International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May - 2021 ISSN: 2582-3930

Identification of appropriate Machine Learning Algorithm to predict Wine Quality

Nilesh Korade[1], Mahendra Salunke[2]

[1]Department of Computer Engineering, PimpriChinchwad College Of Engineering and Research(PCCOER),Ravet, Pune,

[2]Department of Computer Engineering, PimpriChinchwad College Of Engineering and Research(PCCOER),Ravet, Pune,

---***---

Abstract -Nowadays to sell and promote product industries are using quality Certification. The traditional method of wine assessment was done by human expert which is time consuming process which makes assessment of wine very expensive. Several Machine Learning algorithms have already been used on wine quality datasets to evaluate wine attributes such as quality or class. The quality of wine does not depend only on the amount of alcohol, it also depends on different qualities, these attributes change with time and so the quality of wine is also refined. To determine quality of wine It is important to classify it into different categories on the basis of a quality attributes. To predict quality of wine different machine learning methods are used. This study presents comparative study of fundamental and technical analysis based on different attributes of wine. Different machine learning algorithms are compared to identify best suitable for prediction of wine quality. The algorithms include Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Ada Boost Classifier, and Gradient Boosting Classifier. With the help of visualization various machine learning techniques are compared based on methodologies, datasets, and efficiency.

***Key Words*:**Machine Learning, Wine Quality, Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Ada Boost Classifier, Gradient Boosting Classifier, Precision, Recall, F1-Score, Accuracy.

1.INTRODUCTION

The wine consumption has a positive correlation to the heart rate variability so in recent years there is a modest increase in the wine consumption. The wine industries are looking for best way to produce good quality wine at less cost due to the increase in the consumption wine. Although most of the chemicals are same for different type of wine based on the chemical tests, the quantity of each chemical have different level of concentration for different type of wine. Todays all types of industries adopting and applying new technology to ensure and test quality of product. To increase quality of product, testing is important phase which ensure quality of product. Testing a quality of product by human expertise is expensive and time consuming process requires some time to get an output. This study explores different machine learning techniques such as Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Ada Boost Classifier and Gradient Boosting Classifier for wine quality assurance. These techniques performs quality assurance process with the help of available characteristics of product and automate the process by minimizing human interfere. The work also identifies the important features to predict the values of dependent variables. Wine quality assessment is one of the key elements can be used for certification and such type of quality certification helps to assure wine quality in market. The input variable in white wine Data Set are fixed acidity, volatile acidity, citric acid, residual sugar chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol. The quality is in the range of 1 to 10, the higher value indicates better quality of wine.

1.1Objective

The first objective of this study is to apply different machine learning methods to see which yields the highest accuracy. Second objective of this study is to determine which features are the most indicative of a good quality wine.

2. Related Work

To predict wine quality most of researchers have used machine learning algorithms, but still a huge scope is available for improvement. Yogesh Gupta [1] paper explores the usage of linear regression to determine important features for prediction and usage of neural network and support vector machine in predicting the values. This paper explores the usage of machine learning techniques in two ways. Firstly, how linear regression determines important features for prediction. Secondly, the usage of neural network and support vector machine in predicting the values. Nikita Sharma [2] paper mentioned about predicting the quality of red wine using various machine learning algorithms. The feature select on algorithm provided a clear idea about the importance of the attributes for prediction of quality, which was time consuming and expensive when done in the traditional way. The accuracy of each technique used in prediction of quality is compared and it was found that these classifiers performed well. Terence Shin [3] used Kaggle's white wine quality dataset to build various classification models and predict whether a particular white wine is "good quality" or not. DevikaPawar et al. [4] used Logistic Regression, Stochastic gradient descent, Support Vector Classifier, Random Forest to predict the wine quality. The analysis shows that quality increases as residual sugar is moderate and does not have change drastically, so this feature is not essential as compared to others like alcohol and citric acid. Saini, A et al. [5] performed fundamental and technical analysis of various algorithms used for predicting future stock market prices and found that Long Short Memory Neural Network (LSTM NN) producing better results as compared to other techniques. YunhuiZeng et al. [6] This study takes wine quality evaluation as the research object, establishes the analysis and evaluation model of wine quality, and explores the influence of physical with chemical indicators of wine grapes and wine on the wine quality. This study performed multiple linear regression analysis of wine quality and finds that there is a positive correlation linear relationship between the scores of the aroma of wine quality and C_2H_6O, $C_6H_{12}O_2$

