Wine Type Classification Using Random Forest Model

Yiyang Cao[1, †], Haoyu Chen[2, *, †], Bochun Lin[3, †]

[1]College of Letter and Science, University of California, Davis, California, 95618, The United States

[2]School of Science, Rensselaer Polytechnic Institute, Troy, New York, 12180, The United States

[3]School of Management Science and Information Engineering, Changchun, Jilin, 130022, China

* Corresponding Author Email: Chenh20@rpi.edu

[†]These authors contributed equally

Abstract. Wine Type Classification indicates that its indexes can ascertain the wine category. Therefore, it can be applied in modern industrial wine production and identification to reduce the rates of inferior products or to terminate the sale of homemade hooch or watered-down cheap alcohol. This paper explores Random Forest to classify wine. Since there are null values in the data, we first input the wine quality dataset and drop out the null values. Standard scaling is ignored because it expands the differences of data and the original datas are special for its distribution to deviation. Then, principal components analysis (PCA) is applied to reduce the dimensions of variable attributes. Finally, we perform random forest to the dataset to see the precision and F1 scores. We compare our methods with logistic regression, SVM, and naive Bayes model. The accuracies of these methods are 0.884375, 0.88125, and 0.884375, respectively. Our result shows that the random forest strategy generates promising accuracy of wine classification. Therefore, Random Forest can predict the industrial product quality and even can recognize the wine type with a high precision rate.

Keywords: Wine Type Classification, Random Forest, PCA, SVM, Bayes, Logistic Regression.

1. Introduction

Wines have been extremely popularized and widely favored by human beings worldwide from the past to the recent age. The import value of BC wine from California to British increased from Can$26.1 million in 2000 to Can$51.9 million in 2014 (98% higher) [1]. The average per capita annual consumption increased by 42.3% from 14.9l in 2000 to 21.2l in 2013 in the province [2]. Hence, the authentication of wines is critical to secure consumers' interest. Events like fake wines are sold as premium wines have been no longer astounding, which evincing the necessity of wine classification. It forms a protection procedure to regulate the market and correct mistakes.

Wine classification identifies which type of wine a given wine sample belongs to. Common classes of wine are white wine, red wine, beer, and cocktail. Ingredients of wines from distinct classes differ in chemical compounds and the percentage of those compounds. Therefore, by analyzing the basic ingredients that the sample has and comparing it with the database, the quality of this wine sample can be indicated for identifying whether it is pure or mixed with other wines.

Lately, research on wine classification have mostly used machine learning strategies like principal components analysis (PCA), artificial neural networks (ANN), decision trees, and random forest to construct systems of analyzing various factors that contribute to the identification. PCA -based methods are frequently used to process the data since PCA patterns can reduce the dimensions of the dataset. For example, Milovanovic et al. used PCA to classify 31 wine samples from the South Moravia region in the Czech Republic[2]. ANNs can suggest the relationship between every element of a sample and the final class the sample belongs to. Hosu et al. use ANN to "predict the antioxidant activity of different wines, providing an efficient and cost-effective way of determining one of the most appreciated characteristics of wine." [3] Decision trees and random forests are also widely used models [4] to present all the probable consequences that an issue can end with. Hu et al. [5] use Decision trees and random forests to classify their data to categorize wines. Nevertheless, data sets' sizes are not adjusted in these researches, which results in low accuracy for wine type classification.

This paper uses the random forest to classify wines into red wines and white wines. First, we balance the number of red wine and white wine samples and utilize the pre-processed dataset to make precise classification to ensure that our procedure can correctly classify red and white wines. Then, we use PCA to reduce the dimensions of variable attributes. Next, we perform random forest to the dataset to see the precision and F1 scores. We set 30 testing examples with GridSearchCV to realize prediction with bootstrap. The feature values are set as default. Finally, we compare Random Forest with SVM, Logistic Regression, Naïve Bayes with Cross-Validation to evaluate the related performances. The receiver operating characteristic curve (ROC curve) shows that the random forest(RF) curve is almost invisible for being close to the left and top grid. The experimental results show that Random Forest significantly outperforms the other machine learning methods. Therefore, Random Forest can predict the industrial product quality and even can recognize the wine type with a high precision rate.

2. Method

This section introduces our random forest model for our combined white and red wine dataset. Since the dataset we obtained has two parts, and there is no variable to symbolize their type, so we add one more variable called "type" with red wine equal to 1 and white wine equals 0 (Sec. A). Then, we apply the PCA method to reduce the dimension. Since there are 12 attributes, we need to find some redundant attributes and drop them out (Sec. B). Finally, we apply a random forest model to fit our dataset (Sec. C).

