Analysis on White Wine Quality Dataset

Final Report

Qi Zhao

DATA-1030 Fall 2022

Github: https://github.com/kevinz8866/1030project.gits

DSI Brown University

12.07.2022

Introduction

Overview

The dataset I will be working on is ”Wine Quality Data Set” from UCI Machine Learning Repositories. I choose to focus on the white wine dataset of the two datasets included for this project. I am interested in this dataset because I had many experiences with wine and always had trouble distinguishing good wines from bad ones. There are a lot of people drinking wine everyday so it is very useful to come up with a quantitative measure on wine quality.

This data set contains 4898 instances and 11 columns of explanatory variables and one column of target variables. The input features are continuous variables which represent numerical values from the lab results. The output variable is a numerical score based on the median of at least 3 evaluations from wine experts, who scored the wine quality between 0 and 10.

The target variable is "quality", which indicates the quality score of the predicted wine. Because quality score is an ordinal variable, this can be both a regression problem and a classification problem. There is order relation among outputs but they are also discrete integers suitable for classification. For this project, I will first train my models using the regression metrics RMSE by minimizing this metric between my output values and the true scores. Then on the best models, I will compute an accuracy score after rounding the model’s predictions to the nearest integers.

Previous work

The author of this dataset used regression metrics to train their models. Then, model outputs are rounded to the closest integers depending on a threshold(T) and correlates with a class prediction. They found that SVM achieved the best MAD and the overall accuracy(T=0.5) is 64.6% on white wine.

A 2021 study "Analysis of white wine using machine learning algorithms" performed similar research. Researchers reported RMSE and accuracy. Many models showed promising results with accuracy around 0.999 and RMSE around 0.03. But they did not reveal hyperparameter choices or describe their pipeline. Some other studies include "Assessing wine quality using a decision tree". Authors used decision tree models and reached accuracy of 60.4%.

Explantionary Data Analysis

The data has 0% missing values. All of my input features are continuous variables, and I have studied their distribution, skewness and shape. All of my input features are positive numbers, and nine of them are normally distributed with a non-zero mean with some outliers three std away from the mean. These include: fixed acidity, volatile acidity, citric acid, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, and sulfates. Two features have skewness and are not distributed normally. These include residual sugar and alcohol. The correlation of all input features is plotted in figure 1. There is no strong correlation over 90% so it is not necessary to remove any features.