etc. Jambhulkar et al. [7] used various techniques to predict heart disease using wireless sensor network. To predict heart disease they collected data from Cleveland dataset and extracted important. Zaveri et al. [8] predicted different diseases like cancer, TB, diabetes etc. using data mining techniques. YesimEr et al. [9] evaluated different classifiers like k-nearest-neighborhood, random forests, and support vector machines on wine datasets. They applied principal component analysis and find that the success rate of quality classification for white wine has decreased for cross validation mode. The success rate of quality classification for white wine has decreased from for percentage split mode. The success rate of quality classification for white wine has increased for cross validation mode. The success rate of quality classification for white wine samples has increased for percentage split mode.

3.Proposed Methodology

In this study, different machine learning algorithms are used to predict white wine quality. The Figure 1 describes the processing in proposed methodology. First Wine dataset is pre-processed. Further, data is divided into training and testing sets, the training set is used to train model by using Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Ada Boost Classifier, Gradient Boosting Classifier algorithm. The testing set is used to find accuracy of different model, then conclusion is drawn to identify best model to predict quality of white wine. The trained model is used to identify accuracy of testing set. The accuracy of different algorithm is evaluated and compared to find best algorithm to predict quality of white wine.

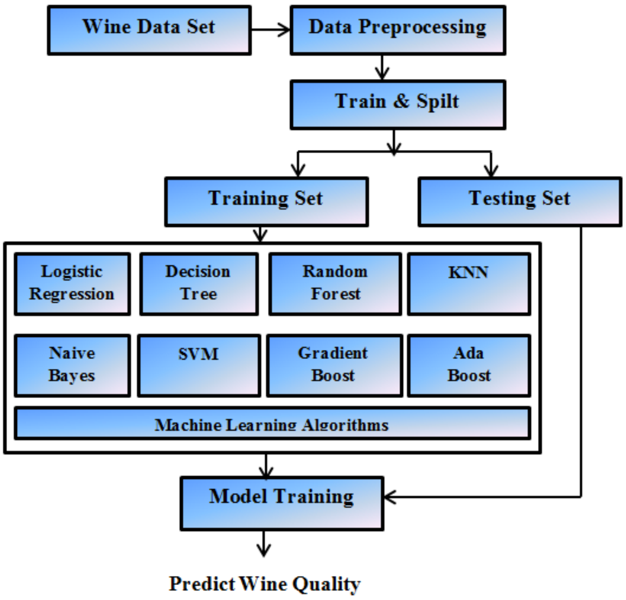


Fig-1: Propose Method to Predict Wine Quality

3.1Dataset

In this study collection of white wines [10] dataset is used. There are total *4898*samples for White Wine. Each sample consists of 12 physiochemical variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality rating. The quality classes range from 0 to 10 where 0 is very bad and 10 is very excellent. Due to some deficiencies in dataset, it is not possible to use wine collections without preprocessing. One of the major deficiencies is the large amplitude of variable values e.g. sulfates (0.3–2) vs. sulfur dioxide (1–72), there are some missing values. The Missing values are filled by taking mean. The inconsistency in dataset affects predictions due to influence making by some variables, such inconsistency is solve by Linear transformation, by dividing all the input values by maximum variable value. The wine quality is converted to a binary output '1' mean good quality wine having a score of 7 or higher and '0' mean bad quality wine having a score below 7.

