DATA 1030 Final Project Report – Wine Quality Prediction

Yijing Gao

Brown University - Data Science Initiative

Github: https://github.com/Yijinggao/final-project.git

1 Introduction

Nowadays wine is increasingly enjoyed by a wider range of customers. Portugal has a big variety of local kinds, producing a very wide variety of different wines with distinctive personality. From the far north of the country, exports of Vinho Verde wine have increased a lot during the past years. Quality assessment is a key element within this context in supporting wine making and quality evaluation. Generally, wine assessment is assessed by physicochemical tests or sensory tests. Physicochemical lab test used to characterize wine into the determination of density, alcohol or pH values, while sensory test relies on human experts. The process is very complex and still has room to improve.

This project attempts to build classification models to predict the quality of a particular white wine. The dataset is from the UCI Machine Learning Repository.[1] Two datasets were created, using red and white wine samples from Vinho Verde, where 86% of the wine is white. In this project, we choose the white wine quality dataset to explore. The target variable I chose is the quality of wines, and the quality scores are between 0 (very bad) and 10 (very excellent). In this dataset, 11 physiochemical features of 4898 white wine samples were recorded: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates and alcohol.

Several authors presented case studies for modeling taste preferences or predicting wine quality based on this dataset. In Paulo Cortez et. al, authors utilized sensitivity analysis for exacting knowledge from the neural network or support vector machine models.[2] The SVM model achieved encouraging results for predicting the wine quality, particularly for white Vinho Verde wine, and the majority of the classes present an individual accuracy higher than 90%. Besides, Terence Shin used the dataset for predicting wine quality with several classification techniques: decision trees, random forests, AdaBoost, Gradient Boost, and XGBoost.[3] By comparing the five models, the random forest and XGBoost yield the highest level of accuracy 92%, and XGBoost has a better f1-score for predicting good quality wines. In the presented projects, the approaches are computationally efficient and achieve more than 90% accuracy. Such study is also useful for the economic sector of the production company to improve the revenue and decision making.

2 Exploratory Data Analysis

In this section, several figures are created for exploring the data in a systematic way.

