Predicting the primary category of wine based on physicochemical qualities

Emilija Trajkovska
89221090@student.upr.si
UP FAMNIT
SI-6000 Koper, Slovenia

ABSTRACT

The aim of this project is to predict the primary category of wine, based on its physicochemical data. The category can be either red or white since it will give a bigger contrast contributing to better model accuracy. For this project, two separate datasets were used which were taken from the UCI Machine Learning Repository [4] and OpenML [5]. After merging, the dataset contained 5852 instances of red wine and 1827 instances of white wine. The features in the dataset consist of: category of wine (either red or white wine), numerical score made by wine experts and 10 physicochemical properties such as alcohol, free sulfur dioxide, total sulfur dioxide, density, chlorides, sulphates, residual sugar and pH. Firstly, proper data preparation was done. Then, the dataset was balanced to have somewhat equal samples of red and white wine. To split the data, the techniques of k-fold cross-validation mode (k-fold cv), and percentage split (holdout method) were used. And lastly, the following data mining classification models were used for predicting the color of the wine: Decision Tree Classifier, Random Forest Classifier, Logistic Regression, Histogram-based Gradient Boosting, K-Nearest Neighbourhood, and also K-Means Clustering as an unsupervised model. As a result, with accuracy of 99.3026%, the most successful prediction was achieved by Histogram-based Gradient Boosting Classification Tree and Random Forest Classifier.

"Nothing more excellent or valuable than wine was every granted by the gods to man." - Plato

1 INTRODUCTION

It is often said that great wine is made in the vineyard, not in the winery. However, winemakers have the ability to modify certain aspects of the wines they produce, such as the level of acidity, sweetness or alcohol, as well as the shelf life [5]. But, can they accurately predict the color of the wine solely based on its physicochemical properties?

Since the 1930s, there has been significant interest in measuring the color of wines, particularly as wines with stronger color were often more expensive [2]. This historical context underscores the objective of this project: to explore the feasibility of predicting wine color accurately based on its physicochemical properties. This can be achieved through the implementation of multiple data mining models. By comparing the results obtained from these models, this project aims to identify the most accurate approach for predicting wine color. Ultimately, this research aims to provide valuable insights into the relationship between physicochemical properties and wine color, potentially enhancing our ability to predict and control color outcomes in the winemaking processes.

2 MATERIALS AND METHODS

2.1 Wine Data

The datasets used, are publicly available for research and can be found on UCI Machine Learning Repository [4] and OpenML [5]. The first one is related to the red and white variants of the Portuguese Vinho Verde wine, and the second one to wines from Unicorn Winery. The merged data set contains 5852 instances of white wine and 1827 instances of red wine. The features include:

- **category of wine** (categorical: 'red', 'white')
- **quality**: Score between 0 (very bad) and 10 (very excellent) by wine experts.
- **fixed acidity**: The acids that naturally occur in the grapes used to ferment the wine and carry over into the wine. (g / dm^3)

Table 1: The physicochemical data statistics of white wine.

Features	Min	Max	Mean
fixed acidity	3.900	11.80	6.885
volatile acidity	0.080	1.100	0.277
citric acid	0.000	1.000	0.332
residual sugar	0.600	65.80	6.368
chlorides	0.009	0.346	0.045
free sulfur dioxide	2.000	289.0	35.39
total sulfur dioxide	9.000	440.0	138.2
density	0.987	1.038	0.994
pH	2.720	3.820	3.186
sulphates	0.220	1.080	0.489
alcohol	8.000	14.20	10.52

Table 2: The physicochemical data statistics of red wine.

