International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 5, 1621-1639, 2025
https://doi.org/10.33889/IJMEMS.2025.10.5.077

Ram Arti
Publishers

Ensemble Learning-Based Wine Quality Prediction Using Optimized Feature Selection and XGBoost

Sonam Tyagi
Department of Electronics & Communication Engineering,
ABES Engineering College, 201009, Ghaziabad, Uttar Pradesh, India.

Ishwari Singh Rajput
Department of Computer Science & Engineering,
Graphic Era Hill University, 263139, Haldwani, Uttarakhand, India.
Corresponding author: ishwarirajput@gehu.ac.in

Bhawnesh Kumar
Department of Computer Science & Engineering,
Graphic Era Deemed to be University, 248002, Dehradun, Uttarakhand, India.

Harendra Singh Negi
Department of Computer Science & Engineering,
Graphic Era Deemed to be University, 248002, Dehradun, Uttarakhand, India.

(Received on February 17, 2025; Revised on April 16, 2025 & May 28, 2025; Accepted on June 2, 2025)

Abstract

Recent years have seen the wine business flourish and become essential to the global economy. Due to rising demand for high-end wines, methods to consistently measure wine quality are needed. In this discipline, machine learning decision-making shows potential. High-dimensional data from several sources may impede processing and classification model performance. Feature selection increases learning and reduces computational costs by picking subsets of features and deleting irrelevant ones. This hybrid architecture uses machine learning to predict wine quality, including a feature selection method to find the most important information. Wrapper-based genetic algorithm (WGA) iteratively removes least significant features and trains a model with the remaining features until the needed number is obtained. We tested the proposed structure on two datasets consisting of 1,599 red wine samples and 4,898 white wine samples, each rated on a scale of 1–10. Additionally, the proposed technique is compared to other cutting-edge machine learning models in the same field. All categorization techniques predicted wine quality well, but WGA-XGB performed best. The study shows how feature selection improves wine quality prediction in different machine learning algorithms. The recommended strategy can be extended to different wine datasets or improved using advanced feature selection or machine learning models to improve forecast performance. Overall, the methodology is designed to be clear, relevant, and reproducible for assessing wine quality in real-world settings.

Keywords- Wine quality, Feature selection, Machine learning, Classification, Prediction.

1. Introduction

The act of consuming wine is a commonly observed phenomenon on a global scale, and its cultural significance is widely acknowledged. In the present competitive market, the quality of wine is of paramount importance to both producers and consumers, as it serves as a means to augment revenue (Bhardwaj et al., 2022). Historically, the evaluation of wine excellence has been carried out via assessments conducted after the production process. This approach necessitates a substantial commitment of both temporal and monetary resources to attain the desired level of quality. If the quality is deemed inadequate, it may be necessary to commence a new set of procedures, resulting in considerable expenses (Chiu et al., 2021). Evaluating the calibre of a product through an individual’s taste can pose a formidable challenge, given that taste-related viewpoints are subjective and can exhibit significant variability across individuals. The utilization of varied testing instruments during the developmental phases has become a prevalent practice among manufacturers, attributable to the progressions in technology. Individuals can optimize their time and financial resources by effectively improving their comprehension of wine quality (da Costa et al., 2021). In addition, this enabled the acquisition of extensive data covering various parameters such as the number of different chemicals and the temperature utilized in the manufacturing process, along with the resulting quality of the wine.

The recent advancements in Industry 4.0 have brought forth a plethora of pioneering technologies. Machine learning is a subfield of artificial intelligence that offers advanced technological solutions for addressing problems across various application domains. The wine quality prediction is a crucial domain that can harness the capabilities of predictive analytics technologies to a significant extent. Various classification techniques based on machine learning are employed to categorize data into distinct groups, utilizing the available features (Dahal et al., 2021). Despite the potential advantages of small datasets, they also present unique challenges to model design, including the risk of overfitting, which can impede the model's ability to generalize effectively. Similarly, the inclusion of a multitude of features in the dataset can result in the aforementioned issue. The inclusion of extraneous and disruptive attributes within a given dataset can significantly affect the efficacy of a model and elevate levels of classification ambiguity. Hence, the selection of an optimal feature set holds equal significance to the identification of an optimal classification model. The utilization of an optimal set of features has the potential to enhance classification accuracy and expedite processing by mitigating overhead and processing time (Al-Shammary et al., 2022). In order to develop an optimal classification approach, it is necessary to employ a range of configuration settings, with the customization of hyperparameters being a crucial factor. The search space for feature combinations is potentially unbounded, and exhaustive exploration of the entire tree is a computationally expensive and resource-intensive task (Krishnaveni et al., 2022). Thus, the manual adjustment of a classification model is a highly challenging endeavour that necessitates a thorough comprehension of the model. Consequently, the task of automatically optimising hyperparameters is highly laborious. Diverse methodologies have been documented in the academic literature, each presenting distinct benefits and drawbacks.

In this article, we have designed a predictive model to predict the quality of wine. The proposed WGA-XGB classification model is trained on the Red and White wine dataset to calculate various performance metrices. For improving the classification accuracy, WGA is used which selects the optimal and non-redundant features from the wine dataset. The WGA selects 7 non-redundant features out of 13 features available in original dataset, which are used to train the proposed classification model and other machine learning models such as decision tree, support vector machine and KNN.

The major contributions of our proposed method are summarized as follows:

- We present a comprehensive study on wine quality prediction, emphasizing the role of data-driven methods in improving classification accuracy.
- We implement a wrapper-based Genetic Algorithm for feature selection, effectively reducing dataset complexity while retaining essential attributes-an approach less explored in prior wine quality studies.
- We benchmark the performance of several machine learning classifiers and demonstrate that XGBoost achieves superior accuracy and robustness on the selected features.
- We discuss the practical applicability of wine quality prediction models in the wine industry for enhancing quality control and minimizing production costs.

The remaining sections of this work are structured as follows. In Section 2, we review previous efforts to predict wine quality using machine learning methods. Section 3 describes the dataset used in this study and the data preprocessing steps. In Section 4, we detail the feature selection algorithms that is implemented in this research. The machine learning algorithms utilized for predicting wine quality are discussed in Section 5. The findings and analysis from the experiments are presented in Section 6. The study concludes with Section 7 where potential paths of research are discussed.

2. Literature Review

The history of wine is very old. It is consumed since ancient times. In present scenario, due to change in lifestyle, it is very necessary to look after the quality of the product before consuming due to health concern. This criterion applies on wine also. Since the consumption of wine is high at now a day, it is not feasible to take advice from experts because it is costly, takes time, and suggested results may also not be as per consumer's taste. So, to overcome these problems, machine learning approach is used. It is an emerging technology which is used now a days in solving various real-life problems. It is also very trusted approach because it showed higher accuracy, less computing time and many more in various type of problems. Hence, it may also be used for predicting wine quality by selecting important features.