2.1. Preprocess of Wine Samples

To develop further examinations and make predictions, we add a new attribute, "type" is to each sample. Samples containing Na values are excluded from the dataset. Red wine samples' type attributes are assigned to 1, and white wine samples' type attributes are assigned to 0. The heading 5 rows of the "type" variable in the dataset can be shown in the Table I.

Table 1. First Five rows of type

	type
0	1
1	1
2	1
3	1
4	1

By applying corr() and describe(), functions in panda package of python, to both red wine and white wine samples, pairwise correlation of all columns and basic statistical details like percentile, mean, standard deviation etc. of the dataset are obtained and plotted by using pairplot() function in seaborn. The sample pairplot of randomly chosen three variables are shown as Fig. 1.

2.2. PCA Method

Principal Component Analysis (PCA) is an unsupervised dimensionality reduction method. It produces a lower-dimension version of the dataset containing as many variations as possible. Since some features seem to be redundant, they may not influence the dependent variable. Denote

$$p = \text{different columns in dataset} \tag{1}$$

$$n = \text{different rows in the dataset.} \tag{2}$$

Then, in mathematical meaning, PCA finds the linear combination of mutually uncorrelated p features with maximal variance. The PCA chooses the number of final features that remain to find the minimum number of features that first reaches 95% cumulative variance. The work can be shown

in Fig. 2. After applying that number of features like the number of components parameters equals to (2) in PCA, it will result in the covariance matrix with the dimension equals to the square of (1), and Fig. 3 is a sample with choosing any two of them to draw in the 2-d plot.