Features	Min	Max	Mean
fixed acidity	4.600	15.90	8.320
volatile acidity	0.120	1.580	0.5289
citric acid	0.000	1.000	0.271
residual sugar	0.900	15.50	2.539
chlorides	0.012	0.611	0.087
free sulfur dioxide	1.000	72.00	15.87
total sulfur dioxide	6.000	289.0	46.47
density	0.990	1.004	0.997
pH	2.740	4.010	3.311
sulphates	0.330	2.000	0.658
alcohol	8.400	14.90	10.42

- **volatile acidity**: Acids that evaporate at low temperatures. (g / dm^3)
- **citric acid**: Used as an acid supplement which boosts the acidity of the wine. (g / dm^3)
- **residual sugar**: The amount of sugar remaining after fermentation stops. (g / dm^3)
- **chlorides**: The amount of salt in the wine. (g / dm^3)
- **free sulfur dioxide**: The free form of SO2 which exists as a dissolved gas. (mg / dm^3)
- **total sulfur dioxide**: The amount of free and bound forms of S02. (mg / dm^3)
- **density**: The density of wine juice depending on the percent alcohol and sugar content. (g / cm^3)
- **pH**: How acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic).
- **sulphates**: Amount of potassium sulphate as a wine additive which can contribute to sulfur dioxide gas (S02) levels. (g / dm^3)
- **alcohol**: The percent alcohol content of the wine. (% by volume)

Due to privacy and logistic issues, there is no data about grape types, wine brand, and wine selling price [4]. Table 1 and Table 2 display the data statistics, including the minimum, maximum and mean, for the 11 distinct physicochemical properties of white wine and red wine, respectively.

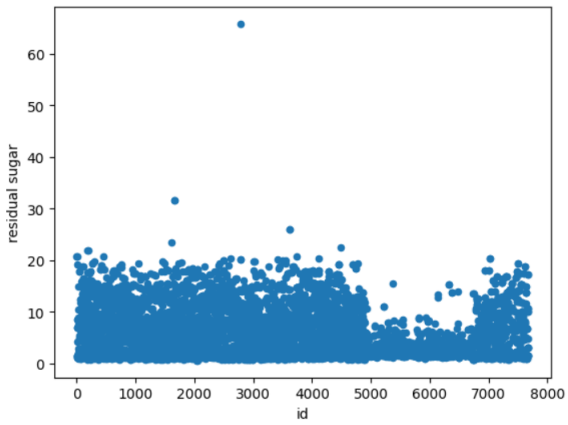

Figure 1: Scatter plot of the feature residual sugar.

2.2 Data Preparation

Firstly, the datasets were merged to combine them into one while ensuring only unique samples were retained. Then, outliers were addressed by using scatter plots and conducting online research to verify the plausibility of the outlier values. An example of an outlier identified can be seen in Figure 1.

Subsequently, to prevent bias in the model, the dataset was balanced to ensure an even representation of each class. Figure 2 depicts the dataset before and after balancing, demonstrating how the distribution of data across categories was equalized.

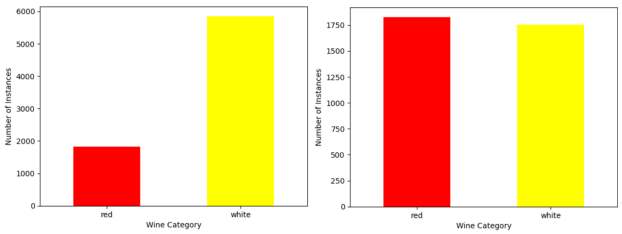

Figure 2: Histograms of before and after balancing.

2.3 Data Mining Approach

For evaluation purposes, two test modes were employed: the k-fold cross-validation mode (CV) and the percentage split (holdout method) mode (TT).
K-fold cross-validation divides and shuffles the dataset into k parts. The model is trained k times, each time using k-1 parts for training and 1 part for validation. This helps assess the model's performance more accurately and reduces the chances of overfitting.
In percentage split, or holdout mode, the dataset is randomly divided into two parts: training and testing dataset. The first set, known as the training set, is used for extracting knowledge (training) by the data mining system. Then the extracted knowledge can then be tested against the second set, referred to as the testing set [3]. Initially, the dataset undergoes 10-fold cross-validation to assess model performance. Following that, the dataset is randomly split into distinct train and test sets, with the train set comprising 80% of the data chosen randomly, and the remaining data assigned to the test set.