This section consists literature of some of the existing methods which were used for predicting red and white wine quality. Mahima et al. (2020) suggested a technique to predict the wine quality. Here they categorized the quality into good, average, and bad by employing machine learning models which were Random Forest and k-NN. To obtain the high accuracy in prediction they used k-NN along with RF. To predict the wine quality, Dahal et al. (2021) suggested an approach which employed several machine learning models which were Artificial Neural Network (ANN), Gradient Boosting Regressor (GBR), Ridge Regression (RR), and Support Vector Machine (SVM). The quantitative analysis showed that Gradient Boosting Regressor outperformed all the methods. Chiu et al. (2021) implemented an approach where they utilized the concept of Genetic Algorithm for predicting wine quality. With the implemented system, a hybrid collection of classifiers and associated hyperparameters may be found automatically to optimise the prediction result. da Costa et al. (2021) suggested a method where they first used feature selection and then classification. For feature selection they used Random Forest Importance, Correlation-based Feature selection and for classification Support Vector Machine ML model was utilized. Bhardwaj et al. (2022) predicted the quality of wine using Machine Learning approach. Here they used SMOTE technique to generate the samples, feature selection technique was applied on that generated samples and then several machine learning algorithms were employed on those features. Out of all, AdaBoost classifiers outperformed all techniques. Mohana et al. (2023) suggested an ensemble approach which consists Support Vector Machine, Random Forest, XGBoost, and Gradient Boosting models for predicting quality of red wine. Jana et al. (2023) implemented an approach which used neural network approach and support vector methods to predict wine quality. Out of which Quadratic SVM showed highest accuracy with lowest training time. Gupta (2018) proposed a two-way approach where they first selected important features using linear regression approach and after selecting features, they used neural networks and SVM for predicting the quality of red and white wine. Kumar et al. (2020) implemented a Machine Learning approach in which they applied SVM, Naive Bayes and Random Forest methods for predicting the quality of red wine, out of which SVM showed the best accuracy.

Chhikara et al. (2023) implemented a number of classifiers on the red wine data, including KNN, Random Forest, Decision Tree, Support Vector Machine, and Logistic Regression. The Random Forests Algorithm outperforms other classifiers in classification tasks and accurately predicted wine quality. John et al. (2024) examined the elements that affected red wine quality and how they led to favourable outcomes. They sought to identify the key characteristics that influence the quality of red wine by classifying and evaluating it using machine learning techniques like SVM and KNN. In order to optimize the parameters, grid search methods were also utilized. The KNN model achieved lower accuracy in the red wine datasets, in comparison to SVM model.

The motivation to use machine learning and feature selection for wine prediction is to improve the accuracy and efficiency of predicting various characteristics of wines. Predicting the quality of a wine is difficult because of many variables involved that affect the quality and taste of wine, such as grape variety, vintage year, weather conditions, and fermentation processes. Machine learning algorithms can learn patterns and relationships in large datasets of wine samples, and use this knowledge to make predictions about various aspects of wine, such as its quality, aroma, and taste. By selecting the most relevant features or variables that contribute to wine quality, feature selection techniques can help to improve the accuracy and efficiency of wine prediction models. The use of machine learning and feature selection for wine prediction can also have practical applications for the wine industry, such as optimizing wine production processes, improving the consistency and quality of wine products, and assisting with wine selection and recommendation for consumers.

3. Feature Selection Algorithms

In machine learning, feature selection plays a vital role as it can significantly improve the accuracy of the resulting models while reducing the computational complexity. The process entails picking out the most important characteristics from the whole set of features, which is often high-dimensional, noisy, and redundant. The features chosen should be able to reveal the data’s hidden structure and relationships, and eliminate irrelevant or redundant information (Cai et al., 2018; Gupta et al., 2022).

There are various feature selection algorithms, and each has its strengths and weaknesses (Xue et al., 2016). Filter approaches, wrapper approaches, and embedding approaches are the three most popular classifications for these techniques. Filter technique evaluates each feature’s importance and chooses the features with the highest scores. Wrapper methods evaluate subsets of features by training and testing a machine learning model on them, and select the subset that achieves the best performance. Selecting the appropriate features during optimisation, embedded approaches include feature selection into the model training process.

The choice of the feature selection algorithm depends on various factors, such as the size and nature of the dataset, the machine learning task, and the computational resources available. In this paper, we are using wrapper based genetic algorithm for feature selection which is described in the subsequent section.

3.1 Wrapper-based Feature Selection

Wrapper-based feature selection (WFS) uses a machine learning algorithm to evaluate subsets of features. It generates a set of candidates feature subsets and trains a model on each subset. These steps are repeated until some stopping criteria is reached or the required performance is achieved. The best feature subset is then selected based on the learning performance of the model (Al-Yaseen et al., 2022). The working of the wrapper-based feature selection is shown in **Figure 1**. WFS is computationally expensive but can potentially find the best feature subset for a given machine learning algorithm.

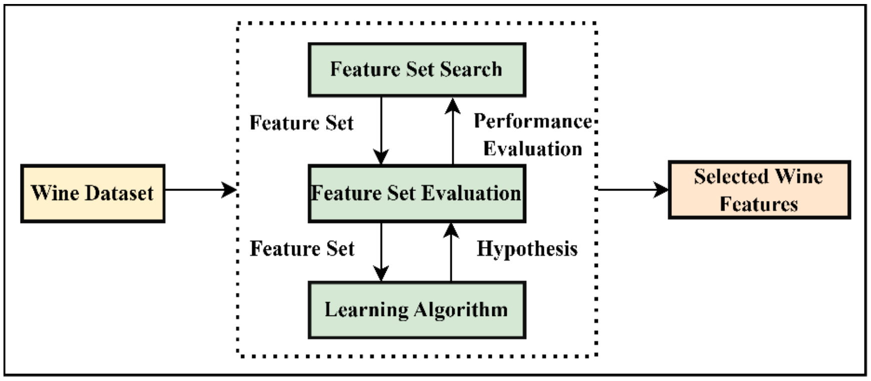


Figure 1. Wrapper approach for feature selection.

3.2 Wrapper based Genetic Algorithm for Feature Selection

The genetic algorithm (Al-Shammary et al., 2022) initiates by stochastically generating a population of prospective solutions (Zhou & Hua, 2022). The representation of chromosomal genes in the wine dataset varies depending on their nature and features, and may be in the form of a bit, a character, or a number (Huang et al., 2007). Individuals are assessed according to their fitness function. Subsequently, the population undergoes the application of selection and recombination operators with a probability of 0.6, aimed at exploring novel solutions within the search space. The mutation operator, which has a probability of 0.033, is utilised to perform random modifications aimed at refining the solutions. As illustrated in Algorithm 1, the procedure iterates until it reaches a predetermined maximum number of generations, which in this instance is 25. After undergoing dimension reduction, the wine dataset was ultimately reduced to contain 7 features.

Algorithm 1 Wrapper based genetic algorithm for feature selection

```
1: Input: Dataset; Population Size: N; Maximum no. of iterations: $T_{max}$
2: Output: Relevant Subset Features
3: Randomly generate initial solution P(0).
4: for k = 1 to N do
5:      Evaluate fitness of individuals of P(0);
6: end for
7: t=0;
8: while P(t<$T_{max}$) do
9:      Select two individuals in P(t) based on their fitness
10:     Apply crossover to selected individuals from step 9 resulting to new individuals
11:     Apply mutation operator to new individuals from step 10
12:     for k = 1 to N do
13:         Evaluate fitness of individuals of P(t);
14:     end for
15:     $t = t + 1$;
16: end while
17: Return individual with highest fitness.
```

4. Machine Learning Algorithms

4.1 Support Vector Classifier

The Support Vector Classifier (SVC) is a supervised learning algorithm primarily used for binary classification tasks. It aims to find the optimal separating hyperplane that maximizes the margin between two classes (Anguita et al., 2005). This hyperplane is defined by the equation $w^T x + b = 0$, where w is the weight vector, b is the bias term, and x is the feature vector (Tax & Duin, 2004). SVC performs well in high-dimensional spaces and is effective when the number of dimensions exceeds the number of samples. It can also utilize different kernel functions to handle non-linearly separable data, enhancing its flexibility across various domains.

