and the second is to produce a prediction using the random forest classifier that was created in the first step.

1. Randomly select “K” features from total “m” features where k << m.
2. Among the “K” features, calculate the node “d” using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat the a to c steps until “l” number of nodes has been reached.
5. Build forest by repeating steps a tod for “n” number times to create “n” number of trees.

In the next stage, with the random forest classifier created, we will make the prediction.

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome.
2. Calculate the votes for each predicted target.
3. Consider the high voted predicted target as the final prediction from the random forest algorithm.



#### Figure 9.2 Random forest

### 9.1.6 XGBoost

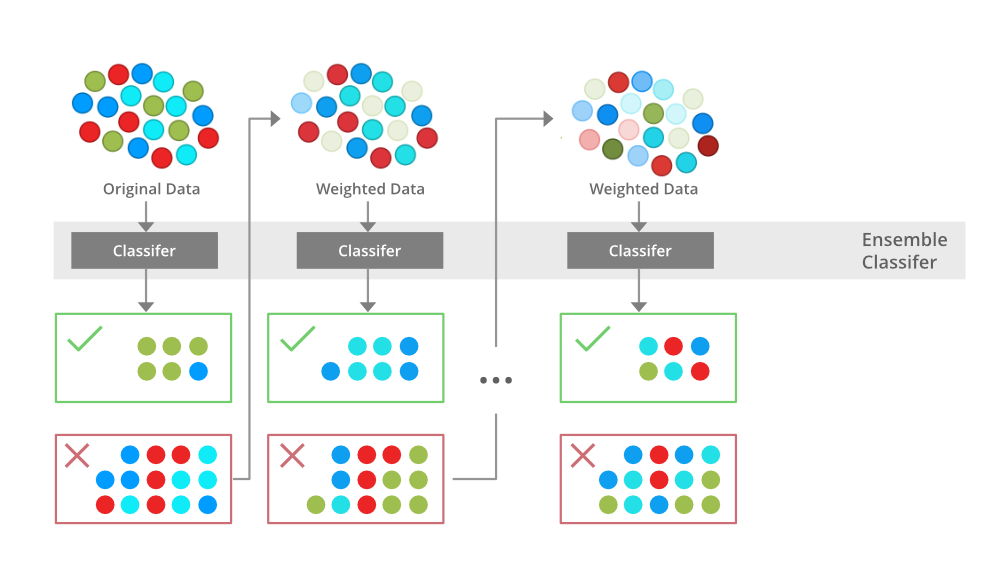
The XG Boost library implements the gradient boosting decision tree algorithm. This algorithm goes by lots of different names such as gradient boosting, multiple additive regression trees, stochastic gradient boosting or gradient boosting machines. Boosting is an ensemble technique where new models are added to correct the errors made by improvements can be made. A popular example is the AdaBoost algorithm that weights data points that are hard to predict.

Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction.

It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

This approach supports both regression and classification predictive modelling problems. Salient features of XG Boost which make it different from other gradient boosting algorithms include:

* Clever penalization of trees
* A proportional shrinking of leaf nodes
* Newton Boosting
* Extra randomization parameter



#### Figure 9.3 XG Boost

### 9.1.7 Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic algorithm used for classification tasks in machine learning. It is based on Bayes' theorem, which states that the probability of a hypothesis (in this case, a class label) given some observed evidence (in this case, a set of features) is proportional to the product of the prior probability of the hypothesis and the likelihood of the evidence given the hypothesis.

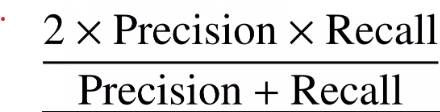
In Gaussian Naive Bayes, it is assumed that the features are independent and normally distributed, and thus the likelihood can be modeled using a Gaussian (normal) distribution. The algorithm calculates the probability of each class label for a given set of features, and assigns the label with the highest probability as the predicted label.

Gaussian Naive Bayes is a simple and efficient algorithm that works well for high-dimensional data and is particularly suited for text classification and spam filtering tasks. However, it may not perform well in cases where the assumption of feature independence or normal distribution is violated, or where there are correlated features.