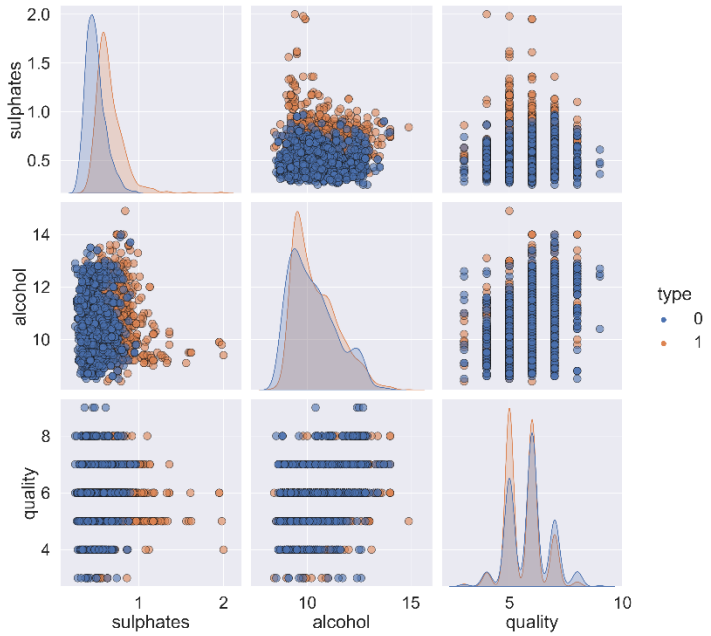

Figure 1. Pairplot of random chose three features

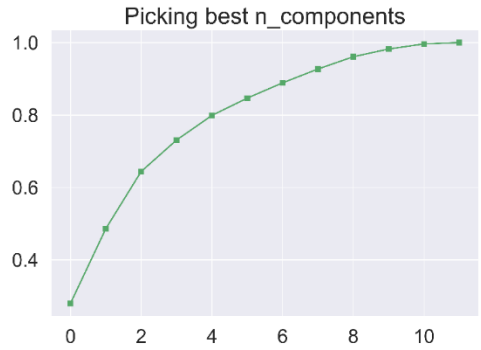

Figure 2. Pick the best number of features for PCA

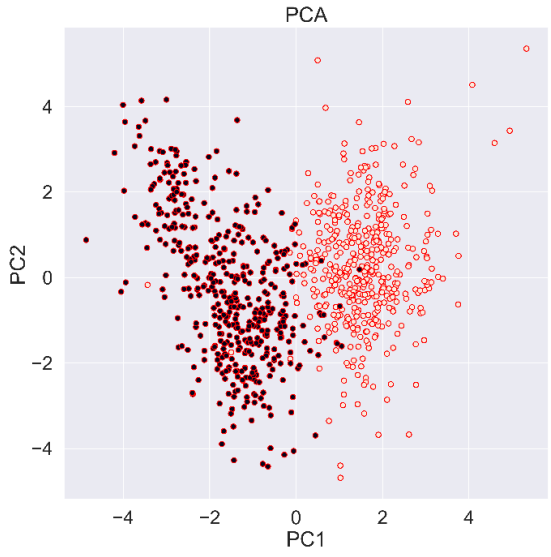

Figure 3. Sample PCA plot for random two features

2.3. Random Forest

RF is an ensemble learning method used to build a classifier model to split the dataset into several small-scale sample datasets [6]. In the RF, Bootstrap aggregation is used for it is proved to be better in most situations of reducing variance to reduce the probability of over-fitting. Based on these datasets, the ensemble models are built. Then, RF generates several decision trees without pruning, forming a forest that owns a predicted value for a particular data sample. The value predicted to be the majority-voting of trees is the final predicted value. Since the datasets have low bias and high variance, Random Forest can perform well by using bagging to gain maximum votes from predicted values of decision trees.

As we can acknowledge, RF is a model based on a decision tree model. Therefore, before RF, explaining the decision tree is necessary.

Mathematically, the Decision Tree Classifier generates the rules by dividing the IF-ELSE based nodes then merges them into a tree. [14] For being the non-linear model, the terminating node performs well on non-linear data while coming across over-fitting problems causing performance degradation. This has a cost of $O\left(n_{features} n_{samples} \log\left(n_{samples}\right)\right)$, where $n_{samples}$ is the number of samples, $n_{features}$ is thenumber of features, at each node. Summing the cost at each node leads to a total cost over the entire trees of $O\left(n_{features} n_{samples}^2 \log\left(n_{samples}\right)\right)$.

The complexity of decision trees, in general, can be considered as how to construct a balanced tree. The run time cost to construct a balanced binary tree is $O\left(n_{samples} n_{features} \log\left(n_{samples}\right)\right)$and query time $\mathrm{O}\left(\log\left(\mathrm{n}_{\mathrm{samples}}\right)\right)$. Although the tree construction algorithm like Decision Tree Classifier tries to build balanced trees, but they are not always balanced as expected. If the subtrees keep approximately balanced, the cost at each node consists of searching $O(n_{features})$ to find the features that provide the maximum reduction in entropy.

Train vector $x_i \in R^n, i = 1, \ldots, l$ and a label vector $y \in R^l$ given, decision trees recursively partition the feature space to group together the samples with the same labels or similar target values. Let the data at node n be represented by Q_m with N_m samples. For each candidate, split $\theta = (\mathrm{j}, t_m)$ consisting of a feature j and threshold t_m divide the data into $Q_m^{left}(\theta)$ defined in (3) and $Q_m^{right}(\theta)$ defined in (4) subsets.

$$Q_m^{left}(\theta) = \{(\mathrm{x}, \mathrm{y}) | x_j \leq t_m\} \tag{3}$$

$$Q_m^{right}(\theta) = Q_m \backslash Q_m^{left}(\theta) \tag{4}$$

Then an impurity function Φ is used to calculate the quality of the candidate split of node m.

$$G(Q_m, \theta) = \frac{N_m^{left}}{N_m} \Phi(Q_m^{left}(\theta)) + \frac{N_m^{right}}{N_m} \Phi(Q_m^{right}(\theta)) \tag{5}$$

Choose the parameters θ^* given in (6) with minimum impurity.

$$\theta^* = argmin_\theta G(Q_m, \theta) \tag{6}$$

Recurse for subsets $Q_m^{left}(\theta^*)$ and $Q_m^{right}(\theta^*)$ until the maximum allowable depth is reached, $N_m < min_{samples}$ or $N_m = 1$

For the wine quality classifier experiment target is a classification outcome in this paper. Only the classification part is presented. If the outcome takes on values $0,1, \ldots, k-1$, for node m, suppose

$$p_{mk} = 1/N_m \sum_{y \in Q_m} I(y = k)$$

be the rate of class k observations in the node m. If m is the terminating node, we use Gini impurity defined in (7) by default but offer Entropy defined in (8) as an alternative for classification.

$$H(Q_m) = \sum_k p_{mk} (1 - p_{mk}) \tag{7}$$

$$H(Q_m) = -\sum_k p_{mk} \log (p_{mk}) \tag{8}$$

Multiple trees are created by constructing trees in randomly selected subspaces of feature space. [7]

In Random Forest, to ensemble the predictions and generate the final results, it is necessary to calculate feature importance for the decrease in the node . The node probability can be calculated through deviding the number of samples that reach the node by the total number of samples. The higher ratio comes the more important feature.