The optimization problem for finding the hyperplane that maximizes the margin can be mathematically represented as (Anguita et al., 2005):

$$\min_{w,b} \frac{1}{2} |w|^2 \tag{1}$$

subject to $y_i(w^T\, x_i + b) \geq 1$ for $i = 1, 2, \ldots, n$ where, $|w|^2$ is the Euclidean norm of the weight vector w and n is the total number of instances in the training set. The optimization problem is solved using quadratic programming.

4.2 Decision Tree Classifier

The Decision Tree Classifier is a widely-used, interpretable model that predicts outcomes by learning decision rules inferred from input features (Avros et al., 2017). It represents data using a tree-like structure, where internal nodes denote tests on features, branches represent outcomes of tests, and leaves correspond to class labels (Swain & Hauska, 1977).

Decision trees handle both numerical and categorical data and can model complex decision boundaries. They are prone to overfitting, but techniques like pruning and limiting tree depth can help control it.

The algorithm recursively selects the feature that best splits the data based on a splitting criterion such as information gain or Gini index, which measure the homogeneity of resulting partitions (Quinlan, 1996). Splitting continues until a stopping condition is met, such as a minimum number of samples per leaf.

4.3 K-Nearest Neighbors (KNN)

K-nearest neighbors (KNN) is a type of instance-based learning algorithm, which is a type of lazy learning (Tan et al., 2020). KNN works by comparing a new instance with the training dataset and finding the k number of closest instances based on a distance metric. The majority class of the k nearest neighbors is used to determine the new instance’s Categorization (Peterson, 2009).

It is non-parametric, meaning it does not make strong assumptions about the data distribution. This makes KNN flexible but also computationally intensive for large datasets.

The KNN algorithm can be represented mathematically as follows: Given a set of training instances S = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n), where x_i is the feature vector of instance i and y_i is its corresponding class label, and a new instance x, the KNN algorithm works by calculating the distance between x and each training instance x_i using a distance metric such as Euclidean distance:

$$d(x, x_i) = \sqrt{(\sum (x - x_i)^2)} \tag{2}$$

All the distances are sorted in ascending order to find the k closest neighbors to x. Then the majority class among the k neighbors is determined and the class is assigned to the new instance x. The value of k is a hyper-parameter that can be tuned to achieve better classification performance. A small value of k will result in a more flexible decision boundary, but may be more susceptible to noise, while a large value of k will result in a smoother decision boundary but may lead to oversimplification of the problem. KNN is a simple and effective algorithm that can be used for both classification and regression tasks. However, it can be computationally expensive for large datasets and high-dimensional feature spaces, as the distance calculation and sorting can be time-consuming (Laaksonen & Oja, 1996).

Preprocessing steps such as feature normalization or scaling are often required to ensure fair distance calculations, especially when features are on different scales.

KNN can also be sensitive to the choice of distance metric and the value of k, and may require additional preprocessing steps such as feature scaling to improve its performance.

4.4 XGBoost (XGB)

XGBoost, an extreme gradient boosting method, was initially introduced by Chen & Guestrin (2016) in 2016 as a highly efficient, adaptable, and portable algorithm. It is specifically designed for speed and performance, making it particularly suitable for structured/tabular data. It belongs to the family of decision tree-based ensemble learning methods and is capable of handling regression and classification problems, as depicted in **Figure 2**.

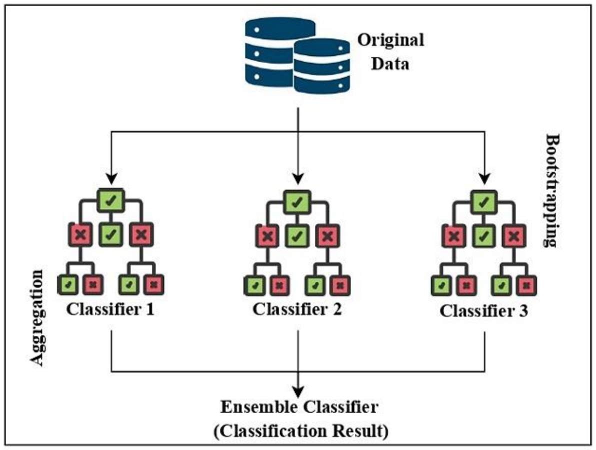


Figure 2. Working process of XGBoost.

XGBoost has been successfully applied in various fields, demonstrates excellent performance, particularly with relatively small datasets. Boosting is a method of machine learning that gradually merges several weak learners into one robust learner with improved computing capability (Zaki et al., 2022). By employing gradient descent techniques, weak learners are iteratively added to improve the overall prediction accuracy by reducing the loss. XGBoost (Isha et al., 2024) builds upon the gradient boosting framework and offers several advanced features such as handling missing data, parallelizing weak learners, depth-first tree pruning, and addressing overfitting problems through regularisation. Among XGBoost's many distinguishing features is its comprehensive collection of functions, which includes out-of-core computing, cache awareness, the ability to deal with sparse data, and a weighted quantile sketch. XGBoost's unique characteristics and impressive performance set it apart from more conventional machine learning techniques (Mittal et al., 2024).

XGBoost is based on the following core concept: Consider a dataset D_S= $\{(x_i,\ y_i)\}$, where x_i and y_i represents the set of input features and the corresponding classification outcome respectively. The objective function is computed using the estimated residuals from each iteration, and an ensemble of decision trees is built during each iteration. To begin the next cycle, a new model is constructed with a strong emphasis on residual fitting, which is in turn largely reliant on the error function matrix first and second derivatives. Consequently, for p iterations, p decision trees will be generated. The next iteration's optimal split point is determined using a greedy procedure. Ultimately, multiple trees are created, each containing numerous leaf nodes with associated scores. Multiplying each leaf node's score by its weight yields the final anticipated value. The overall prediction, determined using Equation (3), is derived after iterating the model p times in the XGBoost ensemble model (Gupta et al., 2022):

$$\hat{y}_i = \sum_{p=1}^{p} f_p(x_i), f_p \in F \tag{3}$$

where, $\hat{y}_i, f_p\ (x_i)$ are the final outcome and predicted outcome respectively at p^{th} iteration by implementing the residual tree, F represents a function that indicates the space of a residual tree. The objective function, which is minimized across iterations, is broken down into two parts -loss function and regularization and assessed with Equation (4) (Gupta et al., 2022).

$$f_p = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega\big(f_p\big) \tag{4}$$

where, l is the loss function and it determines the distance between y_i and $\hat{y}_i$ and Ω represents the regularisation that is used to prevent model overfitting.

5. Proposed Methodology

The methodology presented and utilised for predicting wine quality is detailed here. Preparing the data, choosing the features to use, model training, and classification are the four basic components. In the following sections, each step is described in detail. **Figure 3** shows the proposed architecture for predicting wine quality.

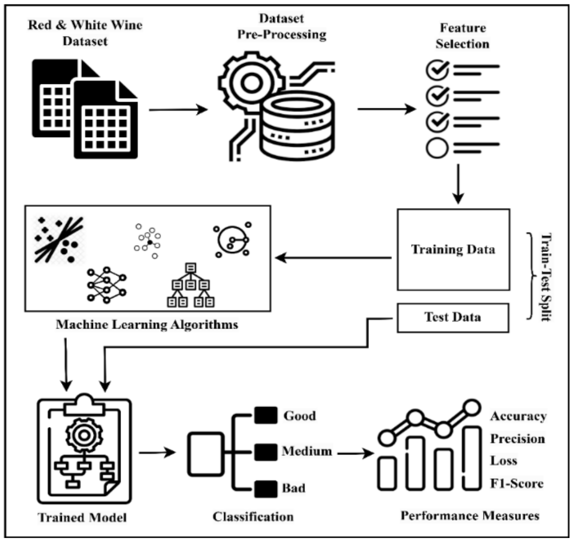


Figure 3. Workflow of the proposed model for prediction of wine quality.