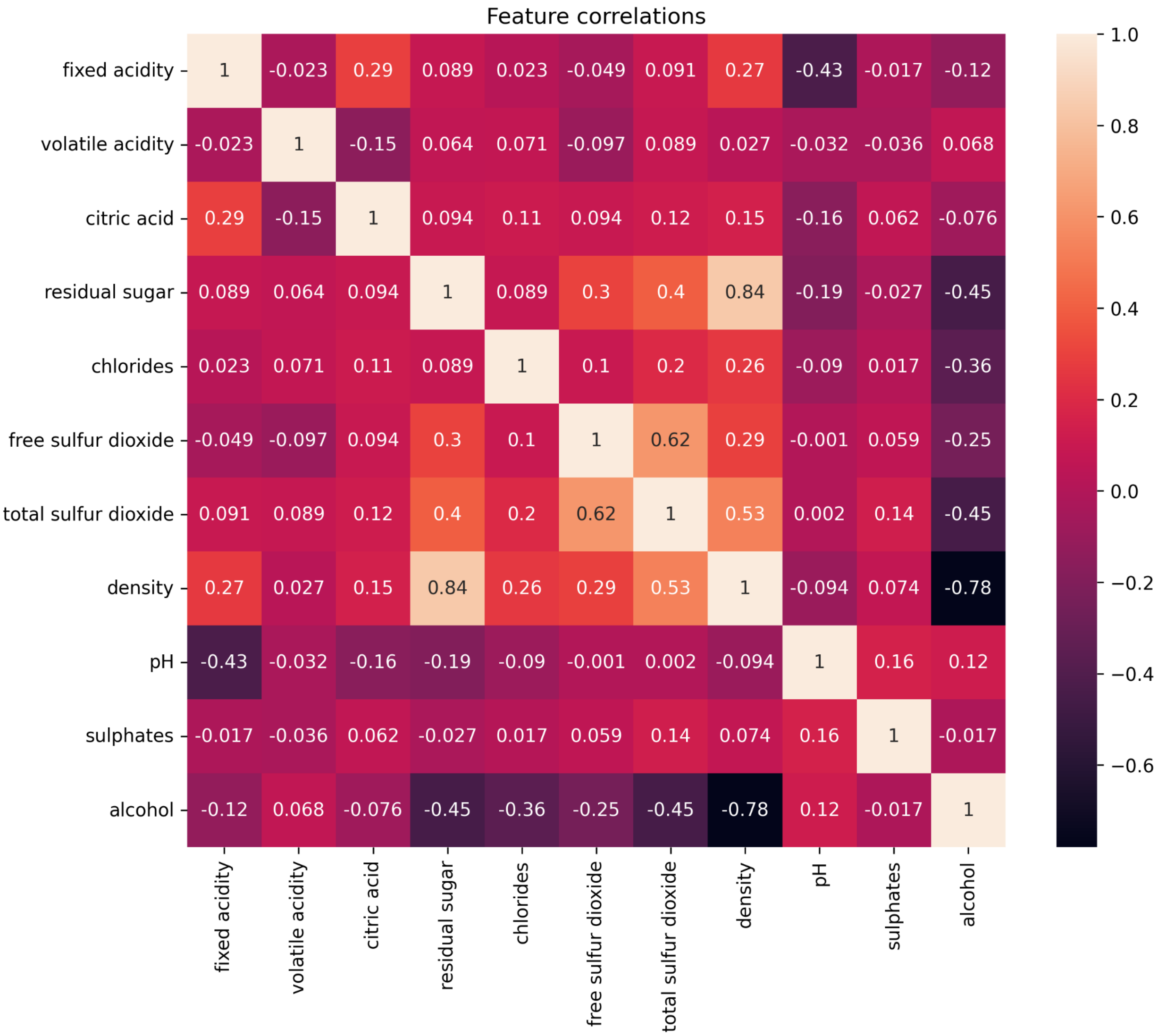

Figure 1. Correlations among input features shown by heatmap

Our target variable is an ordinal variable so it has both regression features as well as class distribution. The distribution of our target variable is normally distributed with a very small standard deviation and no skewness (figure 2) . A quality score of 3 and 9 are very rare in this distribution and they are three std away from the mean. The class distribution (figure 3) confirms that we have an imbalanced class distribution, score 6 is the majority class and 3,4,8,9 have very little presence as compared to 5, 6, 7.

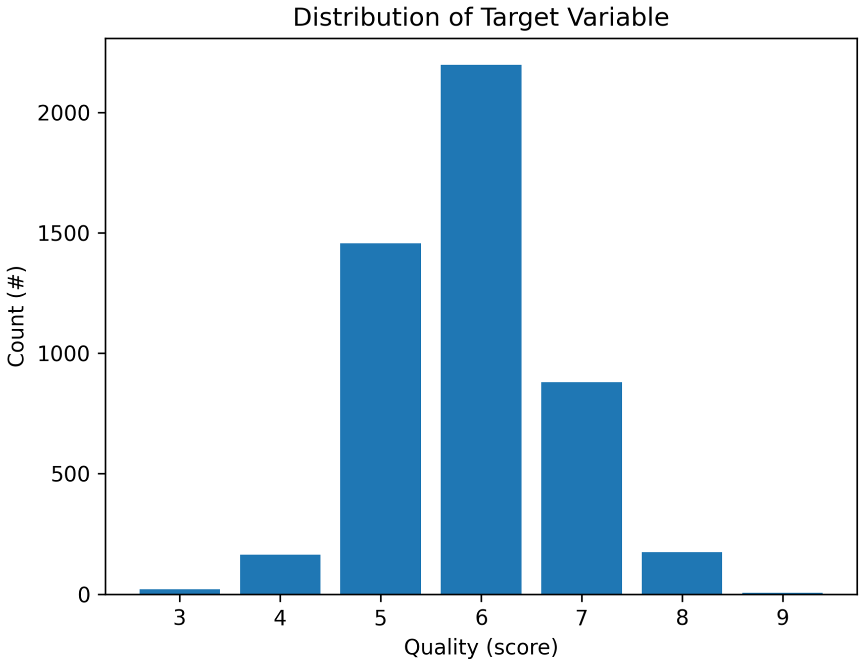

Figure 2. Target variable distributions shown by barplot

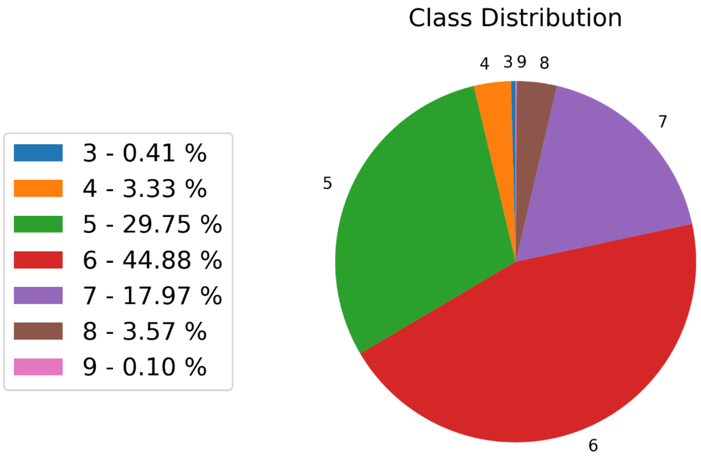

Figure 3. Target variable class distribution

Then I conduct F-statistics analysis to explore the correlation of my input features to the target variable. Figure 4 illustrates the top five most correlated features. After conducting F-statistics analysis, I found that the input feature alcohol has the largest correlation with the target variable. So I plot the distribution of input feature alcohol in each target variable class (figure 5). Surprisingly, alcohol also has a non-linear correlation with the target variable as well. We can see that the mean of alcohol tends to increase with quality score after quality score is greater than 5, but decrease with quality score when quality score is smaller than 5.