For each decision tree, Scikit-learn calculates a node's importance given in (9) using Gini impurity given in (7), assuming it to be a binary tree with only two child nodes i and j:

$$n_{i_j} = w_j C_j - w_{lj} C_{lj} - w_{rj} C_{rj} \ , \tag{9}$$

where n_{i_j}, w_j, C_j, w_{lj}, C_{lj}, w_{rj}, C_{rj} are the importance of node j, weighted number of samples reaching node j, the impurity value of node j, weighted number of samples reaching child node from left split on node j, the impurity value of child node from left split on node j, weighted number of samples reaching child node from right split on node j, the impurity value of child node from right split on node j respectively. For the concern of simplicity, Gini impurity $H(Q_m) = \sum_k p_{mk} (1 - p_{mk})$ is denoted by n_{i_j}, w_j, C_j, w_{lj}, C_{lj}, w_{rj}, C_{rj}, etc.

Then the importance of each individual feature given in (10) on the decision trees is computed.

$$f_{i_i} = \frac{\sum_{node\ j\ splits\ on\ feature\ i} n_{i_j}}{\sum_{k \in all\ nodes} n_{i_j}}, \tag{10}$$

where f_{i_i}, n_{i_j} are the importance of feature i, the importance of node j respectively.

Then with dividing by the sum of all feature importance values, they can be normalized to a value given in (11) between 0 and 1

$$norm_{f_{i_i}} = \frac{f_{i_i}}{\sum_{j \in all\ features} f_{i_j}} \tag{11}$$

At the Random Forest level, the final feature importance is its average, which is defined as a discriminant function [7] given in (12) over all the trees. The sum of the feature's importance value on each tree is calculated and divided by the total number of trees:

$$RF_{f_{i_i}} = \frac{\sum_{j \in all\ trees} norm_{f_{i_{ij}}}}{T}, \tag{12}$$

Where $RF_{f_{i_i}}$, $norm_{f_{i_{ij}}}$, T, are the importance of feature i calculated from all trees in the RF model, the normalized feature importance for i in tree j, the number of the total trees respectively.

In this paper, we apply scikit-learn in our program in python.

Scikit-learn library Random Forest provides three fundamental parameters in Python, which are discussed in the following.

1) Number of estimators: The number of decision trees used in the Random Forest model. In this paper, we utilize the gridsearchCV to adjust the number of decision trees, and eventually select 30 as the number of decision trees generated for the research. Theoretically, the more decision trees mean, the better performance, but also take more run time. Furthermore, if the number exceeds a certain point, it even causes over-fitting with high variance.

2) Bagging: It is one of the Random Forest basic mechanisms, which takes the votes from different trees. Therefore, random Forest can be approximately recognized as a combination of bagging and decision trees.

3) Confusion Matrix: Confusion matrix is used to reveal the correct and incorrect predictions of the Random Forest model [10]. And the sample confusion matrix is shown in the Table II.

Table 2. General Confusion Matrix

	Predicted Bad	Predicted Good
Actual Bad	TP	FN
Actual Good	FP	TN

Where TP/True Positive, FN/False Negative, FP/False Positive, TN/True Negative are case was negative and predicted negative, the case was positive but predicted negative, the case was negative but predicted positive, the case was positive but predicted positive respectively.

3. Experimental settings

This section describes the experiment details of our experiment. We first briefly introduce the experimental settings, including dataset and baselines that we used to compare with chosen model random forest (Sec. A and Sec. B). Then, we discuss the results with accuracy and AUC for each model (Sec. C).

3.1. Dataset

The "wine quality" dataset provided by UCI Machine Learning Repository [8] contains 4898 samples. 1600 of them are red wines, and 3298 are white wines. Each wine sample has 12 attributes which are percentages of 12 elements. Therefore, 1600 red wine samples and 1600 white wines samples are chosen for data analysis to ensure the balance of the number of samples.

3.2. Baselines

To verify the accuracy of our chosen model Random Forest, we need to compare the obtained accuracy, precision, F1 score, confusion matrix, and AUC with other models. Here we apply Logistic regression [9], SVM [10], and NaiveBayes [11] as the baselines.