3.2 Machine Learning Technique

The description of different machine learning algorithm used in this study is as below:

3.2.1 Logistic Regression is asupervised learning technique in machine learning algorithm. Logistic regression predicts the output of a categorical dependent variable and the outcome must be a categorical value. It can be 0 or 1, Yes or No, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. We know the equation of the straight line can be written as:

$y=b_0 + b_1\ x_1+b_2\ x_2+\ldots b_n x_n$

In logistic regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y).

y/(1-y); 0 for y=0 and infinity for y=1

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\frac{y}{1-y} = b0\ +\ b1\ x1 +\ b2\ x2 + \cdots\ bn\ xn.$$

The above equation is the final equation for Logistic Regression

3.2.2 Support Vector Machine is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. SVM constructs a hyperplane in a high or infinite dimensional space, which can be used for classification, regression or other tasks.

The objective of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. Hyperplane can be multiple lines or decision boundaries to segregate the classes in n-dimensional space, but to classify the data points we need to find out the best decision boundary. This best boundary is known as the hyperplane of SVM. Support Vectors are the data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. SVM chooses the extreme points or vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane. Figure2. Shows Support Vector Machine Hyperplane. The Support Vector divides the data points into two classes called as positive hyperplane and negative hyperplane.

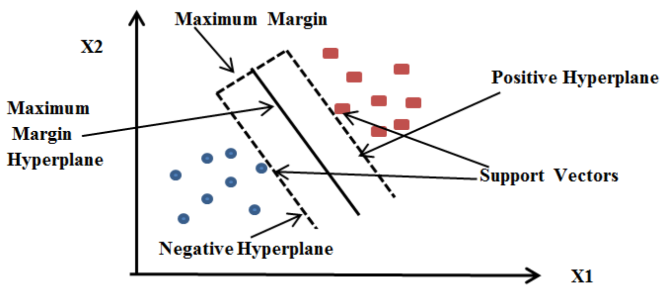


Fig-2: Support Vector Machine Classifier

3.2.3 Decision Tree is a supervised learning technique that can be used for both classification and Regression problems. However, it is mostly used for solving classification problems. It is a tree-structured classifier, where each leaf node represents the outcome, internal nodes represent the features of a dataset, and branches represent the decision rules. As compared to other algorithms decision trees requires less effort for data preparation during pre-processing. A decision tree does not require normalization of data and scaling of data as well. Missing values in the data also do not affect the process of building a decision tree to any considerable extent. But a small change in the data can lead to a large change in the structure of the optimal decision tree. Calculations can get very complex, particularly if many values are uncertain and/or if many outcomes are linked. The main issue while implementing a decision tree is that how to select the best attribute for the root node and for sub-nodes. The technique Attribute selection measure or ASM technique used to solve such problems. There are two popular techniques for ASM Information Gain and Gini Index.

Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data.

Gini Index= 1- $\sum_j P_j^2$

Gini index is a measure of impurity or purity used while creating a decision tree in the Classification and Regression Tree (CART) algorithm.

The decision tree is shown in figure3. where internal nodes represent the features of a dataset, and branches represent the decision rules.

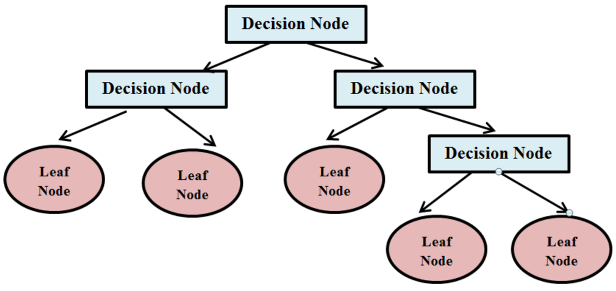


Fig-3: Decision Tree Classifier

3.2.4 Random forest is a supervised learning algorithm can be used for both classification and regression problems. The "forest" it builds, is an ensemble of decision trees, usually trained with the "bagging" method. The bagging method is a combination of learning models increases the overall result. Random Forest is a learning method that operates by constructing multiple decision trees. The final decision is made based on the majority of the trees and is chosen by the random forest. As shown in figure4 Random Forest creates n number of decision trees by randomly selecting records from dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

There are two stages in Random Forest algorithm, first is to create random forest and second is to make a prediction from the random forest classifier created in the first stage.