2.4 Data Mining Techniques

The data mining models employed on the dataset were:

- **Logistic Regression** is used for binary classification, where the goal is to predict the probability that a given input belongs to one of two classes. It models the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic function, with possible results being 0 and 1.
- **Decision Tree Classifier** partitions the data into subsets based on features at each node of a tree structure. It makes decisions by following the branches of the tree from the root node to leaf nodes, where each leaf represents a class label. Decision trees use simple decision rules inferred from the data to classify instances and can be seen as a piecewise constant approximation for predicting the target variable.
- **Random Forest Classifier** fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the accuracy and control overfitting.
- **Histogram-based gradient boosting** is a variant of gradient boosting algorithms that employs histograms to accelerate the training process. It constructs decision trees sequentially, with each tree correcting the errors of the previous ones. In this approach, instead of using individual data points, the algorithm organizes the data into histograms or bins, which reduces the computational complexity and memory usage.
- **K-nearest neighborhood** classifiers are depended on learning by analogy, this means a comparison between a test tuple with similar training tuples. The training tuples are described by n attributes. Each tuple corresponds a point in an n-dimensional space. All the training tuples are stocked in an n-dimensional pattern space. For an unknown tuple, a k-nearest-neighbourhood classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. k training tuples are called as the k "nearest neighbours" of the unknown tuple [3].
- **K-means clustering** starts by randomly selecting k objects as initial cluster centers. Each remaining object is then assigned to the nearest cluster center based on Euclidean distance between the object and the cluster mean. The algorithm iteratively updates the cluster centers by computing the mean of objects assigned to each cluster. This process continues until the assignments stabilize, forming the final clusters [3].

3 EXPERIMENTAL RESULTS

The models were initialized and fitted on the train set, delivered by the two splitting methodologies. After which, evaluation was performed using the distinct methodologies, cross-validation for the K-fold split and the accuracy function of the models for the train-test split.

Table 3: Performance results of the employed models.

Model	CV accuracy (%)	TT accuracy (%)
LogisticRegression	97.3	97.2
DecisionTreeClassifier	97.8	97.3
RandomForestClassifier	99.3	99.3
HistGradientBoostingClassifier	99.3	99.2
KNearestNeighborsClassifier	92.7	98.1

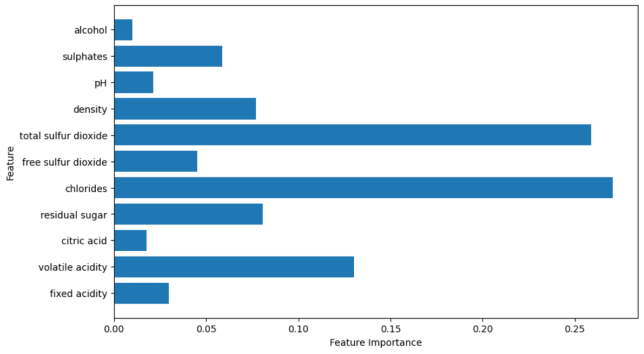

Figure 3: Cross-validation Random Forest Classifier feature importances.

Table 3 displays the obtained results of the models using cross-validation and train-test split methods, respectively.

Also, it clearly demonstrates that the Random Forest Classifier and Histogram-Gradient Boosting Classifier achieve better performance compared to other methods, achieving an accuracy of 99.3026% using both testing methodologies. During the analysis, various graphs were produced, including Figure 3 and Figure 4, which emphasize the significant features used in developing the Random Forest classifier, with cross-validation and train-test approaches, respectively. In addition to these, Decision Trees were also generated as part of the data modeling process. Figure 5 and Figure 6 depict the trees obtained using cross-validation and train-test data splits, respectively. As K-means clustering is an unsupervised learning algorithm primarily used for exploratory data analysis and pattern recognition in unlabeled datasets. It aims to identify natural groupings or clusters within the data based on similarity. Since there are no predefined labels in the clustering process, there is no concept of accuracy as in supervised learning.