1) Logistic Regression: Logistic Regression is one of the well-known classification methods. It typically results in the probability that the response Y belongs to one particular class. In our experiment, logistic regression estimates the possibility that the target wine is categorized to the correct type.

2) SVM: Support vector machine, abbreviated as SVM, is one of the machine learning models used for classification. It was first formulated in the 1960s but finally implemented in the 1990s by Valdimir Vapnik [12][13]. The main idea of SVM is to find the widest street (hyperplane) through (separating) the data. For example, in a 2-d hyperplane, this is a line.

3) NaiveBayes: NaiveBayes models are one of the generative models, which means they model the distribution of x for each class and then assign the label to each class. By the way, Naivebayes models compute the conditional probability. By given a problem with a vector X represents n features $X = x_1, x_2, \dots, x_n$ and assigned to instance probabilities formula (13). By Bayes' theorem, the formula (13) can be decomposed to formula (14) [14]. These models represent how the data was generated and enabled the generation of samples for each class.

$$\mathrm{P}(\mathrm{C_n}|\mathrm{X}) \tag{13}$$

$$\mathrm{P}(\mathrm{C_n}|\mathrm{X}) = \frac{P(X|C_n)P(C_n)}{P(X)} \tag{14}$$

3.3. Evaluation Metrics

Here, we use accuracy obtained from the confusion matrix and area under the curve (AUC) [15] from ROC, which is used to visualize the Random Forest model's performance and other baseline models.

1) Accuracy: Here, the accuracy is calculated by the value in the confusion matrix. From the confusion matrix, we can obtain the True Positive, False Positive, True Negative, and False Negative.

And the combination of True Positive samples (TP) and False Positive samples (FP) will result in Positive samples, which means the category is correctly determined. Meanwhile, the combination of the True Negative (TN) and False Negative (FN) will result in Negative samples, which means the category is wrongly determined. Then, the equation to calculate the accuracy through those data is shown by (15).

$$ACC = \frac{TP+TN}{P+N} \tag{15}$$

2) AUC: Aera under curve (AUC) is calculated by the area under the ROC curve. ROC (Receiver Operating Characteristic Curve) is a visualization of the performance of a classification model. It depicts the balance between true positive rate (TPR) and false positive rate (FPR) at different probability thresholds. Suppose TPR is the recall, calculated as (16), FPR is the probability of a false alarm, calculated as (17).

$$TPR = \frac{TP}{TP+FN} \tag{16}$$

$$FPR = \frac{FP}{FP+TN} \tag{17}$$

4. Experimental results

This section provides detailed results and analysis. Also, depict the ROC plot to visualize the performance of each model.

4.1. Total results analysis

The detailed results for both the random forest model and other baseline models are shown in TABLE III. The variables in a row are corresponding accuracy and AUC.

From the Table Ⅲ, the Random Forest has the highest accuracy among all of the models we apply, even close to 1. The other baselines are all around 0.97. Back to the (14), if the accuracy is very high, it symbolizes that the model can accurately predict the true category. In other words, the model can make the correct category. Thus, the Random Forest model will perform a higher accuracy in correctly classifying the wine type. Consider the accuracy for logistic regression. Since the data are collected directly from wine, it is impossible to exist the linear non-separable, which might explain why logistic regression also performs a high accuracy. As for the SVM, since the data is linearly separable, the kernel is also chosen as 'linear', making the final accuracy high. As for the NaiveBayes, since this is a binary classification, the wines can only be categorized into white or red, so we implement Gaussian distribution. And thus, it results in high accuracy.

Table 3. Final results for all models

	Accuracy	AUC
Logistic Regression	0.971875	0.99
SVM	0.969792	0.99
NaiveBayes	0.958333	0.98
Random Forest	0.992708	1.00