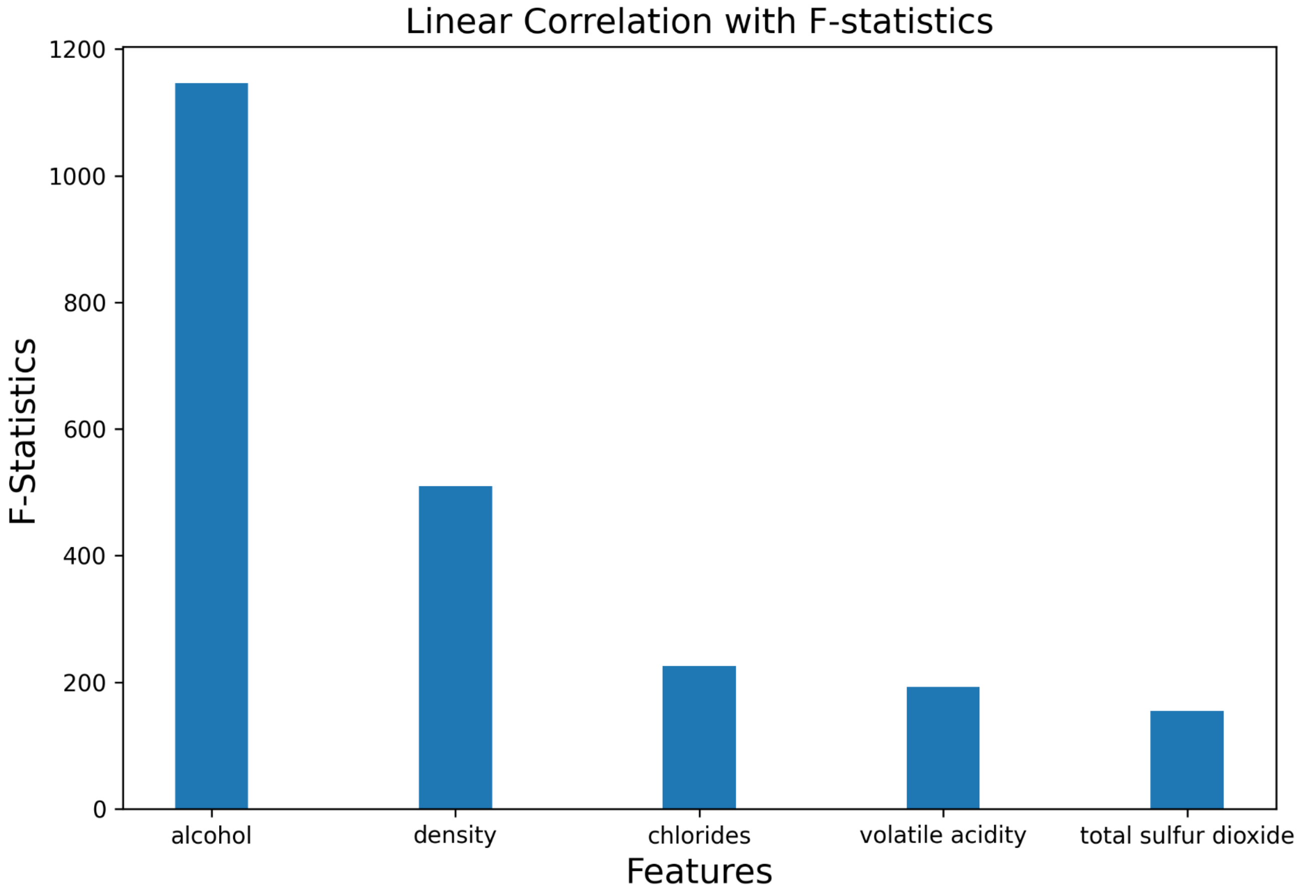

Figure 4. Most correlated input features with target variable

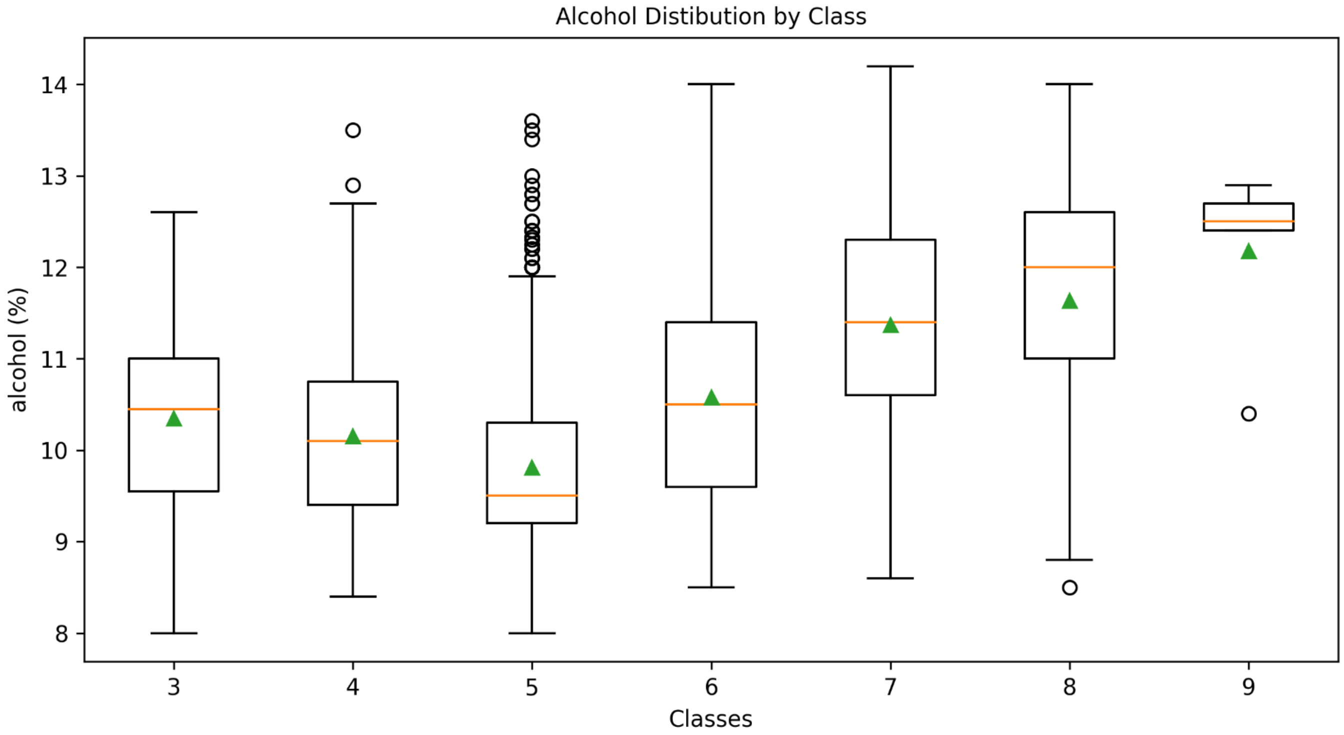

Figure 5. Distribution of feature alcohol by each target variable class

Methods

Splitting

My dataset is independent and identically distributed without group structure. So I will split my dataset into training, validation, and testing of a 60%-20%-20% ratio with stratification. The reason for this arrangement is that the class in the target variable is not balanced. For example, there are only 5 instances with target variable equals to 9. So I need the 60%-20%-20% ratio to ensure that there is at least 1 instance in all of training, validation, and testing with target variable equals to 9.

Preprocessing

I use MinMaxScaler for all my input features because they are all continuous variables. This choice is also motivated by the fact that I am dealing with some skewed distribution and also some features have very large mean and standard deviation. Using MinMaxScaler, I could have all input features in the range of (0,1). After processing, I have 11 features in the input data.

Pipeline

Then, I will select my models based on five different random splits of the data. In this way, I can account for the variance occurring in splitting my data. For each different split, the first step is to run a grid search on the splitted and preprocessed data to find the best hyperparameters for the model, using the training and validation dataset. For non-deterministic models like random forest, I will also run different initialization to account for model randomness. I will select my model from four algorithms: linear regression, support vector machine, random forest, k-nearest neighbor, and xgboost. This decision is motivated by the fact that these algorithms tend to perform well for other regression problems and they are frequently used and recommended in other papers on this topic as well.