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.

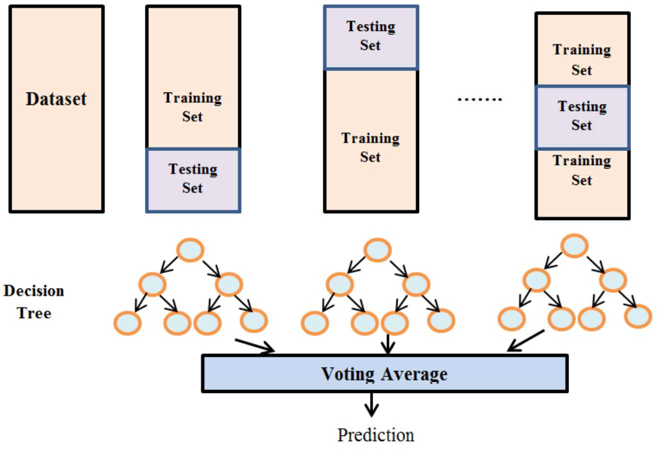


Fig-4: Random Forest Classifier

3.2.5 Ada BoostClassifiers is best used to boost the performance of decision trees on binary classification problems. AdaBoost algorithm, means Adaptive Boosting, is a boosting technique that is used as an ensemble method in machine learning. The weights are re-assigned to each instance, with higher weights to incorrectly classified instances. Boosting is used to reduce bias as well as the variance for supervised learning. It works on the principle where learners are grown sequentially.AdaBoost refers to a particular method of training a boosted classifier. A boost classifier is a classifier in the form $F_T(X)=\sum_{t=0}^{T} f_t(x)$

Where, f_t is a weak learner that takes an object x as input and returns a value indicating the class of the object. For example, in the two-class problem, the sign of the weak learner output identifies the predicted object class and the absolute value gives the confidence in that classification. The T^{th} classifier is positive if the sample is in a positive class and negative otherwise.An output hypothesis h(xi) is produced by each weak learner for each sample in the training set. At each iteration t, a weak learner is selected and assigned a coefficient

$\alpha_t$such that the sum training errorE_t of the resulting t-stage boost classifier is minimized.

$E_t=\Sigma E[F_{t-1}(x_i)+ \alpha_t h(x_i)]$

Where, Ft-1(xi) is the boosted classifier that has been built up to the previous stage of training. E(F) is some error function, $f_t(x)= \alpha th(xi)$ is the weak learner that is being considered for addition to the final classifier.

3.2.6 Gradient Boosting Classifiers are a group of machine learning algorithms. It creates a strong predictive model by combining many weak learning models together. Decision trees are usually used when doing gradient boosting. Gradient boosting classifiers are the AdaBoosting method combined with weighted minimization, after which the classifiers and weighted inputs are recalculated. The main objective of Gradient Boosting classifiers is to minimize the loss, or the difference between the actual class value of the training example and the predicted class value. The Gradient Boosting Classifier depends on a loss function. A custom loss function can be used, and many standardized loss functions are supported by gradient boosting classifiers, but the loss function has to be differentiable. Gradient boosting systems don't have to derive a new loss function every time the boosting algorithm is added, rather any differentiable loss function can be applied to the system. Classification algorithms frequently use logarithmic loss, while regression algorithms can use squared errors. Gradient boosting systems have two other necessary parts a weak learner and an additive component. Gradient boosting systems use decision trees as their weak learners. Regression trees are used for the weak learners, and these regression trees output real values. Because the outputs are real values, as new learners are added into the model the output of the regression trees can be added together to correct for errors in the predictions. A procedure similar to gradient descent is used to minimize the error between given parameters. This is done by taking the calculated loss and performing gradient descent to reduce that loss.