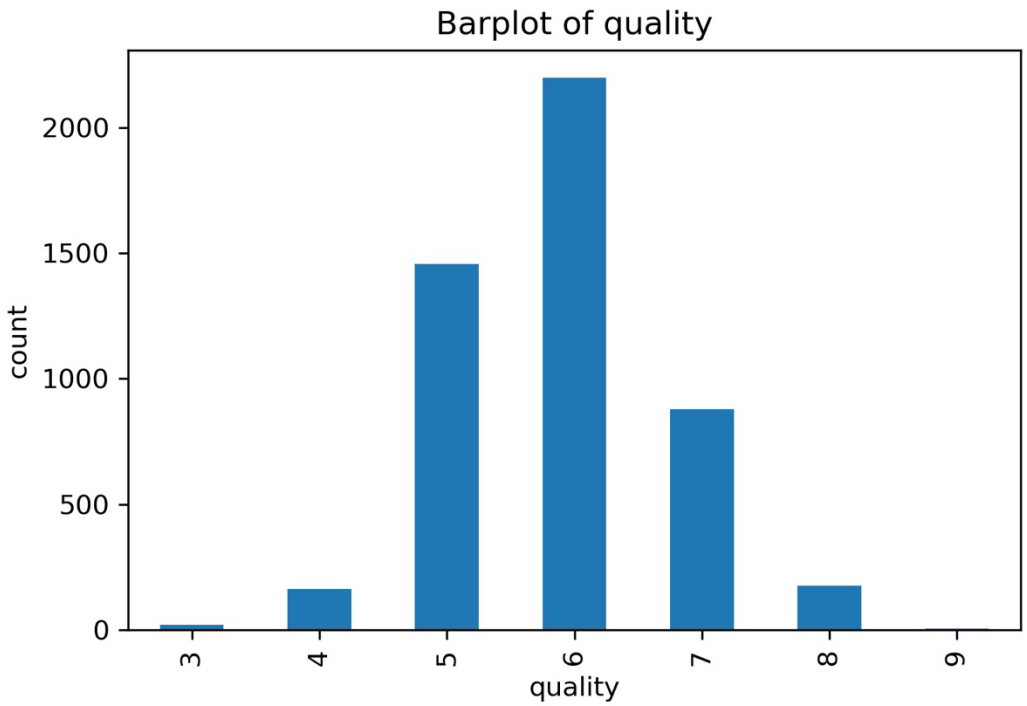

Figure 1

Figure 1 This figure shows each class of the quality variable as a bar. The actual quality scores are from 3 to 9. The classes are ordered and not balanced. From the plot, we can find that there are much more normal wines (graded as 5 or 6) than excellent or poor ones.

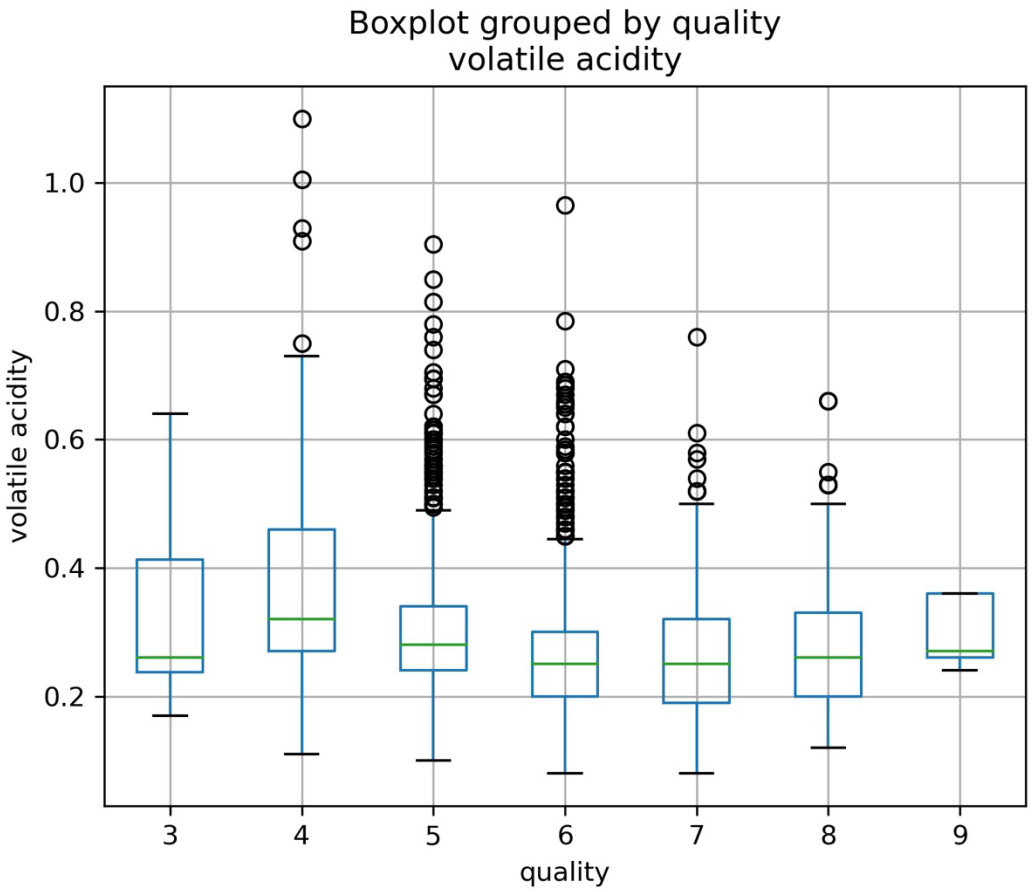

Figure 2

Figure 2 This figure indicates the distributions of volatile acidity in each quality class. The volatile acidity in each class is generally right-skewed. The interquartile range in class 3 and 4 are longer, and class 4 has the largest median value and the maximum value. Based on the data description, too high levels of volatile acidity can lead to an unpleasant, vinegar taste. Additionally, class 5 and 6 has much more outliers than other classes. It is reasonable, since there are more wine samples graded as 5 and 6.

