A Voting Classifier with Wine Quality Dataset

Hiep T. Bui[1,2,*], Thanh H. Le[1,2]

[1]Faculty of Mathematics and Computer Science, University of Science, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam

Abstract
The older the wine, the better it tastes. However, this is not always the case because some wines are made to be consumed immediately. With the goal of developing a simple, fast, but accurate classification method to predict wine quality solely based on its chemical properties, the authors attempted to use some simple machine-learning models in this report. The Wine Quality Dataset from UCI was used in this case study. The model correctly classifies the quality with **86.5%** accuracy for white wine and **80.7%** accuracy for red wine. Various performance methods are also being measured to calculate and the results are available to compare.

Keywords
Machine learning, XGBoost, Random Forest, LightGBM, Wine dataset, Soft Voting, Hard Voting, Classification

1. Introduction

The wine industry has a long history, with a market value estimated to be around 435 billion US dollars in 2021. Despite its high value, this business is heavily reliant on wine quality, but this quality is defined by experts and consumers. Whether or not they are an expert, the individual defining wine quality determines its quality in different ways. Due to the understanding of wine production and the chemical elements included within it, experts have a distinctive perspective on wine quality while normal people may not know about it. When a business owner is experimenting with a new product, testing wine by an expert could be quite expensive while the non-expert may have preconceived notions about price, presentation, and provenance.

Machine learning is used nowadays in many different fields. Current computer capabilities make machine learning more effective, which could be a game changer for many different ways to solve problems, saving time and money. Because of this, even though machine learning was developed in the first half of the 20th century, it did not really take off until the recent decades, and right now its applications are well-known and still on the way to development.

Being introduced first by Cortez et al. [1] to data mining with wine from physicochemical properties. The main objective of this article is to develop a brief, simple but accurate algorithms that can identify the quality using only its chemical properties. Yet there are certain challenges that come with this endeavor. The imbalance between labels has been replaced by using SMOTE method to oversample.

Figure 1: The wine quality

The structure of this study is as follows: The data utilized in this investigation are described in Section 2 along with a quick overview of the baseline origin and our approach. Section 4 suggests a stable way to maintain accuracy multiple times by randomly training and testing, and Section 5 summarizes all the findings along with other methods.

2. Related work

The original idea is introduced by using the red wine dataset to predict quality but with all the attributes by Trivedi and Sehrawat [2], where the author is focus on some famous classification methods as below:

- Logistic regression
- Random Forest

This article is a simulated version of a research paper.

*Corresponding author
22c01007@student.hcmus.edu.vn (H. T. Bui);
22c01018@student.hcmus.edu.vn (T. H. Le)

The process starts with preprocessing to remove all the outliers that may affect the outcome and labeling the quality as "Good" (above 5) or "Bad" (below 5) to bring out the higher accuracy model and its F1 score. As the article has shown, the Random Forest, which is given **84%** in accuracy and **85%** with F1-score is better than Logistic regression.
By Kumar et al. [3], along with Random Forest, Naive Bayesian, and Supper Vector Machine is also introduced but they are not labeling the quality as Good or Bad but only predicting between the range 3-8.
By Bhardwaj et al. [4], the group of authors has used SMOTE method to solve the imbalanced problem but with a different dataset for wine. Then, they simply apply various machine learning algorithms with default parameters.

3. Methodology

The authors utilize some machine learning algorithm that is being well-known and has been used in earlier work. Each model comes with a brief explanation and the confusion matrix in which the dataset is randomly divided 0.8 for training and 0.2 for testing. All the algorithm is executed with default parameters. This baseline is using Red Wine dataset and SMOTE has already been implemented.

3.1. Decision Tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes. One technique to show an algorithm that simply uses conditional control statements is to use this method. A whole dataset's decision tree is constructed using all the relevant features (variables). Up until overfitting, accuracy keeps getting better until it worse.