# 10. EXPERIMENTAL RESULT, ANALYSIS & EVALUATION

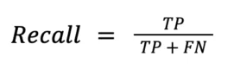
## 10.1 F1 Score

F1 score is an alternative machine learning evaluation metric that assesses the predictive skill of a model by elaborating on its class-wise performance rather than an overall performance as done by accuracy. F1 score combines two competing metrics- precision and recall scores of a model, leading to its widespread use in recent literature.



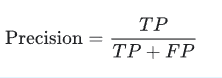
## 10.2 Recall

Recall, also known as the true positive rate (TPR), is the percentage of data samples that a machine learning model correctly identifies as belonging to a class of interest—the “positive class”—out of the total samples for that class. Machine learning recall is calculated on top of these values by dividing the true positives (TP) by everything that should have been predicted as positive (TP + FN). The recall formula in machine learning is:



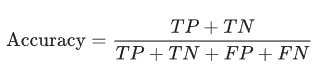
## 10.3 Precision

Precision is a performance metric used in binary classification problems. It measures the model's ability to predict the positive class accurately while minimizing the number of false positives. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model.



## 10.4 Accuracy

Accuracy is a performance metric used to measure how well a classification model is performing. It measures the percentage of correctly predicted instances out of all the instances in the dataset. It is defined as the ratio of the number of correct predictions to the total number of predictions made by the model.



## 10.5 Confusion Matrix

A confusion matrix is a table used to evaluate the performance of a classification model. It is a two-dimensional matrix that summarizes the predicted and actual classifications of a model on a set of data. The matrix contains four values:

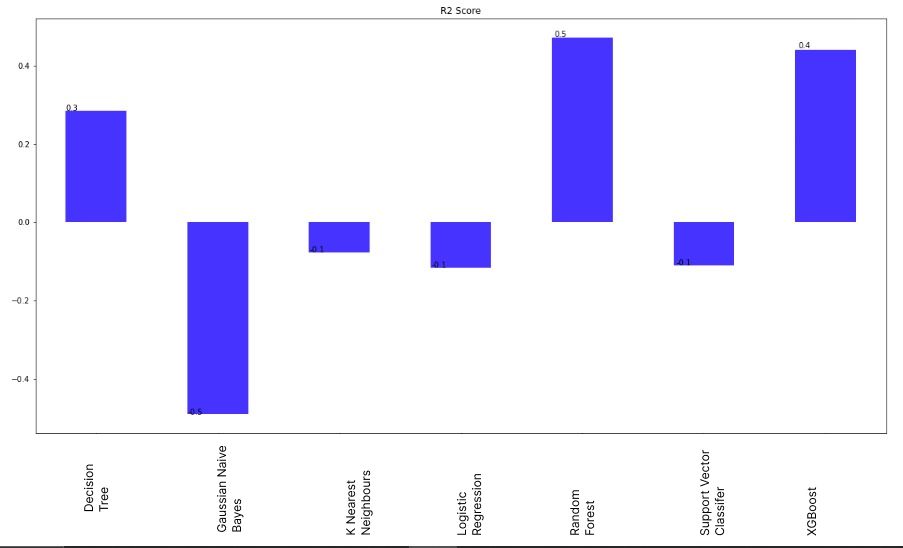
* True Positive (TP): The number of instances that the model correctly predicted as positive.
* False Positive (FP): The number of instances that the model incorrectly predicted as positive.
* True Negative (TN): The number of instances that the model correctly predicted as negative.
* False Negative (FN): The number of instances that the model incorrectly predicted as negative.

The confusion matrix is often used to calculate other performance metrics such as accuracy, precision, recall, and F1 score. It is particularly useful when dealing with imbalanced datasets, where the number of instances in each class is significantly different.

## 10.6 R²Score

R-squared (R²) is a statistical measure used to evaluate how well a regression model fits the data. It measures the proportion of variance in the dependent variable that is explained by the independent variables in the model. In other words, R² measures how well the model is able to capture the variation in the data.