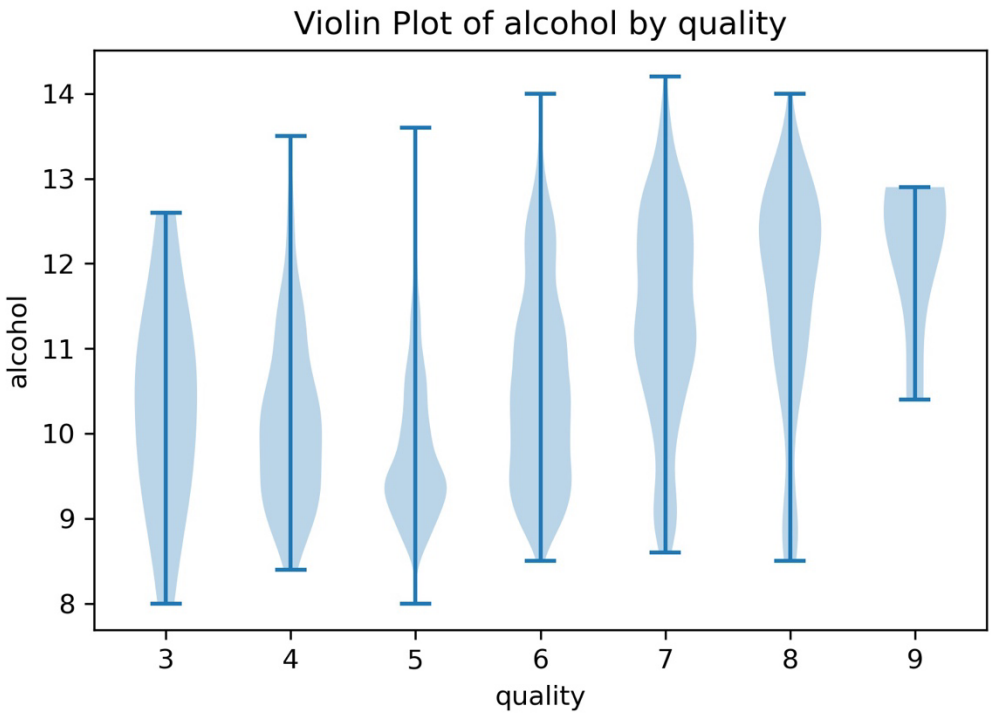

Figure 3

Figure 3 This figure displays the kernel density plots for the percent alcohol content of the wine in each quality class. We can find that among class 7, 8, 9 (can be treated as good quality), the alcohol has a higher mean than that in other quality groups, and the points in those three groups are generally distributed more on the higher alcohol content. The difference among each group is evident.

3 Method

3.1 Data Splitting and Preprocessing

The target variable quality is considered as an ordinal variable, instead of a continuous variable. Based on the exploratory data analysis, the proportions in too low (3 or 4) or too high (8 or 9) categories are small, which may lead to inefficient outputs. Therefore, the wine quality is classified into three categories by combining 3 and 4 into one class (Low), 5 and 6 (Medium) and 7, 8 and 9 into another (High).

In the data splitting section, 20% of the samples are assigned to the test set, and the other 80% are used for stratified k-fold splitting ($k = 3$), since the classes in the quality variable are highly imbalanced, and much more data is in the medium category. The stratified k-fold splitting can help to decrease the variation in balance. The preprocessor fits and transforms the two training folds before transforming validation sets and the test sets. Given that each sample indicates one white wine and samples are made from different grapes and are from different brands, we can consider the dataset to be independent and identically distributed with no group structure or time-series data.

There is no missing value in this dataset. The features are all continuous, and the preprocessor applies StandardScaler to each feature. Most features are not reasonably bounded and they

generally follow a tailed distribution, so compared to the MinMaxScaler, the StandardScaler is preferrable. There are still 11 features in the preprocessed data, since there is no categorical variable in the features.