5.1 Data Pre-Processing

To ensure that a machine learning model can produce accurate predictions, data preprocessing is an integral part of the prediction process. Data pre-treatment improves model performance, reduces overfitting, and makes the model more robust to changes in the input data. Data preprocessing has the ability to reduce computational costs and boost the performance of the machine learning model. The dataset used in this study is the wine quality Dataset 1. While pre-processing of the dataset, two different datasets for Red and White wine are integrated to get a good mix of different wine qualities. After integration we get 1599 instances for red wine and 4898 instances for White wine as shown in **Figure 4**. A summary of key statistical properties for each feature of the dataset is presented in **Table 1**.

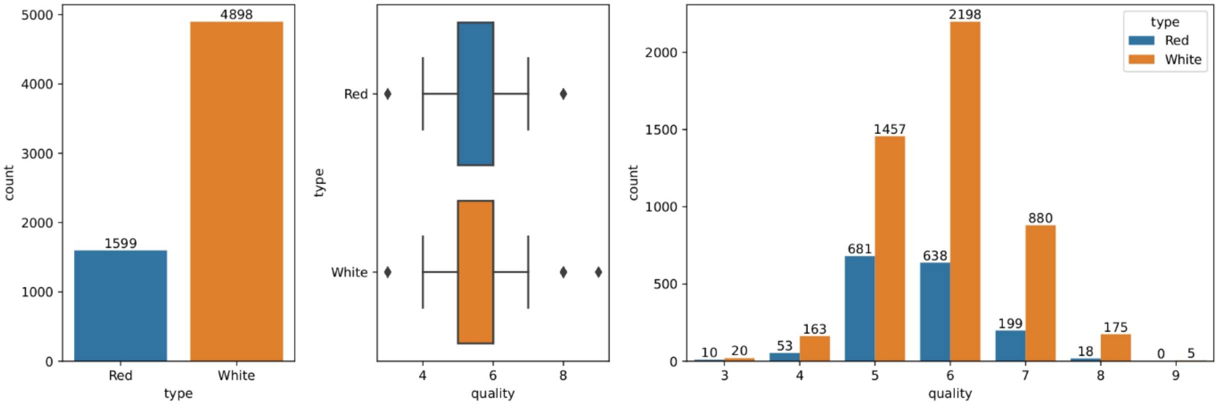


Figure 4. Dataset distribution.

Table 1. Descriptive statistics of selected dataset features.

Feature	Mean	Std dev	Min	Max
Residual sugar	2.44	1.03	0.90	15.50
Free sulphur dioxide	16.52	10.08	1.00	68.00
Total sulphur dioxide	48.27	33.04	6.00	289.00
pH	3.31	0.15	2.74	4.01
Quality	2.69	0.84	0.00	5.00
MSO_2	0.52	0.36	0.02	3.36
Acid/density	9.16	1.80	5.31	16.99
Alcohol density	10.39	1.04	8.39	14.86
Sulphate/density	0.65	0.15	0.33	2.00
Sulphates/acid	1.41	0.72	0.25	6.96
Sulphates/chlorides	8.33	3.03	1.37	60.83
Sulphates × alcohol	0.06	0.01	0.03	0.21

Then the dataset has been pre-processed using the following steps. To begin, the blank spots in the data have been filled up during the preparation phase. Because they add nothing to the dataset, features with an excessive number of missing values have been eliminated. Secondly, the quality column which is rated on the scale of 1 to 10 is converted into three qualities viz. Low, Medium and high as shown in **Figure 5**. Further wine quality is normalized into three numerical values such as Low: ‘0’, Medium: ‘1’ and high: ‘2’. The type of wine is converted into numerical values such as Red: ‘0’ and White: ‘1’. To eliminate attribute bias with sensitivity and lessen the impact of variance in measurement units for various features, the next stage is data normalization, wherein the Minmax Scalar method is used to the dataset’s values. Using Equation 5, the Minmax scalar rescales feature values from 0 to 1.

$$x_{scaled} = (x - x_{min}) \div (x_{max} - x_{min}) \tag{5}$$

where, x_{min} and x_{max} are the minimum and maximum values of that property A_i, respectively, x_{scaled} is the rescaled value obtained from the original value x.

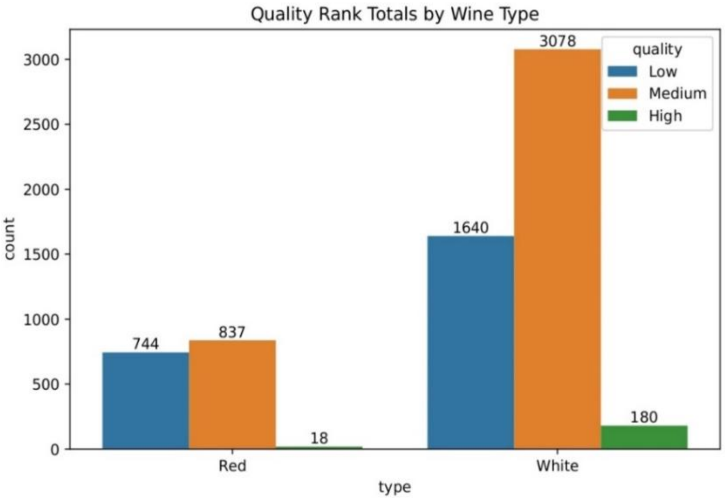


Figure 5. Distribution of red and white wine quality.

5.2 Feature Selection

Feature selection is a powerful strategy for maximising the predictive efficacy of machine learning algorithms by zeroing down on the most relevant characteristics and excluding the noise (Krishnaveni et al., 2022; Zhao et al., 2023). Feature selection is the process of identifying the most useful attributes to describe your data. There are various feature selection techniques available in the literature. After being cleaned up, the dataset used in this study has 13 attributes, which could potentially reduce the classifier's accuracy. Therefore, in order to find the best set of features from the wine data set, we have rigorously prioritised and selected the features using the feature selection strategies discussed in Section 3.

Table 2. Parameters for wrapper based GA.

Parameter	Value
Population_size	100
Max_generations	25
Selection_probability	0.7
Mutation_probability	0.2
Crossover_probability	0.6
Max_iterations	100

In this study we have used wrapper based genetic algorithm which selects 7 important and non-redundant features from the dataset. The parameters used by the genetic algorithms are presented in **Table 2**.

5.3 Model Training

Here, the proposed model is used in conjunction with other machine learning methods to fit the data from the chosen set of features. The dataset has been split into train and test datasets in the data pre-processing process before applying them to classification models of machine learning. All the machine learning models are trained using three train-test split ratio (70%-30%, 80%-20% and 90%-10%) from which thirty percent of the instances were randomly assigned to the test dataset, while the remaining seventy percent were assigned to the train dataset and so on. The training parameters of various machine learning algorithms are presented in **Table 3**.

Table 3. Machine learning algorithms and associated hyperparameters.

S. No.	Algorithm	Hyperparameters
1.	Support vector classifier	decision_function_shape = ovo, C = 1, degree = 5, gamma = 1, kernel = poly
2.	Decision tree classifier	criterion = log_loss, max_depth = 20, max_features = sqrt, min_samples_leaf = 1, min_samples_split = 2, splitter = best
3.	K-Nearest neighbor	learning_rate = 0.01, batch_size = 64, epochs = 500, hidden_layers = 6
4.	XGBoost	learning_rate = 0.01, max_depth = 10, min_child_weight = 1, n_estimators = 500, objective = squarederror

5.4 Classification

Machine learning models can be trained to classify new instances into one of several pre-defined categories or classes using the supervised learning process known as classification. After being exposed to the training dataset, the machine learning algorithms in this study are able to predict and categorise wine quality into the low, medium, and high.