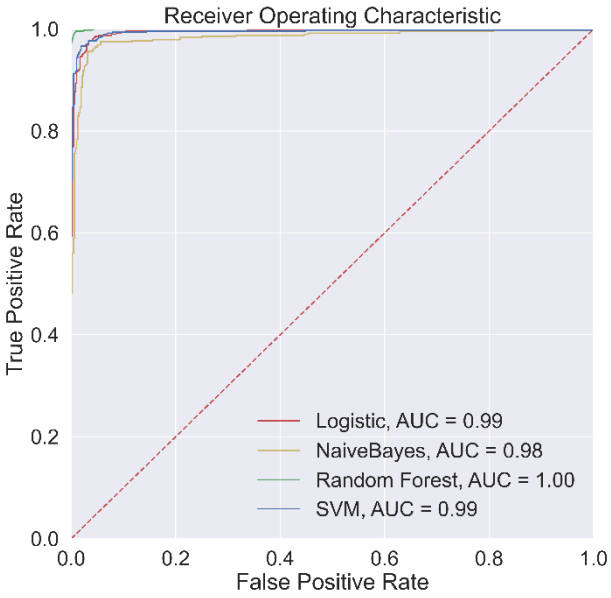

Figure 4. ROC plot for four baseline models

From Fig. 4, which are the ROC plots for Random Forest, SVM, Naïve Bayes and Logistic Regression, we can see that the random forest model has the same an extremely high AUC, approach to 1. The reason might be the Bagging (Bootstrap Aggregation) can reduce the variance, so the prediction accuracy of decision trees has been improved.

5. Conclusions

This paper explored wine quality classification by building a Random Forest model with scikit-learn. We also build a baseline with Support Vector Machine, Logistic Regression, and Naïve Bayes as comparative models. We prove that Random Forest is more suitable for dealing with non-linear data than the other models. Using the gridsearchCV method in sklearn.model_selection, 30 is the best number of estimators, much better than 200, which would possibly cause over-fitting though having astonishingly high accuracy and precision. Our analyses illustrate that using logistic regression and naïve bayes in an inappropriate situation triggers errors and low accuracy.

This research mainly demonstrates the use of Random Foerest to classify different categories of the wine. Compared with other studies, our research can not only be basically applied to commercial wine category classification but be deployed to identify other chemical products as well. In the future, we will continue the research on improving the accuracy and precision of wine quality classification. For our original data sets have only a few different types of wine, the result of the experiment cannot succeed in predicting various types of wine in industrial production. More wine types will be added into the data sets to fulfill the needs of industrial production and trade in the wine market to deal with the challenge.

References

[1] Carew, Richard, Wojciech J. Florkowski, and Ting Meng. “Segmenting wine markets with diverse price functions: evidence from California red and white wines sold in British Columbia.” *Wine Economics and Policy* 6.1 (2017): 48-59.

[2] Milovanovic, M., Žeravík, J., Obořil, M., Pelcová, M., Lacina, K., Cakar, U., ... & Skládal, P. (2019). “A novel method for classification of wine based on organic acids”. *Food chemistry*, *284*, 296-302.

[3] Hosu, Anamaria, Vasile-Mircea Cristea, and Claudia Cimpoiu. "Analysis of total phenolic, flavonoids, anthocyanins and tannins content in Romanian red wines: Prediction of antioxidant activities and classification of wines using artificial neural networks." *Food chemistry* 150 (2014): 113-118.

[4] Hu, Gongzhu, et al. "Classification of wine quality with imbalanced data." *2016 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2016.

[5] Aich, Satyabrata, et al. "A classification approach with different feature sets to predict the quality of different types of wine using machine learning techniques." *2018 20th International conference on advanced communication technology (ICACT)*. IEEE, 2018.

[6] Akanksha Trivedi, Ruchi Sehrawat, "Wine Quality Detection through Machine Learning Algorithms."(2018). *International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering - (ICRIEECE)*

[7] Tin Kam Ho,AT&T Bell Laboratories, "Random Decision Forests".(1995).

[8] IN: https://archive-beta.ics.uci.edu/ml/datasets/wine+quality

[9] A. Trivedi and R. Sehrawat, "Wine Quality Detection through Machine Learning Algorithms, *2018 Internation Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE)*, (2018): 1756-1760.

[10] Yogesh Gupta, "Selection of important features and predicting wine quality using machine learning techniques. "(2017).

[11] S. Kumar, K. Agrawal and N. Mandan, "Red Wine Quality Prediction Using Machine Learning Techniques," *2020 International Conference on Computer Communication and Informatics (ICCCI)*, (2020): 1-6.

[12] Cortes, C., Vapnik, V.Support-vector networks. *Mach Learn* **20**, (1995): 273-297.

[13] T. Dai and Y. Dong, "Introduction of SVM Related Theory and Its Application Research," 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), (2020), pp. 230-233

[14] Berrar, Daniel. "Bayes' theorem and naive Bayes classifier." *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands* (2018): 403-412.

[15] J. -S. Lee, "AUC4.5: AUC-Based C4.5 Decision Tree Algorithm for Imbalanced Data Classification," in *IEEE* Access (2019): 106034-106042.

[16] IN: Scikit learn: https://scikit-learn.org/stable/modules/tree.html#mathematical-formulation