Table 1
Decision Tree Confusion matrix

	False	True
False	106	34
True	42	124

3.2. Random Forest

Random Forest is a classification algorithm developed by Breiman and Cutler uses an ensemble of tree predictors. In essence, random forests are a collection of tree predictors where each tree is reliant on the values of a random vector that is sampled randomly and with the same distribution over all of the trees in the forest. This method may be applied to both classification and regression tasks, making it versatile and simple to use. A random forest is a collection of decision trees that builds several decision trees by randomly choosing observations and particular attributes, then averaging the outcomes. This classification method estimates missing data and large proportion of the data are missing it still maintains accuracy.

Table 2
Random Forest Confusion matrix

	False	True
False	119	21
True	29	137

3.3. Logistic Regression

Logistic regression is analogous to multiple linear regression, except the outcome is binary, which means if the estimated probability is greater than 50%, then the model predicts that the instance belongs to 0 or 1 class. Various transformations are employed to convert the problem to one in which a linear model can be fit. Due to its fast computational speed and its output of a model that lends itself to rapid scoring of new data, this is a popular method.
Logistic regression is based on the assumption that the value of dependent variable is predicted by using independent variables. In the model, Y is the dependent variable we are trying to predict. The value of Y that corresponds to the wine quality is Good (Y=1) or Bad (Y=0) and is summarized by (X=x). Then, the conditional probability follows a logistic distribution given by $P(Y = 1|X = x_i)$.

Table 3
Logistic Regression Confusion matrix

	False	True
False	116	24
True	44	122

3.4. Support Vector Machine

As Géron [5] describe, Support Vector Machine (SVM) was developed by Vapnik in 1995, is based on the principle of structural risk minimization that exhibits good generalization performance. SVM finds an optimal separating hyperplane between classes by focusing on the support vectors is proposed. This hyperplane seperates

the training data by maximal margins.
SVM is an extremely strong and flexible Machine Learning model that can do regression, outlier identification, and linear or nonlinear classification. It is among the most well-liked machine learning models. SVMs are especially effective in classifying complex datasets that are small to medium in size. Hence, the authors are using Gaussian RBF Kernel with the default parameter.

Table 4
Support Vector Machine Confusion matrix

	False	True
False	116	24
True	44	122

3.5. K-Nearest Neighbors

KNN is one of the simpler prediction/classification techniques. A KNN model's fundamental idea is to categorize each record, discover K records with similar features, determine which class predominates among those cases, and then give that class to each subsequent record. Using the distance metrics to measure the closeness between training samples and the test sample; in terms of closeness, the KNN is mostly based on the Euclidean distance. The Euclidean distance between training sample $X_1 = (x_{11}, x_{12}, ..., x_{1n})$ with n features, $X_2 = (x_{21}, x_{22}, ..., x_{2n})$ and $m = 2$ is

$$distance(X_1, X_2) = \left(\sum_{i=1}^{n} (x_{1i} - x_{2i})^m\right)^{1/m} \quad (1)$$

When $m = 1$ the distance is called as Manhattan and m > 2 the distance called as Minkowski. In this case study, choosing $m\_neighbors = 2$.

Table 5
K-Nearest Neighbors Confusion matrix

	False	True
False	126	14
True	67	99

3.6. Gaussian Naive Bayes

Gaussian Naive Bayes is a simple probabilistic classification algorithm based on Bayes' theorem. It is called "naive" because it assumes that the features (or variables) used in the classification are independent of each other. Naive Bayes assumes that all feature values follow a normal distribution and uses the mean and variance of each feature to calculate the probability. Naive Bayes performs well on high-dimensional and complex datasets. Using Bayes' theorem, it combines these probabilities to calculate the final probability of the data point belonging to each class. The Bayes's theorem probability:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

Table 6
Gaussian Naive Bayes Confusion matrix

	False	True
False	111	29
True	55	111

3.7. AdaBoost

AdaBoost (or Adaptive Boosting) is one of the most used and effective ensemble learning methods. The base notion behind AdaBoost is that a strong classifier can be created by linearly combining a number of weak classifiers. It is a group of simple models (such as a Decision Tree) that work together to create a complex model. Each model will focus on the data points that were not predicted correctly by the previous model. By doing that, AdaBoost improves the overall accuracy of the model.
In the training process AdaBoost increases the weights of misclassified data points while is decreasing weights of correctly classified data points. That is, AdaBoost reweights all training data in its every iteration. Weak classifiers are applied in serially then generated classification models are combined according to weighted majority voting.