6. Experimental Results and Analysis

This section describes the environment for implementing the proposed model, various metrices used for evaluating the model and finally the results are discussed.

6.1 Simulation Settings

In this article all the experiments are performed on the system with i7 2.1 GHz eight core processor having RAM of 16 GB. The proposed model is trained by using PyTorch library available in Python 3.10 version.

6.2 Evaluation Measures

Evaluation measures assess the performance of machine learning models in predicting wine quality using confusion matrix that is often used to describe the performance of a classification model or algorithm. The following evaluation measures are commonly used to evaluate the effectiveness of such models:

Precision: Precision refers to how well a model predicts the wine quality in a given category. It quantifies the proportion of correctly predicted instances in a specific category out of all instances predicted as belonging to that category (Shamrat et al., 2023; Valero-Carreras et al., 2023). A higher precision indicates fewer false positives. It is defined by the equation:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6}$$

Recall: Recall evaluates the model's ability to identify instances of a particular wine quality category correctly. It is a statistical measure of how many times a certain category has been predicted accurately relative to how many times that category has been observed (Shamrat et al., 2023). Higher recall suggests fewer false negatives; it is defined by the formula:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{7}$$

F*1- *Score: The *F*1 score is an appropriate measure because it takes into account both accuracy and recall. It's a measure of a model's efficacy as a whole, calculated by harmonic mean of precision and recall (Valero-Carreras et al., 2023). It is mathematically expressed as:

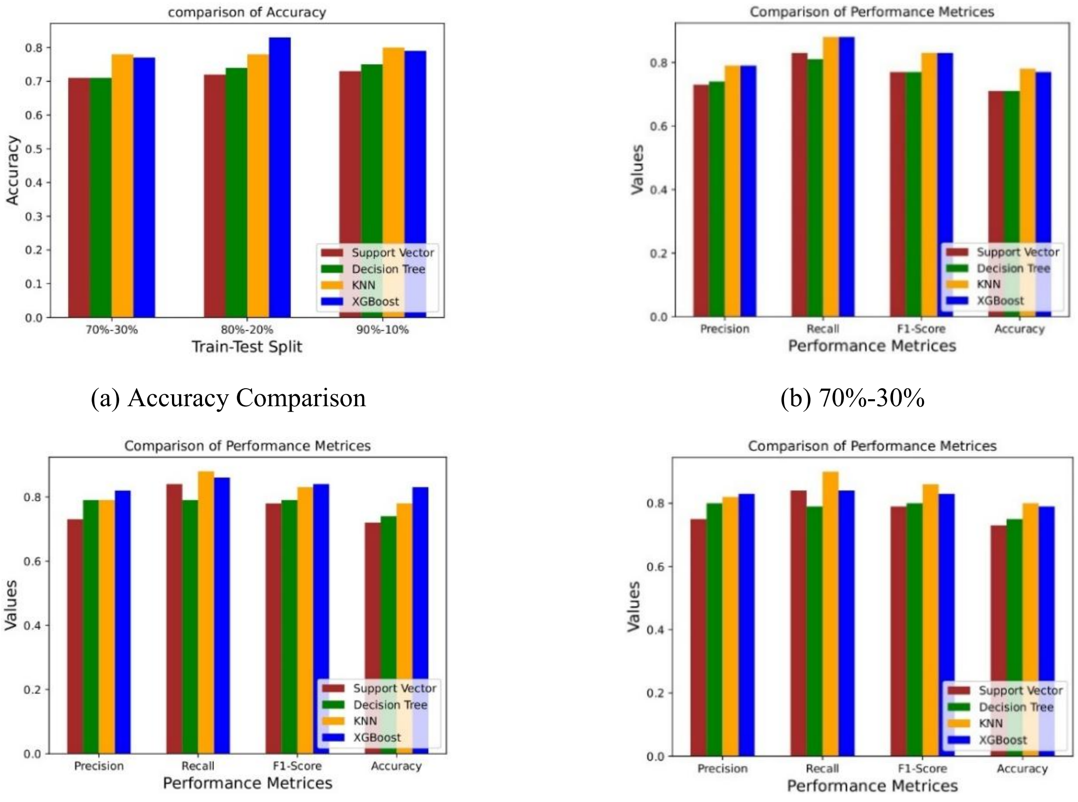
$$F1-\text{ score } = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

***Accuracy*:** In order to gauge how well a model predicts outcomes, accuracy is often utilised. True positive and negative rates are computed as a percentage of the total number of instances predicted (Shamrat et al., 2023; Valero-Carreras et al., 2023). Accuracy is a metric used to evaluate models in a broad sense.

$$\text{Accuracy } = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

6.3 Results and Discussion

This study is divided into two main parts, comprising the complete experimental process. In the first part, the focus is on feature selection from the original dataset. The objective is to identify the most relevant features that contribute significantly to the prediction task. The second section involves conducting a new experiment to find the optimal classifier for optimising the prediction performance. In the first part, wrapper based genetic algorithm were used to extract the most relevant features for wine quality prediction. The purpose of this step is to decrease the dimensionality of the data, which ultimately leads to enhanced performance and increased accuracy. The original dataset comprises of 13 features, out of which 7 features are selected by the feature selection algorithm used in the experiment.



(a) Accuracy Comparison

(b) 70%-30%

(c) 80%-20%

(d) 90%-10%

Figure 6. Performance comparison of classifiers using different data splits and evaluation metrics. (a) shows the overall accuracy comparison among classifiers. (b), (c), and (d) present detailed performance metrics for 70%-30%, 80%-20%, and 90%-10% train-test splits respectively. These plots highlight the effectiveness of the proposed WGA-XGB model across varying experimental settings.