Hyperparameter and Model Selection

The hyperparameters searched for different models are as follows. Then, for the best performing models during cross validation, I will run it on the test sets and compare the test score distribution with other models. Since in this problem our target variable, the quality score, is an ordinal variable, I trained and evaluated my models using a regression approach and the selected metric is RMSE. This choice of metrics is motivated by the fact that RMSE is one of the most general metrics for regression problems, and it is reported in many papers related to this dataset. After I trained and evaluated my models using the regression approach, I also evaluated them from a classification point of view. After the model outputs predicted scores for each instance, I can round them to the nearest integer and check if it matches the correct score label. I can then calculate the accuracy of my model by dividing the number of correctly predicted instances over the total number of instances.

Algorithm	**Hyperparameter**	**Value Searched**
Ridge	alpha	0.001, 0.01, 0.1, 1, 10
SVM	kernel	'linear', 'poly', 'rbf', 'sigmoid'
	degree	2, 3, 4
Random Forest	n_estimators	10, 100, 1000
	max_depth	1, 3, 10, 30, 100
	max_features	0.25, 0.5, 0.75, 1.0
KNN	n_neighbors	1, 5, 10, 20
	weights	'uniform', 'distance'
Xgboost	n_estimators	1, 3, 10, 100
	max_depth	10, 100, 500

Table 1. Hyperparameter tuned for each algorithm

Results

Baseline

The baseline RMSE of this problem is 0.886, which is from the result of using the mean of the target variable as my model prediction. The corresponding baseline accuracy is 0.449 when I round predictions to the nearest integer. This accuracy is consistent with the baseline if we always predict the most prevalent class.