Table 7
AdaBoost Confusion matrix

	False	True
False	115	25
True	46	120

3.8. Gradient Boosting

Similar to the AdaBoost, another boosting algorithm is Gradient Boosting. Gradient Boosting was developed by Friedman (2001) is a powerful machine learning algorithm that has shown considerable success in a wide range of real world applications. GB handles boosting as a method for function estimation, in terms of numerical optimization in function space.

GB works by sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of tweaking the instance weights at every iteration as AdaBoost does, this method tries to fit the new predictor to the residual errors made by the previous predictor.

Table 8
Gradient Boosting Confusion matrix

	False	True
False	124	16
True	41	125

3.9. XGBoost

XGBoost is a stochastic gradient boosting implementation that was first created by Tianqi Chen and Carlos Guestrin at the University of Washington. A computationally efficient implementation with many options, it is available as a package for most major data science software languages. In Python, XGBoost is available as the package xgboost.

Table 9
XGBoost Confusion matrix

	False	True
False	119	21
True	34	132

3.10. LightGBM

LightGBM (or Light Gradient Boosting Machine) is recently a new popular open-source gradient boosting framework developed by Microsoft. LightGBM uses decision tree-based learning algorithm to iteratively boost the performance of the model. Similar to XGBoost, LightGBM is a gradient boosting framework designed for the same purpose of solving complex machine learning problems, but LightGBM is considered that work faster, uses less memory, and handles categorical features more efficiently than XGBoost.

Table 10
LightGBM Confusion matrix

	False	True
False	122	18
True	37	129

3.11. Voting Classifier

A voting classifier is a machine learning model that synthesizes the results of various different models to provide a single, combined forecast. There are two alternative ways to combine the predictions of the component models in a voting classifier: hard voting and soft voting. By the performance of in each of the confusion matrix in the previous part. Hence, the author is using the top 3 highest model: Random Forest, XGBoost, and a new one-LightGBM.

- Hard Voting is the final prediction based on a simple majority vote of the predicted classes from the component models.
- Soft Voting is the final prediction based on the average of the predicted probabilities of the component models.

4. Experiments

4.1. Performance Analysis

4.1.1. Measures

To measure the performance of the result for every single model. The authors are using classification reports, which intensively noting on the Accuracy score and the F1 score. With TB is True Positive, TN is True Negative, FP is False Positive and FN is False Negative. The formulae of accuracy and F1 score is described below:

- $\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$
- $\text{F1 score} = 2\frac{Recall*Precision}{Recall+Precision}$, where:
  - $\text{Precision} = \frac{TP}{TP+FP}$
  - $\text{Recall} = \frac{TP}{TP+FN}$

4.1.2. Confusion Matrix

The confusion matrix is a table showing the number of correct and incorrect predictions categorized by type of response. Hence, the best classifier will have a confusion matrix with only diagonal elements and the rest of the elements set to zero. A confusion matrix generates actual values and predicted values after the classification process. The effectiveness of the system is determined according to the following values generated in the matrix. The content of a confusion matrix is display in Figure 2 Please notify that in Section 3, the authors follow Python format in the confusion matrix.

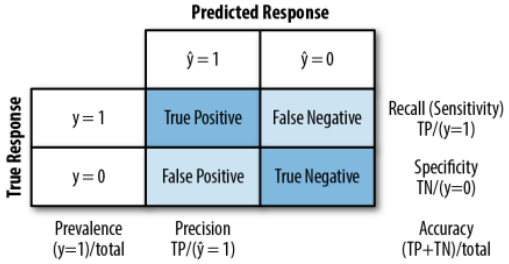

Figure 2: Confusion matrix for a binary response

4.1.3. Normalized features dataset

As some attribute values in the dataset are quite high, this may have an impact on other attributes. To obtain more accurate predictions, the dataset needs to be normalized. To put all attributes on an equal footing, normalizing the data is crucial. Here is how the formula is explained:

$$x_{norm} = (x - \mu) / \sigma \tag{3}$$

4.1.4. Synthetic Minority over Sampling Technique - SMOTE

The Synthetic Minority over Sampling Technique (SMOTE) is a standard framework for learning from imbalanced data. As Bruce et al. [6] described, the SMOTE method locates a record that is comparable to the unsampled record and then generates a synthetic record that is a randomly weighted average of the original record and the nearby record. SMOTE can be seen as an advanced version of oversampling, or as a specific algorithm for data augmentation. The advantage of SMOTE is that we are not generating duplicates, but rather creating synthetic data points that are slightly different from the original data points.
In other words, utilizing SMOTE to produce synthetic data that is compatible with scikit-learn and is modeled after real-world uncommon cases. After eliminating all of the outliers from the dataset, use the SMOTE approach.

4.2. Dataset

The dataset is contributed by Cortez et al. [1], in UCI Machine Learning Repository. This dataset contains 1599 observations for red wines and 4898 observations for white wines. Both dataset contains the total of 12 different feature variables. But due to the assumption from the beginning that we only focus on chemical properties. The dataset of red wine and white wine is divided into training and testing set with the probabilities 0.8 & 0.2 respectively. A description of the features is given below

Feature	Explain	Feature Characteristic
fixed acidity	Most of the acids associated with wine are either fixed or non-volatile	float
volatile acidity	Too much acetic acid in wine can cause an unpleasant vinegar taste	float
citric acid	A small amount of citric acid can increase the freshness and flavor of wine	float
residual sugar	The residual sugar after fermentation has stopped is rarely found in wines below 1 gram per liter, and wines above 45 grams per liter are considered sweeet	float
pH	Describe the acidity or alkalinity of wine, from 0 (acid) to 14 (alkaline); most wines have a pH between 3-4	float
alcohol	Alcohol content of wine	float
quality	Output variables	integer

4.3. Implementation

First, import the library and load the dataset. In Data preprocessing step, we will focus on:

- Labeling the quality with 1: Good ($>= 6$); 0: Bad (< 6);
- Remove outlier;
- Normalized data;
- Drop unnecessary column;
- Applying SMOTE method to oversampling.

Next, divide the dataset as 0.8 and 0.2 for training and testing. Then we apply machine learning algorithms, and compare each other. Based on the confusion matrix for each algorithm, the author realizes Random Forest, XGBoost and LightGBM is giving the best outcome. Apply 3 algorithms by Voting.

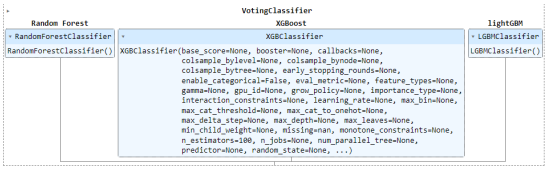

Figure 3: Voting apply with Random Forest, XGBoost, and LightGBM

In a word, each steps in this progress can be displayed as figure 4

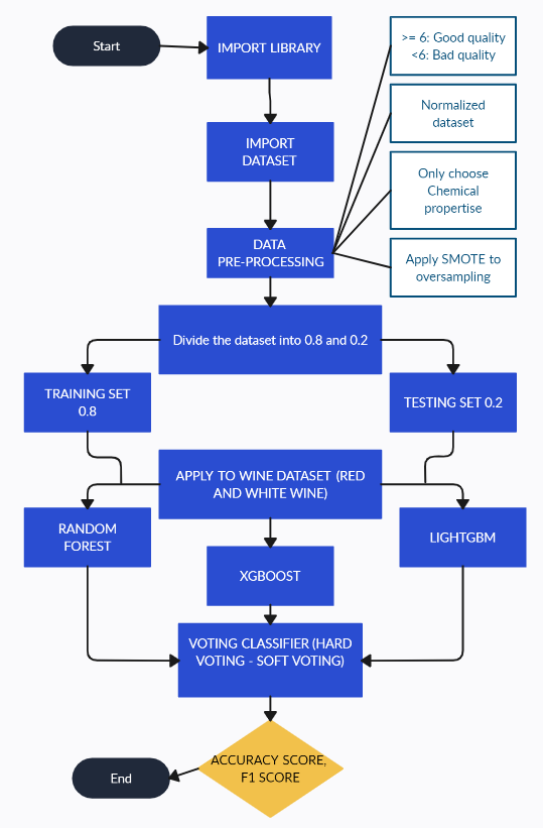

Figure 4: Explaination step

4.3.1. Red Wine

Normalized data for Red wine in figure 5.
Apply SMOTE for Red wine display as figure 6 and 7.