Gradient Boosting Classifiers Algorithm:

Input training set $\{x_i, y_i\}$ where i=1..n, a differentiable loss function L(y, F(x)), and M is number of iteration.

Step 1: Initialize model with constant value $F_0(x) = \text{argmin}\sum_{i=1}^{n} L(y_{i,}\gamma)$

Step 2: for m=1 to M

1. Compute so called pseudo residuals

$r_{im}= -[\frac{\delta L(yi,F(xi))}{\delta F(xi)}]_{F(x)=Fm-1(x)}$ for i=1..n.

2. Fit a base learner (or weak learner, e.g. tree) $h_m(x)$ to pseudo-residuals, i.e. train it using the training set{xi,yi} where i=1..n.

3. Compute multiplier γ_m by solving the following one-dimensional optimization problem

$\gamma_{m\,=}$ arg $\min\sum_{i=1}^{n} L(y_{i,}F_{m-1}(x_i) + \gamma h_m(x_i))$

4. Update the model

$F_m(x)= F_{m-1}(x_i) + \gamma_m h_m(x))$

Step 3: Output $F_M(x)$

3.2.7. K-Nearest Neighbour (KNN) Classifieris Supervised Learning technique which assumes the similarity between the new case and available cases and put the new case into the category that is most relevant to the available categories. o

KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data. The KNN Working steps are as below:

1. Load the data from dataset
2. Initialize K to your chosen number of neighbors
3. For each example in the data
    3.1 Calculate the distance between the query example and the current example from the data.
    3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

3.2.8 Naive Bayes Classifier is based on Bayes theorem and used for solving classification problems. Naive Bayes Classifier is one of the simple and most effective Classification algorithms which help in building the fast machine learning models that can make quick predictions.

Bayes' theorem is used to determine the probability of a hypothesis with prior knowledge is also known as **Bayes' Rule** or **Bayes' law**. It depends on the conditional probability. The formula for Bayes' theorem is as below:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability that is Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability that is Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability that is Probability of hypothesis before observing the evidence.

P(B) is Marginal Probabilit that is Probability of Evidence.

4. Experimental results and analysis

There are total *12* variables in white wine collections as discussed in above section. The variable quality rating is considered as dependent variable and other *11* variables are assumed as predictors or independent variables in this work. The distribution of the quality variable is shown in figure 2. The accuracy of different machine Learning algorithm is shown in below Tables.

The performance of the classification models for a given set of test data is drawn by using confusion matrix. It can only be determined if the true values for test data are known. In, information retrieval and classification in machine learning, precision is also called positive predictive value which is the fraction of relevant instances among the retrieved instances, while recall is also known as sensitivity which is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance. In statistical hypothesis testing, a type-I error is the rejection of a true null hypothesis is also known as a "false positive" finding or conclusion for example an innocent person is convicted, while a type-II error is the non-rejection of a false null hypothesis is also known as a "false negative" finding or conclusion for example a guilty person is not convicted. The different term used are described below:

Classification Accuracy: It defines how often the model predicts the correct output. It is one of the important parameters to determine the accuracy of the classification problems. The classification accuracy can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:

$$Classification\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:

$$Precision = \frac{TP}{TP + FP}$$

Recall: It is defined as the how our model predicted correctly out of total positive classes. The recall must be as high as possible.

$$Recall = \frac{TP}{TP + FN}$$

F-measure: It is difficult to compare models if two models have low precision and high recall or vice versa. The F-score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

True Negative (TN) is the model has given prediction No, and the real or actual value was also No. True Positive (TP) is the model has predicted yes, and the actual value was also true. False Negative(FN) is the model has predicted no, but the actual value was Yes, it is also called as Type-II error. False Positive (FP)is the model has predicted Yes, but the actual value was No. It is also called a Type-I error. The distribution of quality attribute and their count is shown in below figure 5.