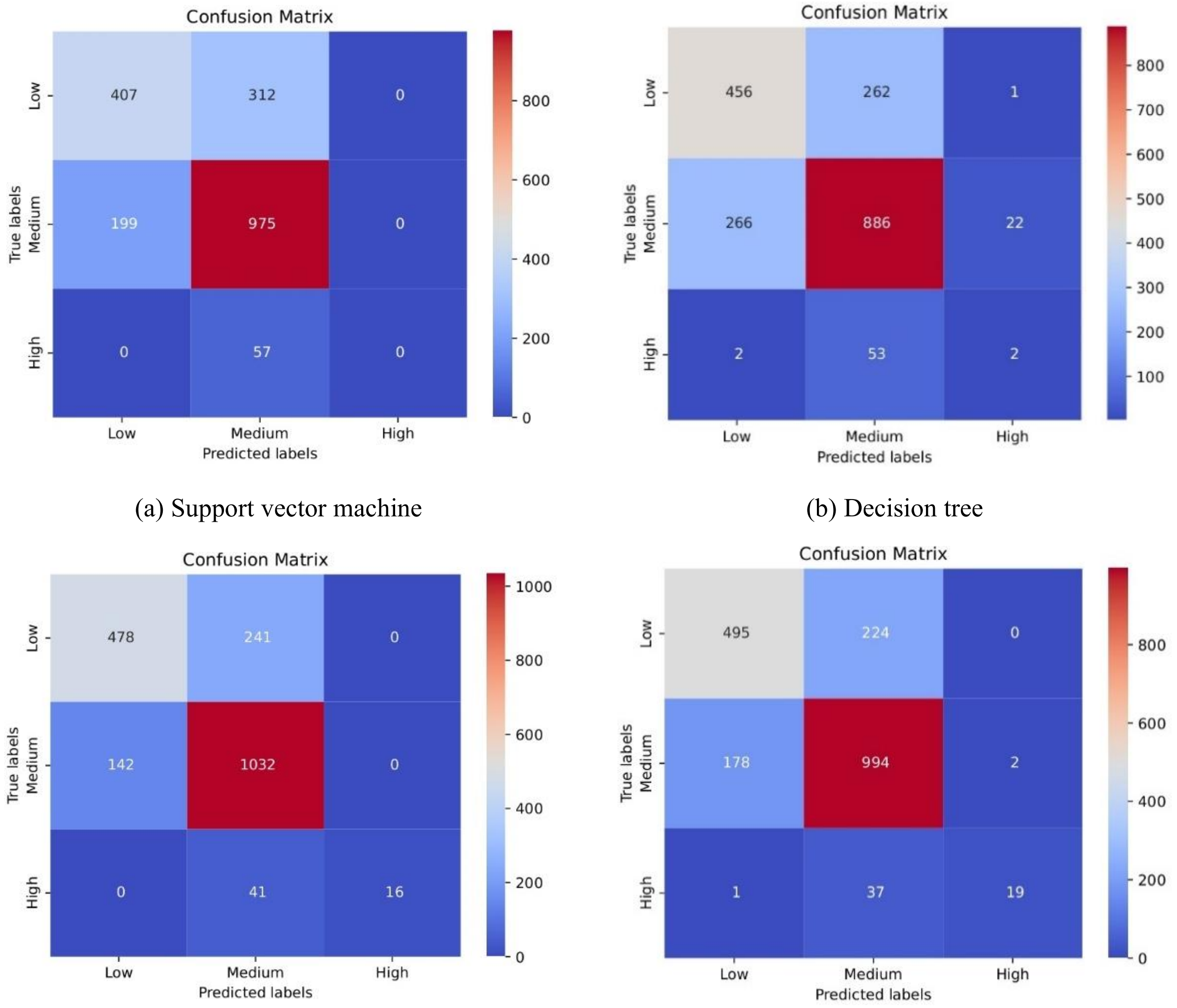
The second part of the experiment aims to identify the optimal classifier that maximizes the prediction performance. For evaluating the effectiveness of the XGBoost based prediction model, a comparison is made with three other state- of-the-art machine learning algorithms, namely support vector classifier (SVC), Decision tree Classifier (DT), and K- Nearest Neighbour (KNN). The best accuracy of 83% is achieved through XGBoost classifier. Different accuracy values are obtained by using different Train-Test splits, as shown in **Table 4**. When comparing other train-test splits (90:10, 70:30, and 80:20), it is clear that the 80:20 split yields the highest accuracy. **Figure 6(a)** presents the accuracy of several classifiers. Also, we have computed Precision, Recall and F1-score for different classifiers for different train-test splits (90:10, 70:30, and 80:20) as presented in **Figures 6(a), 6(b)** and **6(c).** It is clearly observed that XGBoost classifier model with a train-test split of 80:20 ratio has achieved a precision of 82%, Recall value of 86% and *F*1-Score value of 84%. On the basis of these results, it is evident that XGBoost classifier outperforms in comparison to other classifiers.

Table 4. Performance comparison of different classifiers.

Train-test split	Classifier	Wine quality	Precision	Recall	F1-score	Accuracy
70%-30%	Support vector classifier	Low	0.67	0.57	0.61	0.71
		Medium	0.73	0.83	0.77	
		High	0.00	0.00	0.00	
	Decision tree classifier	Low	0.66	0.59	0.62	0.71
		Medium	0.74	0.81	0.77	
		High	0.19	0.07	0.10	
	KNN classifier	Low	0.77	0.66	0.71	0.78
		Medium	0.79	0.88	0.83	
		High	1.00	0.28	0.44	
	XGBoost	Low	0.73	0.69	0.71	0.77
		Medium	0.79	0.85	0.82	
		High	0.90	0.33	0.49	
80%-20%	Support vector classifier	Low	0.68	0.58	0.63	0.72
		Medium	0.73	0.84	0.78	
		High	0.00	0.00	0.00	
	Decision tree classifier	Low	0.67	0.69	0.68	0.74
		Medium	0.79	0.79	0.79	
		High	0.43	0.38	0.41	
	KNN classifier	Low	0.77	0.66	0.71	0.78
		Medium	0.79	0.88	0.83	
		High	0.93	0.33	0.49	
	XGBoost	Low	0.76	0.74	0.75	**0.83**
		Medium	0.82	0.86	0.84	
		High	0.86	0.43	0.57	
90%-10%	Support vector classifier	Low	0.68	0.59	0.63	0.73
		Medium	0.75	0.84	0.79	
		High	0.00	0.00	0.00	
	Decision tree classifier	Low	0.69	0.69	0.69	0.75
		Medium	0.80	0.79	0.80	
		High	0.46	0.60	0.52	
	KNN classifier	Low	0.81	0.72	0.76	0.80
		Medium	0.82	0.90	0.86	
		High	1.00	0.35	0.52	
	XGBoost	Low	0.73	0.75	0.74	0.79
		Medium	0.83	0.84	0.83	
		High	0.75	0.45	0.56	

For different Train-Test splits we have presented the results in the form of confusion matrices as shown in **Figures 7, 8** and **9**. **Figure 7** represents the confusion matrix for 70:30 train-test splits for different classifiers, which specifies that how accurately the model predicts the quality of wine in Low, Medium and

High quality. It is clearly evident that support vector classifier had not able to predict high quality wine whereas other classifiers perform well in predicting the high-quality wine. **Figures 8** and **9** depicted that XGBoost performs well in comparison to other state of the art machine learning algorithms. XGBoost outperforms other classifiers, most likely due to its ability to simulate complicated feature interactions and its resistance to overfitting via regularization. Simpler models, such as KNN and Decision Tree, may struggle with the dataset's high dimensionality and nonlinearity.



(a) Support vector machine

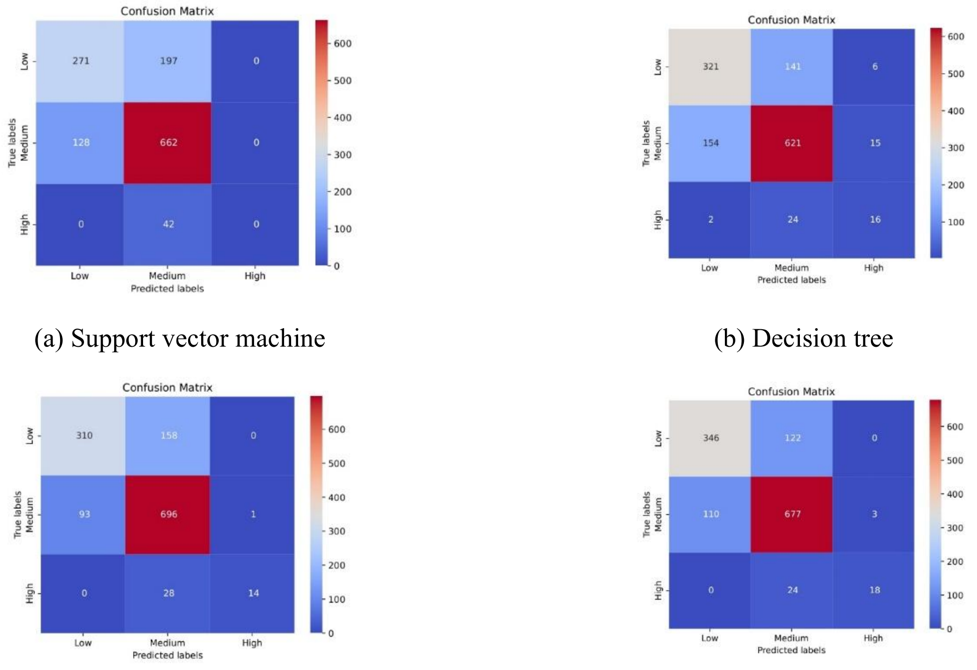
(b) Decision tree

(c) K-Nearest neighbour

(d) XGBoost

Figure 7. Confusion matrix for 70:30 split of different ML models.