Model Selection Result

The mean and standard deviation of the test scores of the best models from each split is illustrated in the table below. Random forest is the most predictive model for this problem. The optimal parameter choice of random forest is max_depth=100, max_features=0.25, n_estimators=1000, and other parameters are as default. It reached a RMSE of 0.607 and accuracy of 66.6% after rounding its predictions to the nearest integers.

Algorithm	Ridge	SVM	RF	KNN	Xgboost
RMSE Baseline	0.886				
RMSE Mean	0.745	0.701	0.617	0.645	0.671
RMSE Std	0.011	0.006	0.007	0.012	0.009
*Std over Baseline	12.2	27.8	38.3	19.9	25.0
ACC_Baseline	0.449				
ACC_Mean	0.518	0.562	0.660	0.648	0.635
ACC_Std	0.010	0.007	0.013	0.005	0.022

Table 2. Results of each algorithm

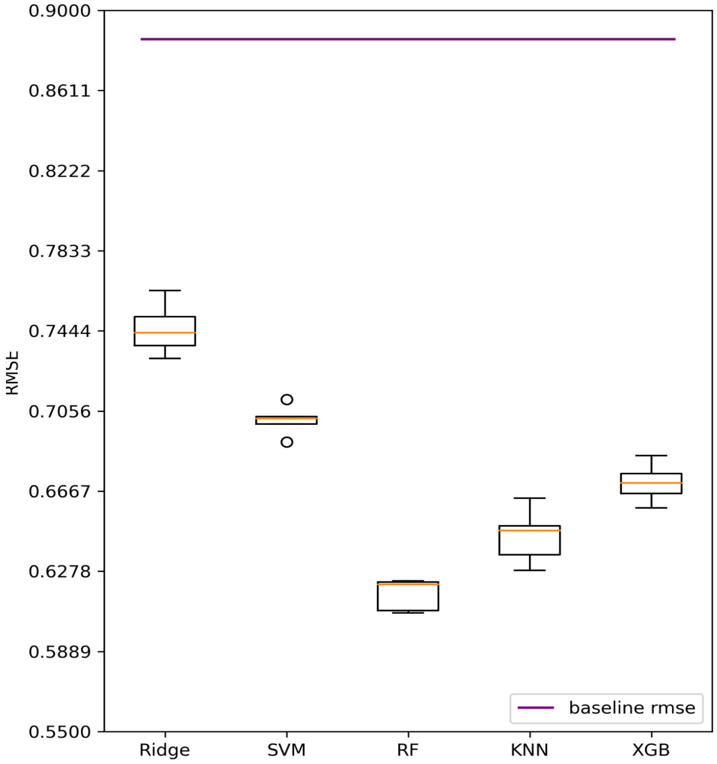

Figure 6. RMSE of the best models of each algorithm

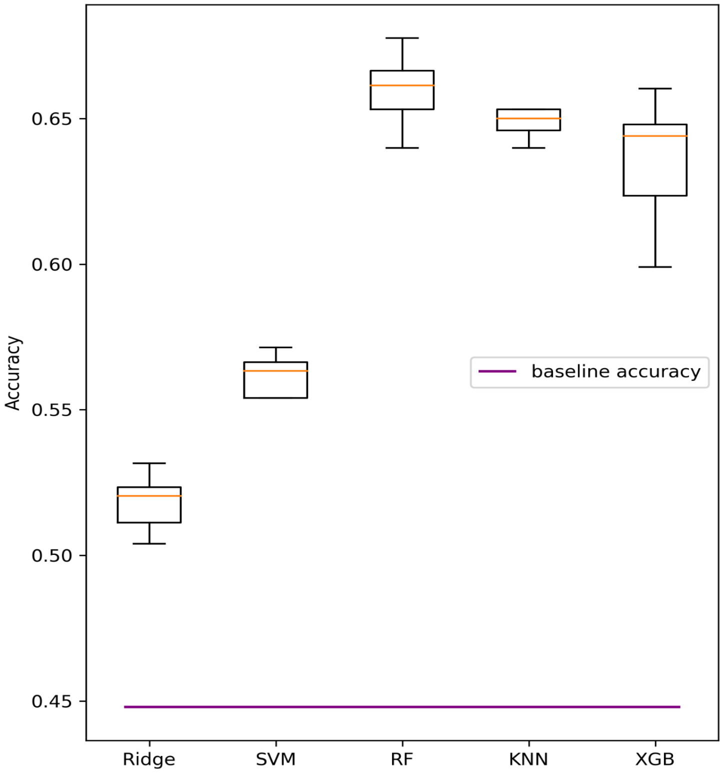

Figure 7. Accuracy scores of the best models of each algorithm

Feature Importance and Discussion

For this best model, I studied the global feature importance for all features by calculating their Shapely values, using permutation importance, and using random forest's impurity-based feature importance attribute. Their results are shown in figure 8, 9, 10, respectively. It turns out that alcohol percentage is always the most important feature. This finding is largely consistent with the EDA. Density, free sulfur dioxide, and volatile acidity are always in the top-5, although their order is different in three methods. Sulfates and fixed acidity are always the least two important features.