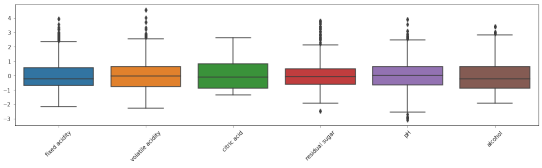

Figure 5: Normalized data with Red wine

Running 10 times and taking the average accuracy and F1 score. The authors have concluded that with Red wine, Soft Voting is helping stable and high accuracy with $F1 = 0.807$.

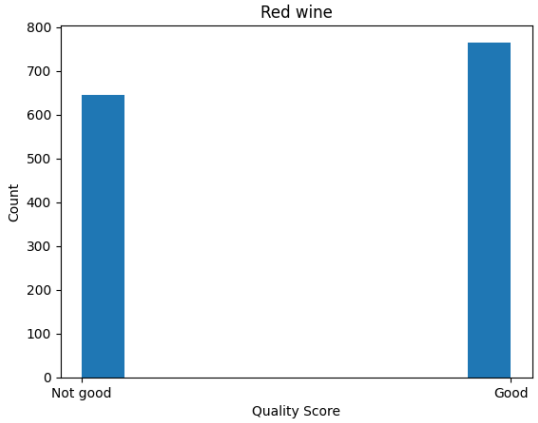

Figure 6: The origin imbalanced label in Red wine

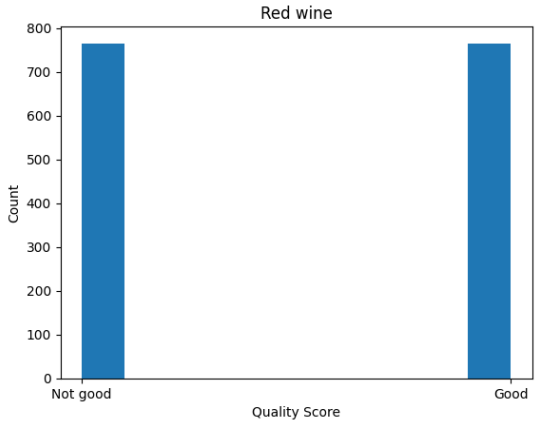

Figure 7: Applied SMOTE to get balanced data for Red wine

Algorithms	F1-train	F1-test	Acc-train	Acc-test
Decision Tree	1	0.767	1	0.762
Random Forest	1	0.8	1	0.8
Logistic Regression	0.734	0.74	0.746	0.748
Support Vector Machine	0.748	0.734	0.762	0.745
K-Nearest Neighbors	0.856	0.681	0.875	0.731
Gaussian Naive Bayes	0.685	0.694	0.713	0.715
AdaBoost	0.766	0.731	0.775	0.737
Gradient Boosting	0.842	0.759	0.846	0.761
XGBoost	1	0.801	1	0.8
LightGBM	0.992	0.799	0.992	0.797
Hard voting	1	0.803	1	0.803
Soft voting	1	0.807	1	0.805

4.3.2. White Wine

Normalized data for White wine in figure 8
Apply SMOTE for White wine display as figure 9 and 10.