3.2 Model Selection

By using the data splitting and preprocessing methods above, the machine learning pipeline is developed to train six different machine learning models: a logistic regression with L1 and L2 regularization, a logistic regression with ElasticNet regularization, a random forest classifier, a support vector machine classifier, a k-nearest neighbors classifier and an XGBoost classifier. A cross-validated grid research method is used for tuning over a hyperparameter grid of each model. We apply this method ten times on ten different random states for the splits. For each random state, we find the model with the best hyperparameter combination, and the corresponding accuracy score. Since the target variable has three categories, the accuracy is calculated by using micro-averaged method. Micro-averaging will put more emphasis on the common classes in the dataset since it gives each sample the same importance, so this may be the preferred behavior for multi-label classification problems. And rare classes will not influence the overall scores heavily. Here is the parameters tuned and values tried for each model:

Model	**Parameters**
Logistic Regression (L1 and L2 regularization)	**penalty**: l1, l2; **C**: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3, 1e4
ElasticNet	**C**: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3, 1e4; **l1_ratio**: 0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99
Random Forest	**max_depth**: 1, 3, 5, 10, 20, 100, 300; **n_estimators**: 1, 3, 10, 100; **min_samples_split**: 8, 16, 32, 64, 128
SVC	**C**: 0.01, 0.1, 1, 10, 100; **gamma**: 0.01, 0.1, 1, 10, 100
KNN	**n_neighbors**: 1, 3, 5, 11, 50, 100, 200; **weights**: ‘uniform’, ‘distance’; **metric**: ‘euclidean’, ‘manhattan’
XGBoost	**min_child_weight**: 1, 3, 5, 7; **gamma**: 0.0, 0.1, 0.2, 0.3, 0.4

Figure 4

After the tuning process, the best model parameter combination is returned. We save the accuracy score for each random state to compare the models. The average accuracy scores are calculated to show the best model performance on the ten random states.

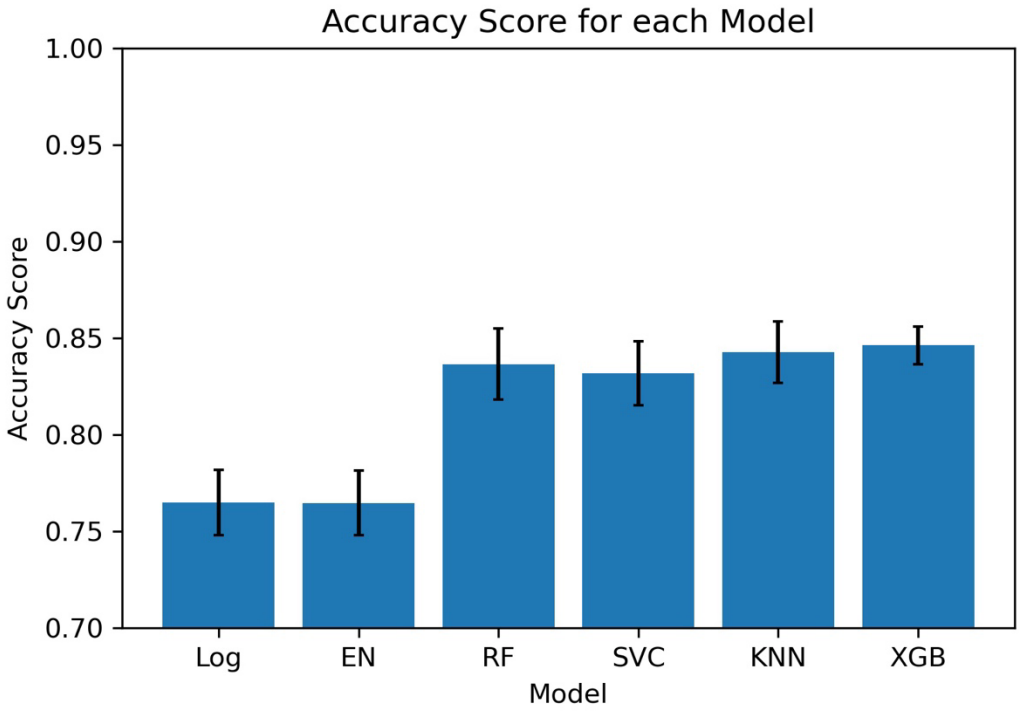

Figure 5

The XGBoost classifier has the highest average accuracy (0.85) on the test set. The k-nearest neighbors classifier and the random forest classifier also perform well. The standard deviation of the accuracy scores for the random forest classifier is highest among these models. In the pipeline step, the random state is added to some models to check the reproducibility and avoid uncertainty.