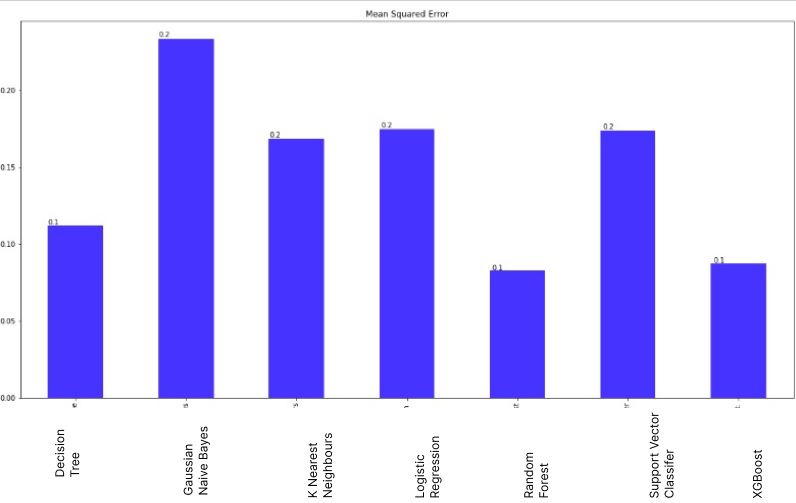
R² ranges from 0 to 1, with higher values indicating a better fit. A value of 1 indicates that the model perfectly fits the data, while a value of 0 indicates that the model does not explain any of the variance in the data.



#### Figure 10.1 R² Score

## 10.7 Mean Square Error

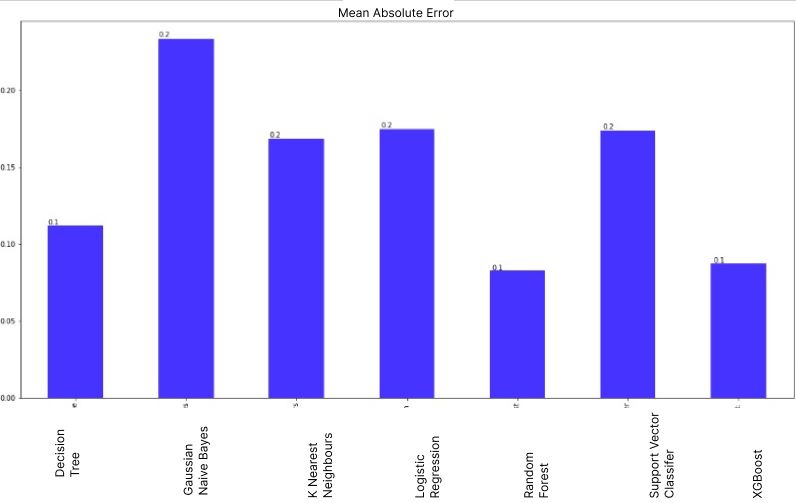
Mean Squared Error (MSE) is a commonly used metric to evaluate the performance of a regression model. It measures the average squared difference between the predicted and actual values of the dependent variable. It is perhaps the simplest and most common loss function, often taught in introductory Machine Learning courses. To calculate the MSE, you take the difference between your model’s predictions and the ground truth, square it, and average it out across the whole dataset.



#### Figure 10.2 Mean Square Error

## 10.8 Mean Absolute Error

Mean Absolute Error (MAE) is another commonly used metric to evaluate the performance of a regression model. It measures the average absolute difference between the predicted and actual values of the dependent variable. It is only slightly different in definition from the MSE, but interestingly provides almost exactly opposite properties! To calculate the MAE, you take the difference between your model’s predictions and the ground truth, apply the absolute value to that difference, and then average it out across the whole dataset.