The proposed approach has a limitation in that only one dataset is used for the evaluation. In future research, we plan to assess the performance and robustness of the presented model by deploying it in a cloud environment and testing it with multiple wine datasets. We will also focus on addressing numerous security concerns with the existing framework.



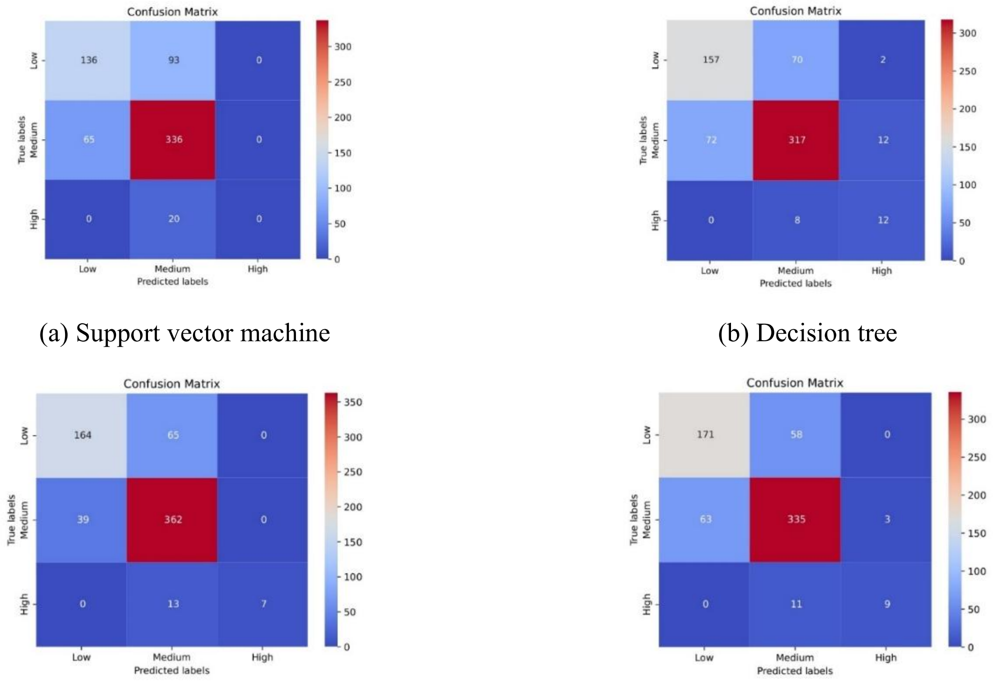
(a) Support vector machine

(b) Decision tree

(c) K-Nearest neighbour

(d) XGBoost

Figure 8. Confusion Matrix for 80:20 split of different ML models.



(a) Support vector machine

(b) Decision tree

(c) K-Nearest neighbour

(d) XGBoost

Figure 9. Confusion Matrix for 90:10 split of different ML models.

6.4 Comparison with Existing Studies

Table 5 compares existing studies on wine quality prediction. It describes the methods employed, feature selection algorithms, and the classification accuracy. The comparison shows that most research either lack strong feature selection or depend on narrow classifiers. In contrast, the proposed approach uses a wrapper-based genetic algorithm for feature selection and achieves comparable accuracy with XGBoost, demonstrating its efficacy and uniqueness.

Table 5. Comparison of existing studies on wine quality prediction.

Reference(s)	Techniques used	Feature selection	Best performing model	Accuracy
Mahima et al. (2020)	Random forest, kNN	Not specified	Random Forest + kNN	82%
Dahal et al. (2021)	ANN, GBR, RR, SVM	Not specified	Gradient boosting regressor	82%
Chiu et al.(2021)	Genetic algorithm + ML	Genetic algorithm	Hybrid model (optimized by GA)	83.5%
Mohana et al. (2023)	Ensemble (SVM, RF, XGB)	Not specified	Ensemble model	85%
Jana et al.(2023)	Neural network, SVM	Not specified	Quadratic SVM	83%
Gupta (2018)	SVM, neural network	Linear regression	SVM	82%
Kumar et al. (2020)	SVM, Naive Bayes, RF	Not specified	SVM	81%
Proposed method	GA + XGBoost, RF, KNN, etc.	Wrapper-based genetic algorithm	XGBoost	**83%**

7. Conclusion and Future Scope

As the demand for premium wines continues to rise, it becomes imperative to devise methodologies that can effectively forecast wine quality. While machine learning has exhibited considerable promise in this domain, the abundance of features that can impact wine quality necessitates the implementation of feature selection techniques. This step assumes paramount importance in enhancing the precision and effectiveness of predictive models. The proposed framework incorporates a feature selection algorithm based on the wrapper based genetic algorithm, which effectively identifies the most influential features for wine quality prediction. By applying the framework to a dataset of red and white wine samples, the results demonstrate that the feature selection algorithm significantly reduces the number of features required while maintaining high prediction accuracy. Different machine learning classifiers are trained using the selected features which demonstrates that XGBoost outperforms in terms of accuracy and other statistical measures than other machine learning algorithms. According to the outcome of thorough experimentation, the proposed model has a higher accuracy of 83% in comparison to other machine learning algorithms. Precision (82%), Recall (86%) and F1-Score (84%) are other statistical measures which proves the efficacy of the proposed model in wine quality prediction. Future research directions may involve applying the proposed framework to additional wine datasets to validate its robustness and scalability. Additionally, exploring more advanced feature selection algorithms and machine learning models could further improve predictive performance. The study is constrained by a slight class imbalance, which may influence generalisability to minority groups. Future validation of varied wine data is required to confirm resilience.

Conflict of Interest

I declare that the authors have no competing interests or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Acknowledgments

The authors have no acknowledgments to declare.

AI Disclosure

The author(s) declare that no assistance is taken from generative AI to write this article.

References

Al-Shammary, D., Albukhnefis, A.L., Alsaeedi, A.H., & Al-Asfoor, M. (2022). Extended particle swarm optimization for feature selection of high-dimensional biomedical data. *Concurrency and Computation: Practice and Experience*, *34*(10), e6776. https://doi.org/10.1002/cpe.6776.

Al-Yaseen, W.L., Idrees, A.K., & Almasoudy, F.H. (2022). Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system. *Pattern Recognition*, *132*, 108912. https://doi.org/10.1016/j.patcog.2022.108912.

Anguita, D., Boni, A., Ridella, S., Rivieccio, F., & Sterpi, D. (2005). Theoretical and practical model selection methods for support vector classifiers. In: Wang, L. (ed) *Support Vector Machines: Theory and Applications.* Springer, Berlin, Heidelberg, pp. 159-179. ISBN: 978-3-540-32384-6(e), 978-3-540-24388-5(p). https://doi.org/10.1007/10984697_7.