4 Results

4.1 Evaluation of Models

Based on the best models with tuned parameters, we trained these models with new splits. We save the training sets and test sets with ten different random states. Each time 80% samples are assigned to the training set and 20% are assigned to the test set. As Figure 5 shows, the XGBoost classifiers generally perform best. The recorded best XGBoost models are trained on the new training sets. For the ten random states, the XGBoost classifiers achieve an average accuracy score of 0.85 and a standard deviation of 0.01. In contrast, the baseline models return an average accuracy of 0.74 with a standard deviation of 0.02. The accuracy of XGBoost models is 5.5 standard deviations above the baseline. In addition, the baseline's accuracy is 11 standard deviations below the average accuracy of the trained models.

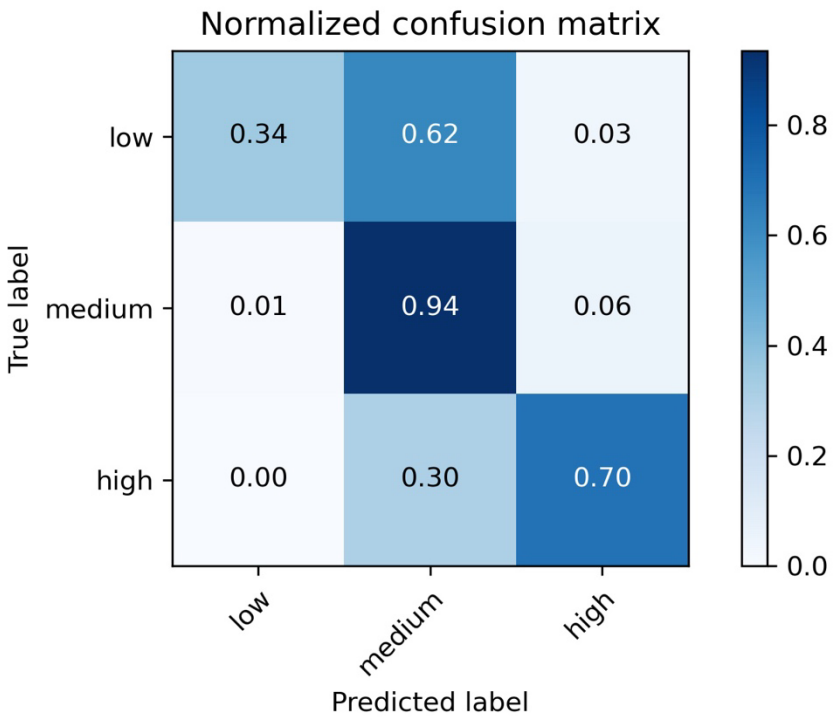

Figure 6

Among ten random states, the XGBoost model with the highest score is used to do the predictions on the test set. As can be seen in Figure 6, the classifier performs best at classifying the medium class and also works well at predicting the high class.

4.2 Model Comparison

From Figure 7, the performance of the six models can be generally divided into two levels: the logistic regression models perform bad, whereas random forest classifiers, support vector machine classifiers, k-nearest neighbors classifiers and XGBoost classifiers generally perform well. The k-nearest neighbors classifier on the largest random state achieves the highest accuracy among all the trained models. As the above Figure 5 shows, the average accuracy score of XGBoost models is highest. Hence the XGBoost model is the most predictive on the wine quality dataset. The k-nearest neighbors models and random forest classifiers also perform well.