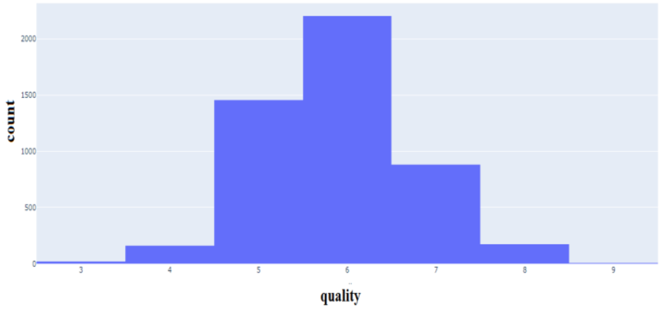


Fig-5: Distribution of the quality variable

4.1 Performance Matrix and Accuracy

The performance matrix and accuracy for Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Ada Boost Classifier, and Gradient Boosting Classifier is shown below.

Logistic Regression gave us an accuracy of 80%. The Performance Matrix for Logistic Regression Algorithm is shown in Table 1.

Table-1: Performance Matrix for Logistic Regression Algorithm

	precision	**recall**	**f1-score**	**support**
0	0.82	0.95	0.88	963
1	0.58	0.26	0.35	262
accuracy			0.80	1225
macro avg	0.70	0.60	0.62	1225
weighted avg	0.77	0.80	0.77	1225

Decision Tree Classifier gave us an accuracy of 82%. The Performance Matrix for Decision Tree Classifier Algorithm is shown in Table 2.

Table-2: Performance Matrix for Decision Tree Classifier Algorithm

	precision	**recall**	**f1-score**	**support**
0	0.89	0.88	0.88	963
1	0.57	0.58	0.58	262
accuracy			0.82	1225
macro avg	0.73	0.73	0.73	1225
weighted avg	0.82	0.82	0.82	1225

Random Forest Classifier gave us an accuracy of 86%. The Performance Matrix for Random Forest Classifier Algorithm is shown in Table 3.

Table-3: Performance Matrix for Random Forest Classifier Algorithm

	precision	**recall**	**f1-score**	**support**
0	0.88	0.95	0.92	963
1	0.76	0.54	0.63	262
accuracy			0.86	1225
macro avg	0.82	0.75	0.77	1225
weighted avg	0.86	0.86	0.86	1225

Support Vector Machine gave us an accuracy of 81%. The Performance Matrix for Support Vector Machine Algorithm is shown in Table 4.

Table-4: Performance Matrix for Support Vector Machine Algorithm

	precision	**recall**	**f1-score**	**support**
0	0.82	0.97	0.89	963
1	0.65	0.22	0.33	262
accuracy			0.81	1225
macro avg	0.74	0.59	0.61	1225
weighted avg	0.78	0.81	0.77	1225

Ada Boost Classifier gave us an accuracy of 80%. The Performance Matrix for Ada Boost Classifier Algorithm is shown in Table 5.

Table-5: Performance Matrix for Ada Boost Classifier Algorithm

	precision	**recall**	**f1-score**	**support**
0	0.84	0.92	0.88	963
1	0.54	0.34	0.42	262
accuracy			0.80	1225
macro avg	0.69	0.63	0.65	1225
weighted avg	0.77	0.80	0.78	1225

Gradient Boosting Classifier gave us an accuracy of 83%. The Performance Matrix for Gradient Boosting Classifier Algorithm is shown in Table 6.

Table-6: Performance Matrix for Gradient Boosting Classifier Algorithm

	precision	**recall**	**f1-score**	**support**
0	0.85	0.95	0.90	963
1	0.66	0.39	0.49	262
accuracy			0.83	1225
macro avg	0.76	0.67	0.69	1225
weighted avg	0.81	0.83	0.81	1225

K-nearest neighbors (KNN) algorithm gave us an accuracy of 82%. The Performance Matrix for Gradient Boosting Classifier Algorithm is shown in Table 7.