To study the local feature importance, I picked 10 individual points from the test set and created their force plots. These points include representatives from the ones closer to the mean like 5,6,7, and from the ones far away from the mean like 3,4,8,9. I selected some of the results in figure 11, 12, 13. It turns out that on points closer to the mean and the outliers with low quality score, feature alcohol usually negatively contributes to the prediction. But for outliers with high quality scores, feature alcohol usually positively contributes to the prediction. This also echoes my findings in EDA where I pointed out that feature alcohol has a non-linear correlation with the target variable. Another interesting finding is that for wines that have a very low quality score, a very low value of free sulfur dioxide is usually present. This suggests that this feature might have a correlation with wines that have low quality scores.

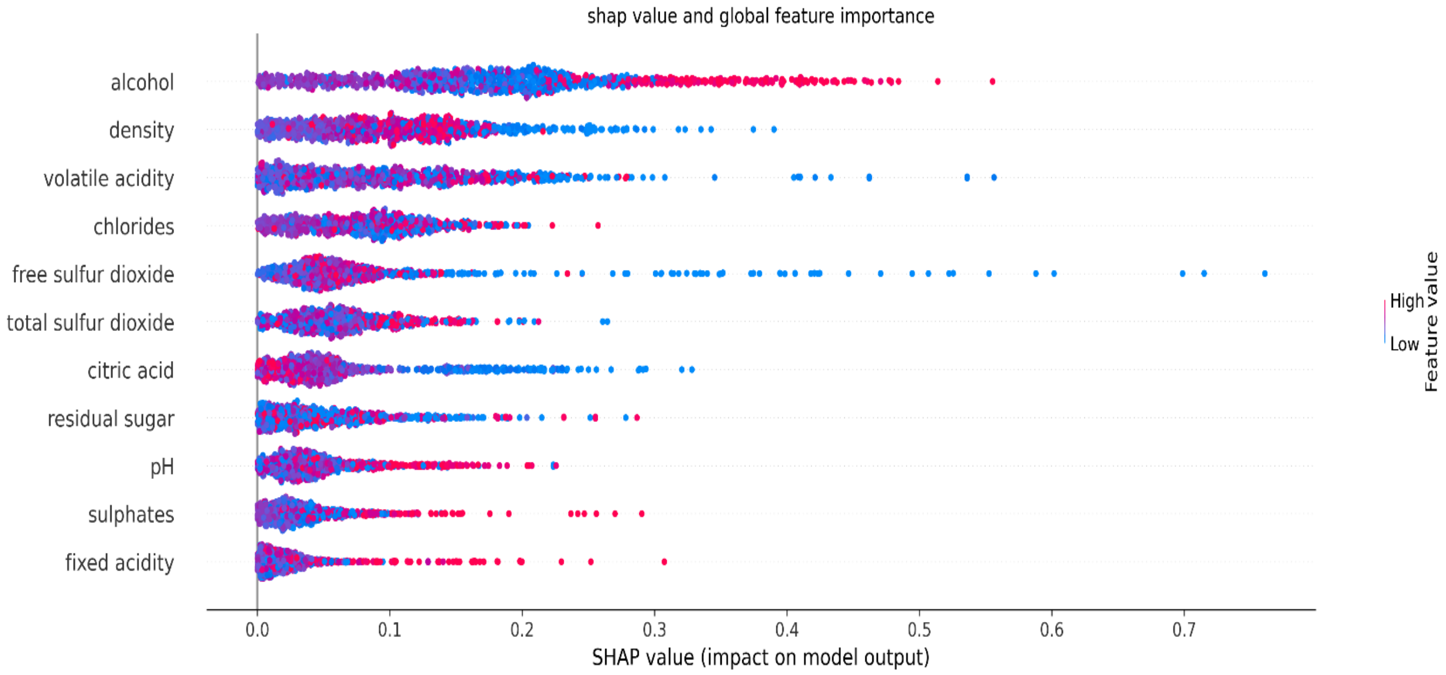

Fig 8. Shapley values distribution for each feature ranked by sum of absolute value