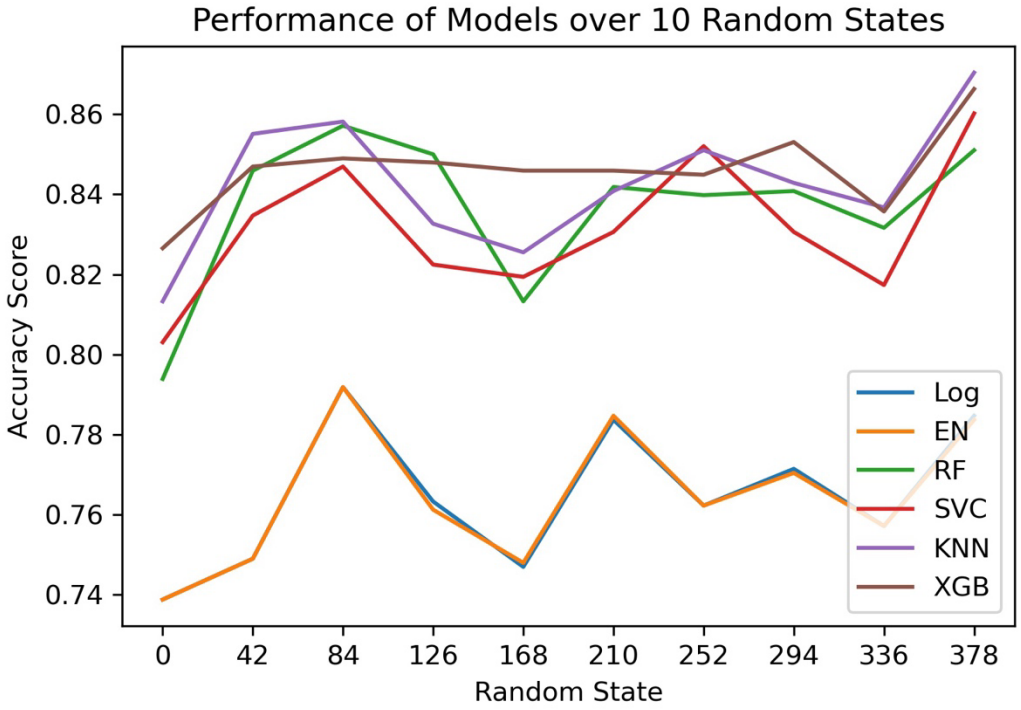

Figure 7

4.3 Feature Importance

In this section, global feature importance is calculated by using the feature importance of random forest models, permutation tests and SHAP values. The results are presented in three figures below:

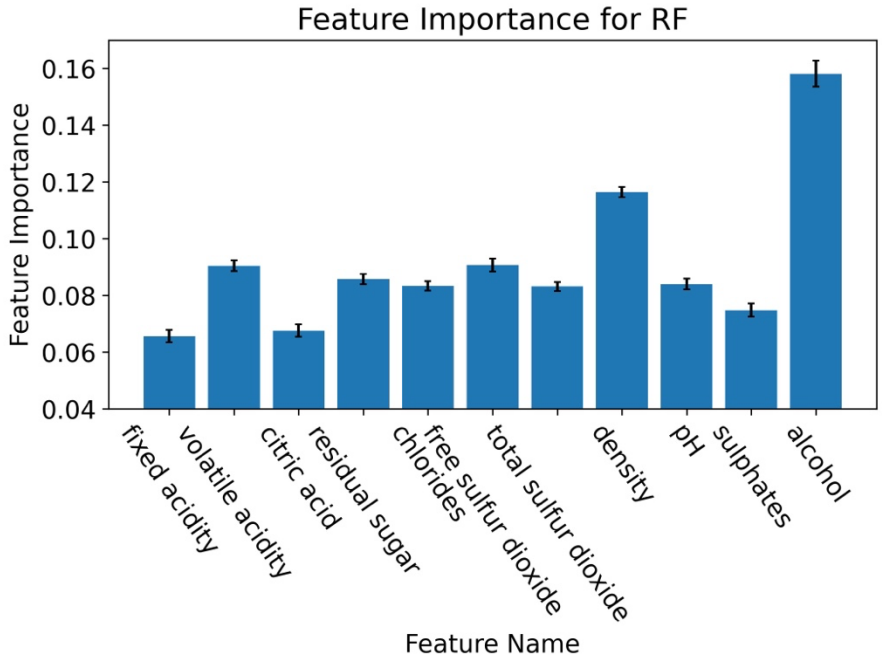

Figure 8

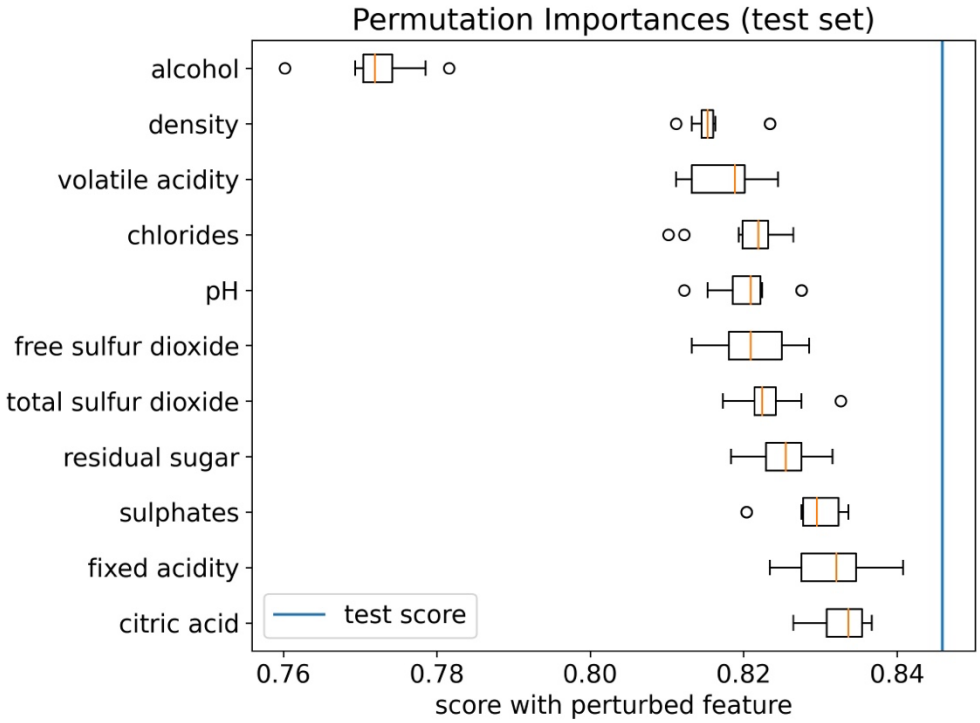

Figure 9

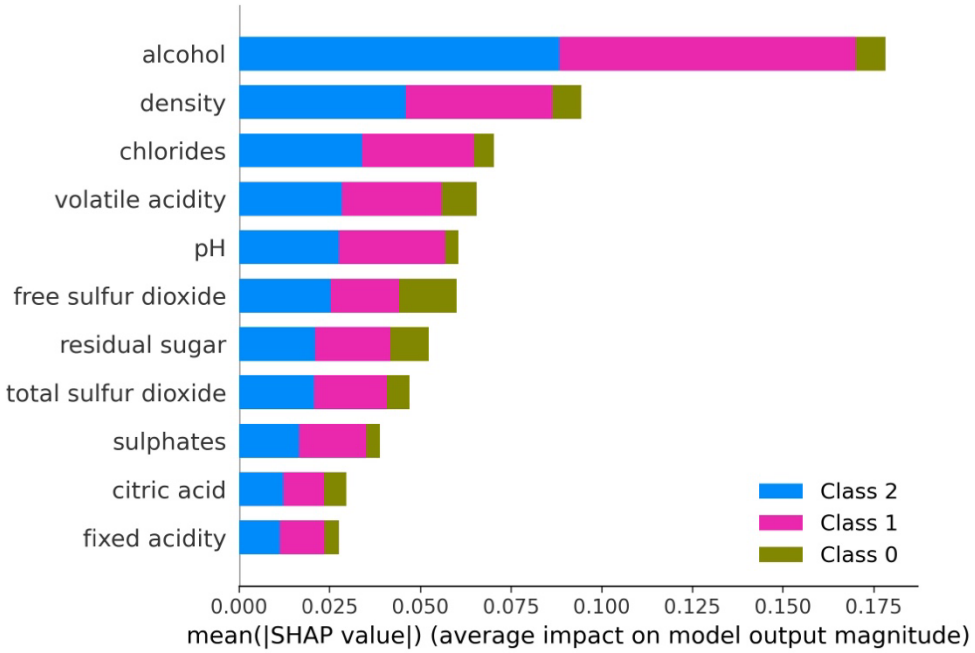

Figure 10

According to the results, alcohol is the most importance feature among the three calculations above. It contributes a lot on predicting class 1 (Medium) and class 2 (High). Using the feature importance of the random forest classifiers and SHAP values, fixed acidity is the least important feature. However, from the average scores of permutation importance, citric acid is the least important feature. Based on Figure 10, free sulfur dioxide has the largest impact on predicting class 0 (Low).

In order to explore the local feature importance, SHAP values are used on some individual cases. Several observations are chosen. Below is the 42nd observation for explaining class 1 (Medium) predictions in the wine quality dataset:

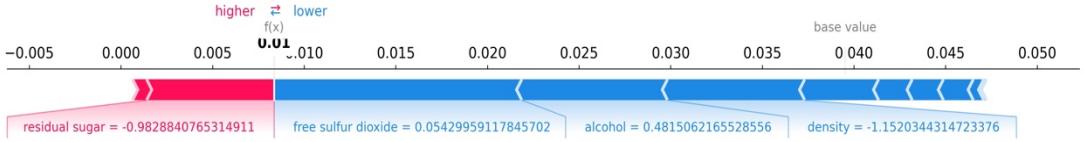

Figure 11

For this observation, we predict 0.01, whereas the base value is 0.040. Free sulfur dioxide, alcohol and density in blue cause the decreased prediction. The biggest impact comes from free sulfur dioxide. Besides, the residual sugar value has a meaningful effect increasing the prediction.

5 Outlook

From all the trained models, XGBoost models and k-nearest neighbors classifiers achieve the top two highest average accuracy scores. The k-nearest neighbors classifiers are easy to interpret and take low calculation time. In the feature importance section, features' impact on the model is