Table-7: Performance Matrix for K-nearest neighbors (KNN) algorithm

	precision	**recall**	**f1-score**	**support**
0	0.87	0.91	0.89	963
1	0.59	0.50	0.54	262
accuracy			0.82	1225
macro avg	0.73	0.70	0.71	1225
weighted avg	0.81	0.82	0.81	1225

Naive Bayes Classifier gave us an accuracy of 74%. The Performance Matrix for Gradient Boosting Classifier Algorithm is shown in Table 8.

Table-8: Performance Matrix for Naive Bayes Classifier algorithm

	precision	**recall**	**f1-score**	**support**
0	0.90	0.75	0.82	963
1	0.43	0.70	0.53	262
accuracy			0.74	1225
macro avg	0.67	0.72	0.67	1225
weighted avg	0.80	0.74	0.76	1225

4.2Comparison of Accuracy by different Algorithm

The different machine learning algorithm like Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Ada Boost Classifier, and Gradient Boosting Classifier are train using physiochemical variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol variables. The variable quality rating is considered as dependent variable and other 11 variables are assumed as predictors or independent variables in this work. Two types of analysis are done in this study firstly, the importance of each algorithm to predict wine quality is identified and secondly, the feature selction is done using best predictors. The accuracy provided by different algorithm is shown in below table. The Logistic Regression gives 80%accuracy; Decision Tree Classifier gives 82% accuracy and so on. The Random Forest Classifier Algorithm provides highest accuracy 86%.

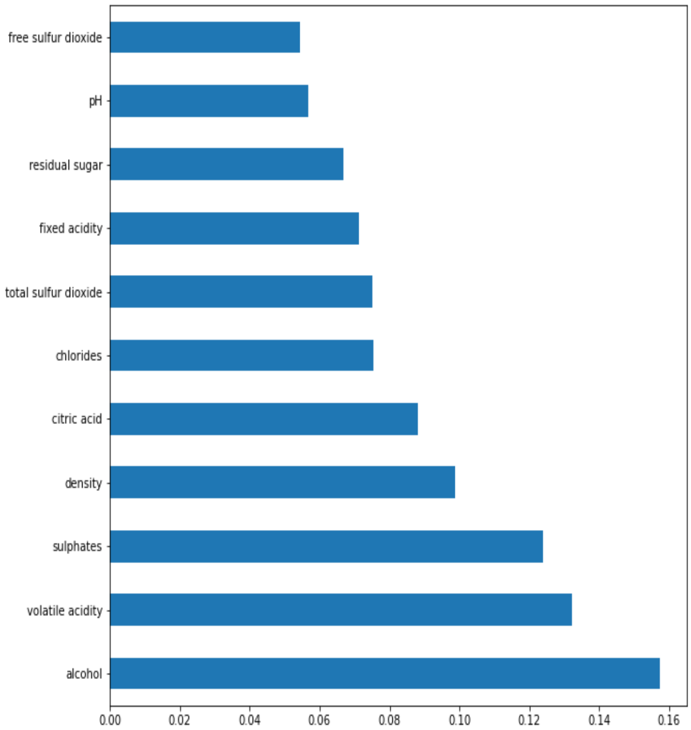
Table-9: Comparison of Accuracy obtained by different Algorithm

Sr. No.	**Algorithm**	**Accuracy**
1	Logistic Regression Algorithm	80%
2	Decision Tree Classifier Algorithm	82%
3	Random Forest Classifier Algorithm	86%
4	Support Vector Machine Algorithm	81%
5	Ada Boost Classifier Algorithm	82%
6	Gradient Boosting Classifier Algorithm	83%
7	K-nearest neighbors (KNN) algorithm	82%
8	Naive Bayes Classifier algorithm	74%

4.3Comparison of Accuracy by different Algorithm

Below graph shows feature importance based on the Random Forest model. The 11 physiochemical variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol are considered to predict quality of wine. To predict quality of wine it is not necessary to consider all features, we can remove some of the features while training our model like free sulfur dioxide, pH, residual sugar, fixed acidity etc. The top 3 features are alcohol, volatile acidity, and sulphates.