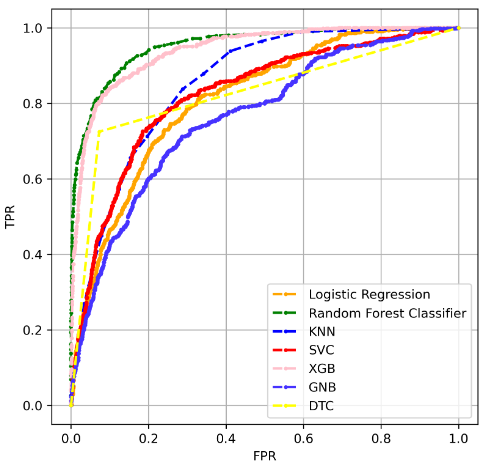
#### Figure 10.3 Mean Absolute Error

## 10.9 ROC AUC Score AND Curve

ROC (Receiver operating characteristic) curve and AUC (Area under the curve) score are commonly used evaluation metrics for binary classification models.

The ROC curve is a graphical representation of the trade-off between the true positive rate (TPR) and the false positive rate (FPR) for different classification thresholds. The TPR is also known as sensitivity or recall, while the FPR is equal to 1 - specificity. The ROC curve is obtained by plotting the TPR against the FPR at various threshold settings. The closer the ROC curve is to the top-left corner of the plot, the better the model is at distinguishing between positive and negative samples.

The AUC score is a single numeric value that summarizes the performance of a binary classification model over all possible threshold settings. The AUC score ranges between 0 and 1, with a higher score indicating better performance. An AUC score of 0.5 indicates that the model is no better than random guessing, while a score of 1 indicates perfect classification



#### Figure 10.4 ROC AUC Score AND Curve

# 11. CONCLUSION

This report uses the two types of wine dataset red and white, of Portuguese wine to predict the quality of the wine based on the physicochemical properties.

First, we used oversampling to balance the dataset in the data preprocessing stage to optimize the performance of the model. Then we look for features that can provide better prediction results. For this, we used Pearson coefficient correlation matrices and ranked the features according to the high correlation among the features. After applying the sampling datasets which is balancing dataset the performance of the model is improved. In general, removing irrelevant features of the datasets improved the performance of the classification model. To conclude that the minority classes of a dataset will not get a good representation on a classifier and representation for each class can be solved by oversampling and under sampling to balance the representation classes over datasets.

The accuracy of the Logistic Regression algorithm is 82.52%, Gaussian naïve Bayes (GNB) algorithm is 76.66%, K-Nearest Neighbors (KNN) is 83.16%, XGBOOST is 91.26%, Decision Tree is 88.80, Random Forest is 91.91% and Support Vector Classifier (SVC) is 82.26. Among these seven machine learning algorithms, we achieved the best accuracy result from the Random Forest.

Therefore, in the classification algorithms by selecting the appropriate features and balancing the data can improve the performance of the model.

# 12. FUTURE WORK

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

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

# REFERENCES

1. YunhuiZeng1, Yingxia Liu1, Lubin Wu1, Hanjiang Dong1. “Evaluation and Analysis Model of Wine Quality Based on Mathematical Model ISSN 2330-2038 E-ISSN 2330-2046, Jinan University, Zhuhai, China.
2. Paulo Cortez1, Juliana Teixeira1, Ant´onio Cerdeira2.“Using Data Mining for Wine Quality Assessment”.
3. YesimEr\*1, Ayten Atasoy1. “The Classification of White Wine and Red Wine According to Their Physicochemical Qualities”, ISSN 2147-67992147-6799, 3rd September 2016.
4. Dahal, K., Dahal, J., Banjade, H., Gaire, S., 2021. Prediction of Wine Quality Using Machine Learning Algorithms. Open J. Stat. 11, 278–289.
5. Er, Y., Atasoy, A., 2016. The Classification of White Wine and Red Wine According to Their Physicochemical Qualities. Int. J. Intell. Syst. Appl. Eng. 4, 23–26.
6. Er, Y., Atasoy, Ayten, 2016. The Classification of White Wine and Red Wine According to Their Physicochemical Qualities. Int. J. Intell. Syst. Appl. Eng. 23–26.