calculated. We could further explore the positive or negative correlations between the features and the model output to improve the interpretability. One limitation that the model presents is the ability to predict the low and high quality, since the target variable is highly imbalanced. It may lead to underestimation of a great wine or overestimation of a low-quality wine. Furthermore, this issue may cause pricing difficulties and economic losses for wineries and liquor distributors. The model could be improved by using penalization, which imposes an additional cost on the model for making classification mistakes on the minority category during training. In addition, we could use over-sampling (such as SMOTE) or under-sampling techniques (such as Cluster) to resample the dataset.[4] It would change the data that we use to build models to have more balanced data. Additional data that accurately reflects high-quality wine (class 8-10) and low-quality wine (class 1-3) could help to improve the model performance.

6 References

[1] UCI Machine Learning Repository: Wine quality data set. (n.d.). Retrieved December 6, 2021, from https://archive.ics.uci.edu/ml/datasets/wine+quality.

[2] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553. ISSN: 0167-9236.

[3] Shin, T. (2020, May 8). *Predicting wine quality with several classification techniques*. Medium. Retrieved October 12, 2021, from https://towardsdatascience.com/predicting-wine-quality-with-several-classification-techniques-179038ea6434.

[4] Contributor, T. T. (2018, November 27). *What is over sampling and under sampling? - definition from whatis.com*. WhatIs.com. Retrieved December 6, 2021, from https://whatis.techtarget.com/definition/over-sampling-and-under-sampling.