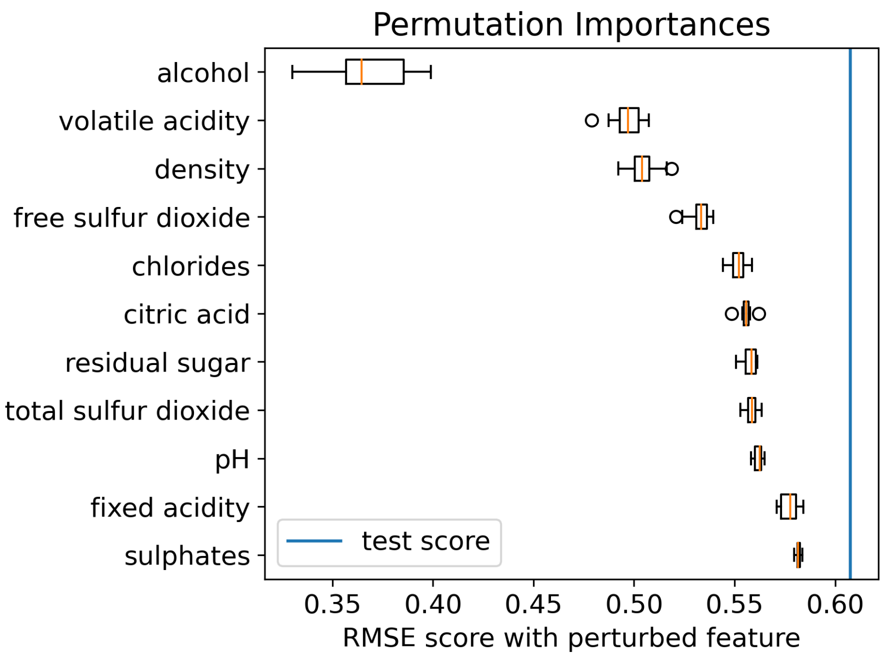

Figure 9. Test scores after apply permutation on each feature

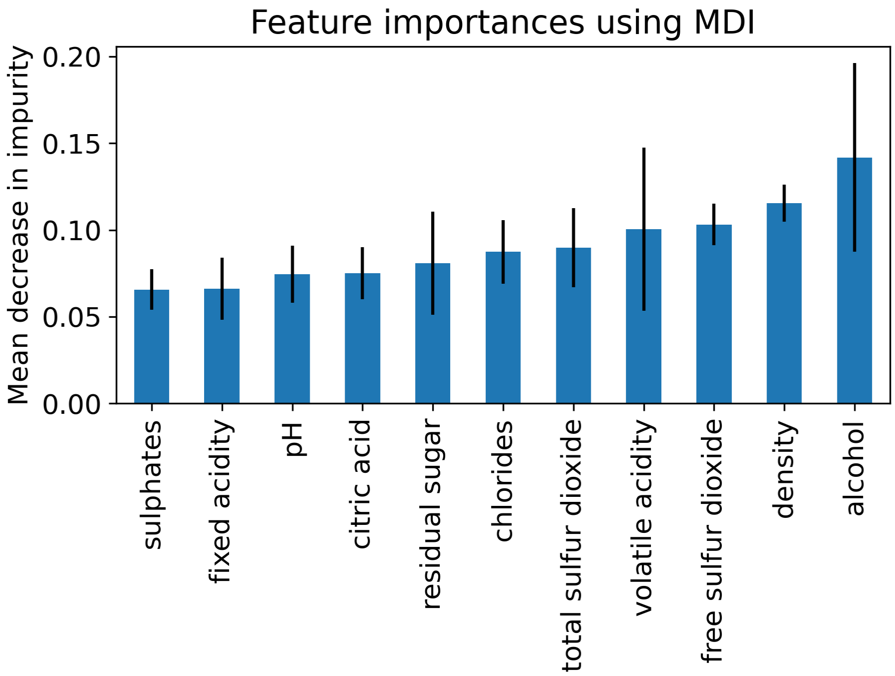

Figure 10. RF model attribute: feature importance_

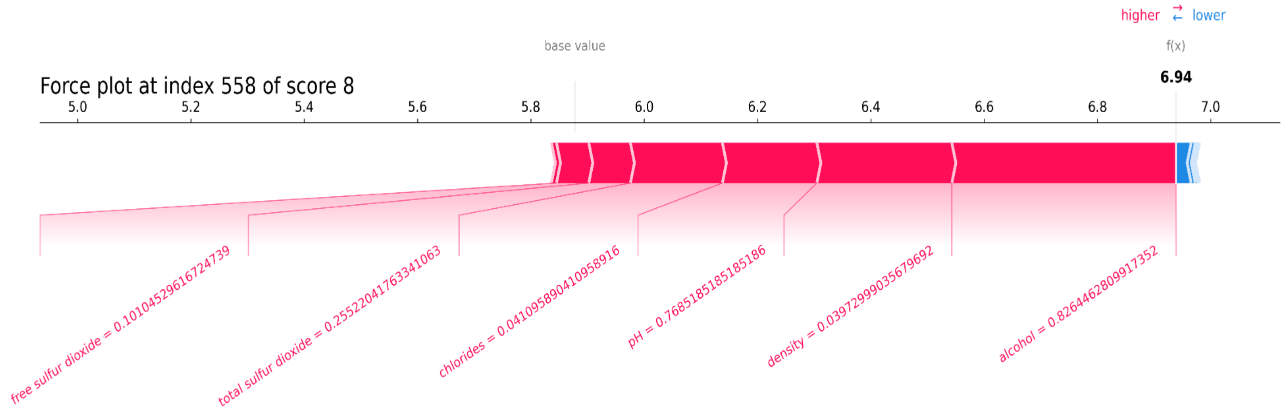

Figure 11. Feature contributions in predicting for an instance of score 8

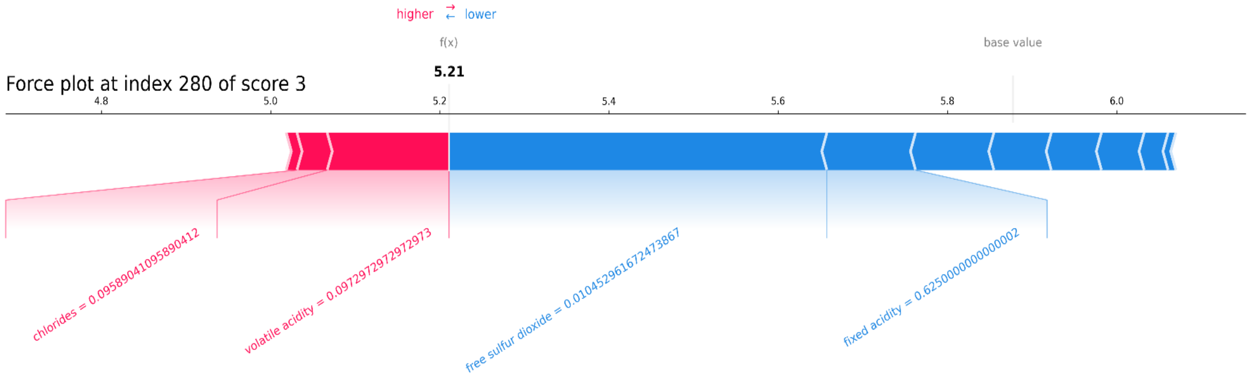

Figure 12. Feature contributions in predicting an instance of score 3

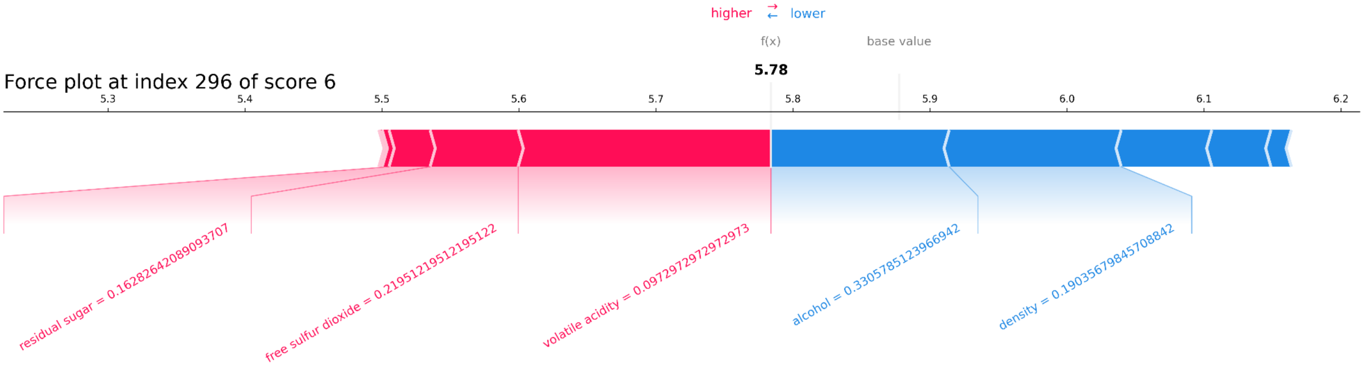

Figure 13. Feature contributions in predicting an instance of score 6

Outlook

My best model shows satisfying performance above the baseline and compared to some other research. However, it is not perfect especially for that it does not do well on predicting the outliers. Since the data is highly imbalanced, the model tends to make predictions closer to the mean of the target variable. I could have set higher sample weights to those points that have a label far away from the mean, but doing so does not usually improve the overall performance of the model. If I set the sample weight too large, then the model sacrifices its performance on the instances whose labels are closer to the mean.

If I could collect more data on the wines that have either very high quality scores, or very low quality scores to make the dataset more balanced, then I think my model would have a better performance. Also, my model could also make better predictions if I could get more features on the wine samples, for example other metrics like color, sediment, etc. The reason is that wine samples that score 5,6,7 consist of more than 80% of the total samples, but my model could not reach a perfect regression on these points, which means there is some randomness among these points if we only look at the currently available features.

Word count: 1911

References

1. Cortez, Paulo. “ Wine Quality Data Set.” UCI Machine Learning Repository: Wine Quality Data Set, 7 Oct. 2009, 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, 2009.

3. Koranga, Manisha, et al. "Analysis of white wine using machine learning algorithms." Materials Today: Proceedings 46 (2021): 11087-11093.

4. Gupta, Ujjawal, et al. "Wine quality analysis using machine learning algorithms." Micro-Electronics and Telecommunication Engineering. Springer, Singapore, 2020. 11-18.

5. Lee, Seunghan, Juyoung Park, and Kyungtae Kang. "Assessing wine quality using a decision tree." 2015 IEEE International Symposium on Systems Engineering (ISSE). IEEE, 2015.