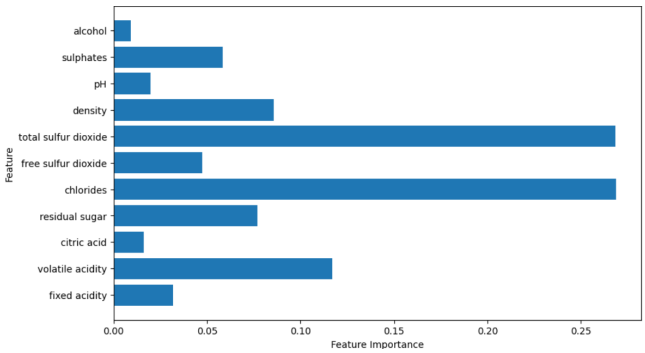

Figure 4: Train-test Random Forest Classifier feature importances.

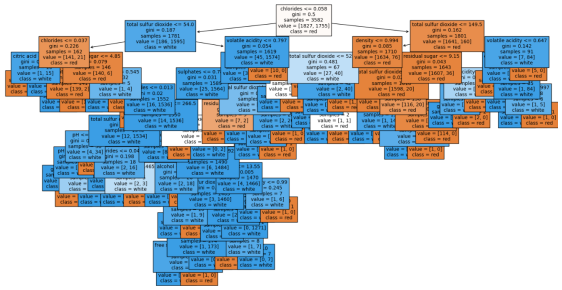

Figure 5: Cross-validation Decision Tree.

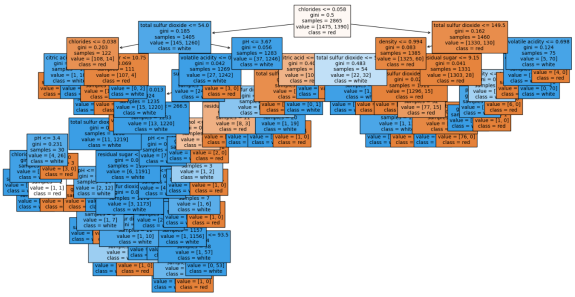

Figure 6: Train-test Decision Tree.

Nonetheless, K-means clustering can be visually represented with clarity. In Figures 7 and 8, the graphical representations of K-means clustering and K-nearest neighbors classification are displayed. In the K-means plot (Figure 7), each point represents a data sample, and its position on the plot is determined by its feature values. The x and y axes provide a visual representation of the features' values. Each data point is depicted with a distinct color, signifying its assigned cluster, while

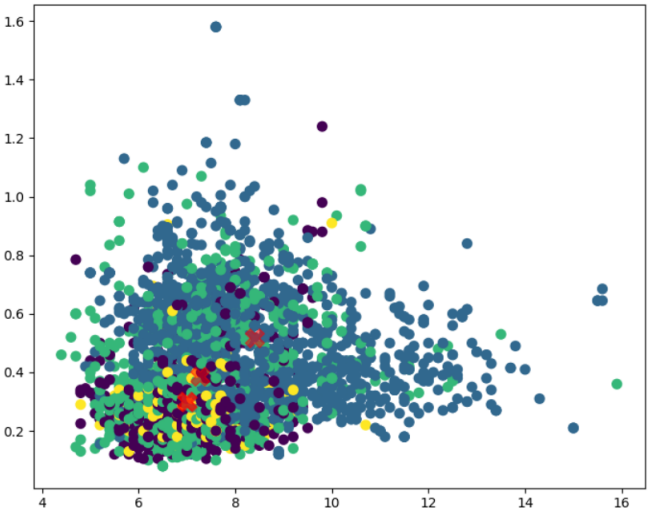

Figure 7: K-Means Clustering.

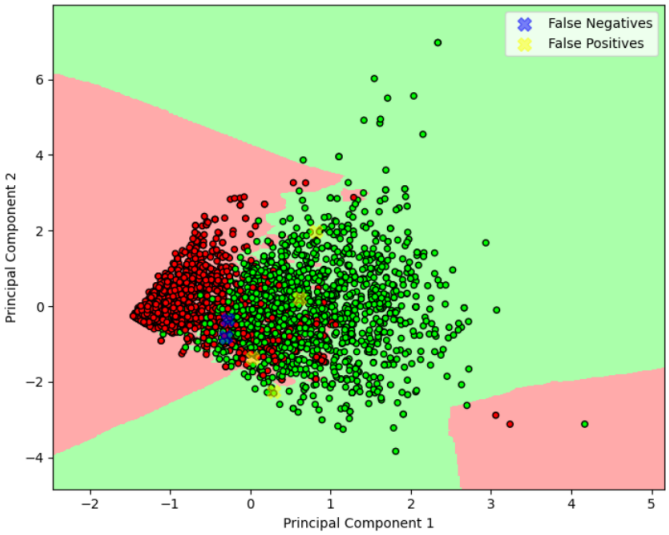

Figure 8: K-Nearest Neighborhood.