Avros, R., Dudka, V., Křena, B., Letko, Z., Pluháčková, H., Ur, S., Vojnar, T., & Volkovich, Z. (2017). Boosted decision trees for behaviour mining of concurrent programmes. *Concurrency and Computation: Practice and Experience*, *29*(21), e4268. https://doi.org/10.1002/cpe.4268.

Bhardwaj, P., Tiwari, P., Olejar, K., Parr, W., & Kulasiri, D. (2022). A machine learning application in wine quality prediction. *Machine Learning with Applications*, *8*, 100261. https://doi.org/10.1016/j.mlwa.2022.100261.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: a new perspective. *Neurocomputing*, *300*, 70-79. https://doi.org/10.1016/j.neucom.2017.11.077.

Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM. New York, USA. https://doi.org/10.1145/2939672.2939785.

Chhikara, S., Bansal, P., & Malik, K. (2023). Wine quality prediction using machine learning techniques. In *International Conference on Smart Trends in Computing and Communications* (pp. 137-148). Springer Nature, Singapore. https://doi.org/10.1007/978-981-99-0769-4_14.

Chiu, T.H.Y., Wu, C., & Chen, C.H. (2021). A generalized wine quality prediction framework by evolutionary algorithms. *International Journal of Interactive Multimedia and Artificial Intelligence*, *6*(7), 60-70.

da Costa, N.L., Valentin, L.A., Castro, I.A., & Barbosa, R.M. (2021). Predictive modeling for wine authenticity using a machine learning approach. *Artificial Intelligence in Agriculture*, *5*, 157-162. https://doi.org/10.1016/j.aiia.2021.07.001.

Dahal, K.R., Dahal, J.N., Banjade, H., & Gaire, S. (2021). Prediction of wine quality using machine learning algorithms. *Open Journal of Statistics*, *11*(2), 278-289. https://doi.org/10.4236/ojs.2021.112015.

Gupta, A., Rajput, I.S., Gunjan, Jain, V., & Chaurasia, S. (2022). NSGA-II-XGB: meta-heuristic feature selection with XGBoost framework for diabetes prediction. *Concurrency and Computation: Practice and Experience*, *34*(21), e7123. https://doi.org/10.1002/cpe.7123.

Gupta, Y. (2018). Selection of important features and predicting wine quality using machine learning techniques. *Procedia Computer Science*, *125*, 305-312. https://doi.org/10.1016/j.procs.2017.12.041.

Huang, J., Cai, Y., & Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters*, *28*(13), 1825-1844. https://doi.org/10.1016/j.patrec.2007.05.011.

Isha, Thapliyal, N., Solanki, S., Pandey, N.K., & Papola, S. (2024). Employee attrition analysis using XGBoost. *2024 International Conference on Communication, Computer Sciences and Engineering* (pp. 1-6). IEEE. Gautam Buddha Nagar, India. https://doi.org/10.1109/ic3se62002.2024.10593326.

Jana, D.K., Bhunia, P., Adhikary, S.D., & Mishra, A. (2023). Analyzing of salient features and classification of wine type based on quality through various neural network and support vector machine classifiers. *Results in Control and Optimization*, *11*, 100219. https://doi.org/10.1016/j.rico.2023.100219.

John, A., Damodar, N., Mohith, N., & Vijayan, B. (2024). Understanding red wine quality with feature importance in machine learning and explainable AI. In *2024 15th International Conference on Computing Communication and Networking Technologies* (pp. 1-6). IEEE. Kamand, India. https://doi.org/10.1109/icccnt61001.2024.10724210.

Krishnaveni, S., Sivamohan, S., Sridhar, S., & Prabhakaran, S. (2022). Network intrusion detection based on ensemble classification and feature selection method for cloud computing. *Concurrency and Computation: Practice and Experience*, *34*(11), e6838. https://doi.org/10.1002/cpe.6838.

Kumar, S., Agrawal, K., & Mandan, N. (2020). Red wine quality prediction using machine learning techniques. In *2020 International Conference on Computer Communication and Informatics* (pp. 1-6). IEEE. Coimbatore, India. https://doi.org/10.1109/iccci48352.2020.9104095.

Laaksonen, J., & Oja, E. (1996, June). Classification with learning k-nearest neighbors. In *Proceedings of International Conference on Neural Networks* (Vol. 3, pp. 1480-1483). IEEE. Washington, DC, USA. https://doi.org/10.1109/icnn.1996.549118.

Mahima, Gupta, U., Patidar, Y., Agarwal, A., & Singh, K.P. (2020). Wine quality analysis using machine learning algorithms. In: Sharma, D.K., Balas, V.E., Son, L.H., Sharma, R., Cengiz, K. (eds) *Micro-Electronics and Telecommunication Engineering.* Springer, Singapore, pp. 11-18. ISBN: 978-981-15-2329-8. https://doi.org/10.1007/978-981-15-2329-8_2.

Mittal, K., Gill, K.S., Rajput, K., & Singh, V. (2024). Utilizing machine learning and employing the XGBoost classification technique for evaluating the likelihood of autism spectrum disorder (ASD). In *2024 5th International Conference for Emerging Technology* (pp. 1-5). IEEE. Belgaum, India. https://doi.org/10.1109/incet61516.2024.10593455.

Mohana, R., Sharma, P., & Sharma, A. (2023). Ensemble framework for red wine quality prediction. *Food Analytical Methods*, *16*(1), 30-44. https://doi.org/10.1007/s12161-022-02367-3.

Peterson, L.E. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883. https://doi.org/10.4249/scholarpedia.1883.

Quinlan, J.R. (1996). Learning decision tree classifiers. *ACM Computing Surveys*, *28*(1), 71-72. https://doi.org/10.1145/234313.234346.

Shamrat, F.J.M., Azam, S., Karim, A., Ahmed, K., Bui, F.M., & De Boer, F. (2023). High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images. *Computers in Biology and Medicine*, *155*, 106646. https://doi.org/10.1016/j.compbiomed.2023.106646.

Swain, P.H., & Hauska, H. (1977). The decision tree classifier: design and potential. *IEEE Transactions on Geoscience Electronics*, *15*(3), 142-147. https://doi.org/10.1109/tge.1977.6498972.

Tan, Y., Wu, W., Liu, J., Wang, H., & Xian, M. (2020). Lightweight edge-based kNN privacy-preserving classification scheme in cloud computing circumstance. *Concurrency and Computation: Practice and Experience*, *32*(19), e5804. https://doi.org/10.1002/cpe.5804.

Tax, D.M.J., & Duin, R.P.W. (2004). Support vector data description. *Machine Learning*, *54*(1), 45-66. https://doi.org/10.1023/b:mach.0000008084.60811.49.

Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two SVM models through different metrics based on the confusion matrix. *Computers & Operations Research*, *152*, 106131. https://doi.org/10.1016/j.cor.2022.106131.

Xue, B., Zhang, M., Browne, W.N., & Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606-626. https://doi.org/10.1109/tevc.2015.2504420.

Zaki, J., Nayyar, A., Dalal, S., & Ali, Z.H. (2022). House price prediction using hedonic pricing model and machine learning techniques. *Concurrency and Computation: Practice and Experience*, *34*(27), e7342. https://doi.org/10.1002/cpe.7342.

Zhao, T., Zheng, Y., & Wu, Z. (2023). Feature selection-based machine learning modeling for distributed model predictive control of nonlinear processes. *Computers & Chemical Engineering*, *169*, 108074. https://doi.org/10.1016/j.compchemeng.2022.108074.

Zhou, J., & Hua, Z. (2022). A correlation guided genetic algorithm and its application to feature selection. *Applied Soft Computing*, *123*, 108964. https://doi.org/10.1016/j.asoc.2022.108964.

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.