Running 10 times and taking the average accuracy and F1 score. The authors have concluded that with White

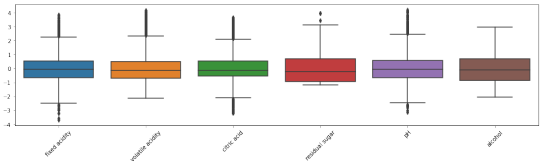

Figure 8: Normalized data with White wine

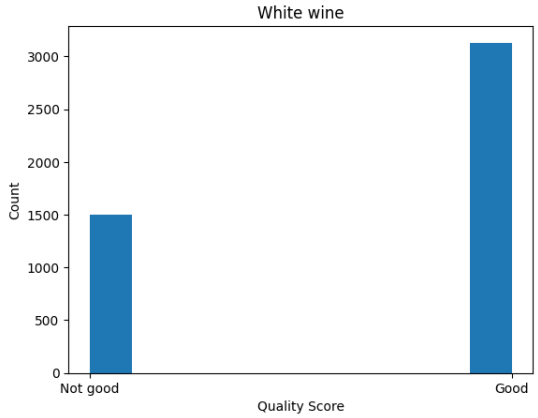

Figure 9: The origin imbalanced label in White wine

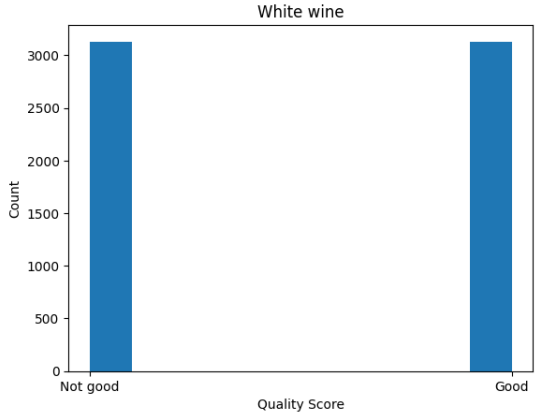

Figure 10: Applied SMOTE to get balanced data for White wine

wine, Soft Voting is helping stable and high accuracy with $F1 = 0.865$.

5. Conclusion and Future Work

Correct understanding of red wine and white wine with only chemical properties is the basis and premise for quality success. Random Forest, XGBoost and LightGBM achieve higher classification accuracy than the other machine learning algorithms. Then with Voting, the accuracy estimated 0.807 for Red and 0.865 for White wine. For further work, applying deep learning, tuning parameters for XGBoost and LightGBM or using SMOTEENN instead of SMOTE is promising to give a better solution for predicting wine quality.

Algorithms	F1-train	F1-test	Acc-train	Acc-test
Decision Tree	1	0.806	1	0.807
Random Forest	1	0.863	1	0.865
Logistic Regression	0.717	0.707	0.722	0.715
Support Vector Machine	0.772	0.749	0.774	0.753
K-Nearest Neighbors	0.91	0.789	0.917	0.819
Gaussian Naive Bayes	0.695	0.681	0.704	0.692
AdaBoost	0.755	0.735	0.761	0.743
Gradient Boosting	0.811	0.777	0.813	0.78
XGBoost	0.976	0.856	0.976	0.856
LightGBM	0.932	0.839	0.932	0.84
Hard voting	0.979	0.861	0.979	0.861
Soft voting	0.985	0.865	0.985	0.865

Acknowledgments

We would like to acknowledge and thank Associate Professor. Triet M. Tran for the enthusiasm in teaching us. We would like giving sincerely thank to Nghia T. Le, Duy K. Le, Le T. Do for being supervised for helpful discussions and organizing seminars to help us finish this work.

References

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decision support systems 47 (2009) 547–553.

[2] A. Trivedi, R. Sehrawat, Wine quality detection through machine learning algorithms, in: 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), IEEE, 2018, pp. 1756–1760.

[3] S. Kumar, K. Agrawal, N. Mandan, Red wine quality prediction using machine learning techniques, in: 2020 International Conference on Computer Communication and Informatics (ICCCI), IEEE, 2020, pp. 1–6.

[4] P. Bhardwaj, P. Tiwari, K. Olejar Jr, W. Parr, D. Kulasiri, A machine learning application in wine quality prediction, Machine Learning with Applications 8 (2022) 100261.

[5] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, " O'Reilly Media, Inc.", 2022.

[6] P. Bruce, A. Bruce, P. Gedeck, Practical statistics for

data scientists: 50+ essential concepts using R and Python, O'Reilly Media, 2020.