the red 'X' markers denote the centroids of the four clusters identified by the K-means algorithm. Moving to the KNN plot (Figure 8), principal component analysis (PCA) is utilized to reduce the dimensionality of the dataset, condensing it into its principal components while preserving crucial information. Here, the x and y axes correspond to the first two principal components derived from the original dataset. The yellow 'X' markers represent false positives, which are instances incorrectly classified, as white wine, by the KNN algorithm, while the blue 'X' markers denote false negatives, which are instances incorrectly classified, as red wine. Additionally, the data points in the bottom-right corner hints at potential outliers that may have been overlooked during data preparing processes.

Moreover, it's worth mentioning that experiments were conducted with various folds for cross-validation and different percentages for train-test splits. While 5-fold and 20-fold cross-validation yielded slightly better accuracy sometimes, 10-fold cross-validation consistently produced the most optimal accuracy for every model. Similarly, train and test splitting with a 70% training set ratio was attempted, but a ratio of 80% for training data resulted in the most optimal performance.

4 CONCLUSIONS

In summary, our analysis consistently highlighted the Random Forest Classifier and Histogram-based Gradient Boosting Classifier as top performers in predicting outcomes. As anticipated, their ensemble nature proved advantageous, leveraging the strengths of multiple models to enhance accuracy. The Random Forest Classifier's ensemble of decision trees effectively combats overfitting and handles noisy data, making it a dependable choice. Its ability to efficiently process large datasets is particularly advantageous for real-world applications in the wine industry. Similarly, the Histogram-based Gradient Boosting Classifier's iterative training approach produces highly accurate models suited for classification tasks.

Our findings underscore the effectiveness of ensemble methods, as evidenced by the superior performance of these classifiers. However, it's worth noting that developing a custom model tailored to specific needs may yield even better results. This approach provides a focused framework for addressing unique challenges and priorities, complementing the strengths of existing models.

RESOURCES

I was inspired to embark on this project during my exploration of Kaggle. Upon encountering a dataset that captured my interest (referenced as [4]), I began searching for projects utilizing this dataset. While I came across numerous studies focused on predicting wine quality, such as the one titled 'Prediction of Wine Quality Using Machine Learning Algorithms' [1], I noticed a gap in research concerning the prediction of wine color. This observation motivated me to undertake the

project. You can explore the outcomes in my Kaggle notebook using this *link*, which is now accessible on GitHub via this *link*.

REFERENCES

[1] Keshab R. Dahal, Jiba Nath Dahal, Huta Raj Banjade, and Santosh Gaire. 2021. Prediction of Wine Quality Using Machine Learning Algorithms. *Open Journal of Statistics* 11, 02 (2021), 278–289. https://www.researchgate.net/publication/350110244_Prediction_of_Wine_Quality_Using_Machine_Learning_Algorithms

[2] F.J. Heredla and M. Guzman-Chozas. 1993. The color of wine: A historical prespective. I. Spectral evaluations. *Food Quality* 16 (1993), 429–437.

[3] M. Kamber J. Han and J. Pei. 2012. *Data Mining Concepts and Techniques.* Waltham, MA, USA: Morgan Kaufmann.

[4] UCI Machine Learning Repository. 2009. Wine Quality Data Set (Red & White Wine). [Online]. https://archive.ics.uci.edu/dataset/186/wine+quality

[5] Unicorn winery. 2022. Red–White-wine-Dataset. [Online]. https://www.openml.org/search?type=data&sort=runs&id=43620&status=active