Charts 1: Feature importance based on the Random Forest model



5. CONCLUSIONS

In wine industry the interest has been increasing in recent years which demands growth in this industry. Therefore, to improve wine production and selling, companies are investing in new technologies. Wine quality certification plays a very important role to sell product in market and it requires wine testing by human experts. This study explores different machine learning technique to predict the quality of wine. For each classification model, how the results vary whenever test mode is changed is shown in this study. The study includes the analysis of classifiers on white wine datasets. The results are described in percentage of correctly classified instances, precision, recall, F measure. Different classifiers like Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Ada Boost Classifier, and Gradient Boosting Classifier are evaluated on datasets. Results from the experiments lead us to conclude that Random Forests Algorithm performs better in classification task as compared against to other classifiers. The prediction of wine quality achieves maximum accuracy of 92% using Random Forest Algorithm. We can see that good quality wines have higher levels of alcohol on average, higher levels of sulphates on average, have a lower volatile acidity on average, and higher levels of residual sugar on average. The study shows that instead of considering all features, the value of dependent variable can be predicted more accurately if only important features are considered in prediction. In future, large dataset can be taken for study and other machine learning techniques may be explored for wine quality prediction.

REFERENCES

[1] Yogesh Gupta," Selection of important features and predicting wine quality using machine learning techniques", Procedia Computer Science 125 (2018) 305–312.

[2] Nikita Sharma, "Quality Prediction of Red Wine based on Different Feature Sets Using Machine Learning Techniques", International Journal of Science and Research, Volume 9, Issue 7, 1358-1366, July 2020, ISSN: 2319-7064.

[3] Terence Shin, "Predicting Wine Quality with Several Classification Techniques", May 8, 2020, URL:"https://towardsdatascience.com/predicting-wine-quality-with-several-classification-techniques-179038ea6434".

[4] DevikaPawar, AakankshaMahajan, SachinBhoithe, "Wine Quality Prediction using Machine Learning Algorithms", International Journal of Computer Applications Technology and Research, Volume 8–Issue 09, 385-388, 2019, ISSN:-2319–8656.

[5] Saini, A., Sharma, A., "Predicting the Unpredictable: An Application of Machine Learning Algorithms in Indian Stock Market", *Ann. Data. Sci.* (2019). https://doi.org/10.1007/s40745-019-00230-7.

[6] YunhuiZeng, Yingxia Liu, Lubin Wu, Hanjiang Dong, Yuanbiao Zhang, HongfeiGuo, ZishengGuo, Shuyang Wang, Yao Lan, "Evaluation and Analysis Model of Wine Quality Based on Mathematical Model", Studies in Engineering and Technology Vol. 6, No. 1; August 2019, ISSN 2330-2038 E-ISSN 2330-2046 .

[7] Jambhulkar and Baporikar. (2015) "Review on Prediction of Heart Disease Using Data Mining Technique with Wireless Sensor Network". *International Journal of Computer Science and Applications* 8 (1) 55-59.

[8] Zaveri, and Joshi. (2017) "Comparative Study of Data Analysis Techniques in the domain of medicative care for Disease Predication". *International Journal of Advanced Research in Computer Science* 8 (3) 564-566.

[9] YesimEr, AytenAtasoy. "The Classification of White Wine and White Wine According to Their Physicochemical Qualities", International Journal of Intelligent Systems and Applications in Engineering, ISSN 2147-67992147-6799, December 2016.

[10] URL: https://www.kaggle.com/uciml/white-wine-quality-cortez-et-al-2009