Red Wine Quality Prediction

Github: https://github.com/tianqi-cheng/Wine-Prediction

1. Introduction

My ML problem is using the physicochemical properties of red wines to predict their quality. The target variable is quality. It is a regression problem.

According to a report (Conway), the US consumes the largest volume of wine in the world, at 33 million hectoliters in 2021. Therefore, exploring different properties of red wines and assessing which factors influence red wine quality the most would be important and interesting, especially in the US. I would take advantage of this dataset to predict red wine quality so that we do not have to rely on humans for assessing the quality of wine, which is time-consuming and expensive.

This dataset has 1599 data points and 11 features. All the features contain data of physicochemical properties of red wines, including fixed acidity (g/L), volatile acidity (g/L), citric acid (g/L), residual sugar (g/L), chlorides (g/L), free sulfur dioxide (mg/ L), total sulfur dioxide (mg/L), density (g/mL), pH, sulphates (g/L) and alcohol (vol%).

The dataset is from UCI Machine Learning Repository. The 11 features were recorded by a computerized system. The data of the target variable - quality consists of scores rated by human assessors. Each score ranges between 0 and 10. '0' means very bad wine and '10' means excellent wine.

I read several public projects and publications that used this dataset. The ML questions of the publication (Cortez et al.) and the project (Sivalenka) are both to predict human wine taste preferences. The authors of that publication found the SVM method achieved the best results among other methods they used, including multiple regression and neural network methods. Its overall accuracy is 62.4%. The author of the project found SVC and the Random Forest led to great prediction accuracies, and they are both 99%.

2. Exploratory Data Analysis

As I mentioned earlier, the target variable is quality, and it is a score between 0 to 10. However, we can see from this bar plot (Fig. 1) that those human assessors are neither very strict nor very generous. All of the scores are between 3 to 8. The majority of scores are between 5 to 7. In fact, they account for nearly 95% of all points. Therefore, the data is very imbalanced.

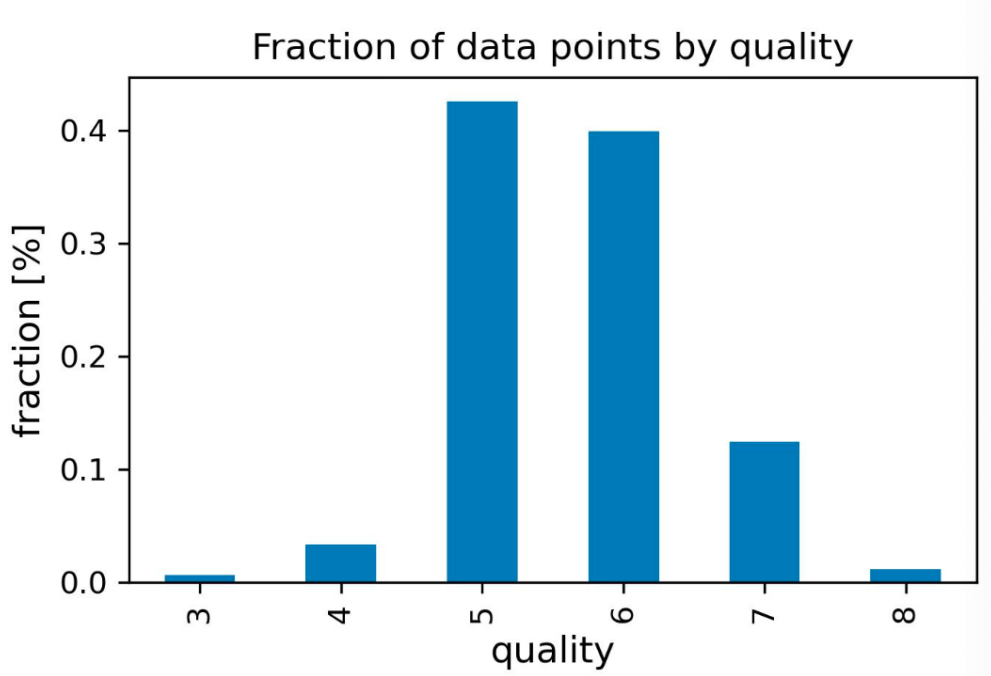

Fig. 1. The bar plot for the target variable - quality.

I explored what properties are very relevant to our target variable - quality. Surprisingly, alcohol has the strongest linear correlation with quality: the higher the alcohol content, the higher the score given by the evaluators. As this box plot (Fig. 2) shows, there is an upward trend.

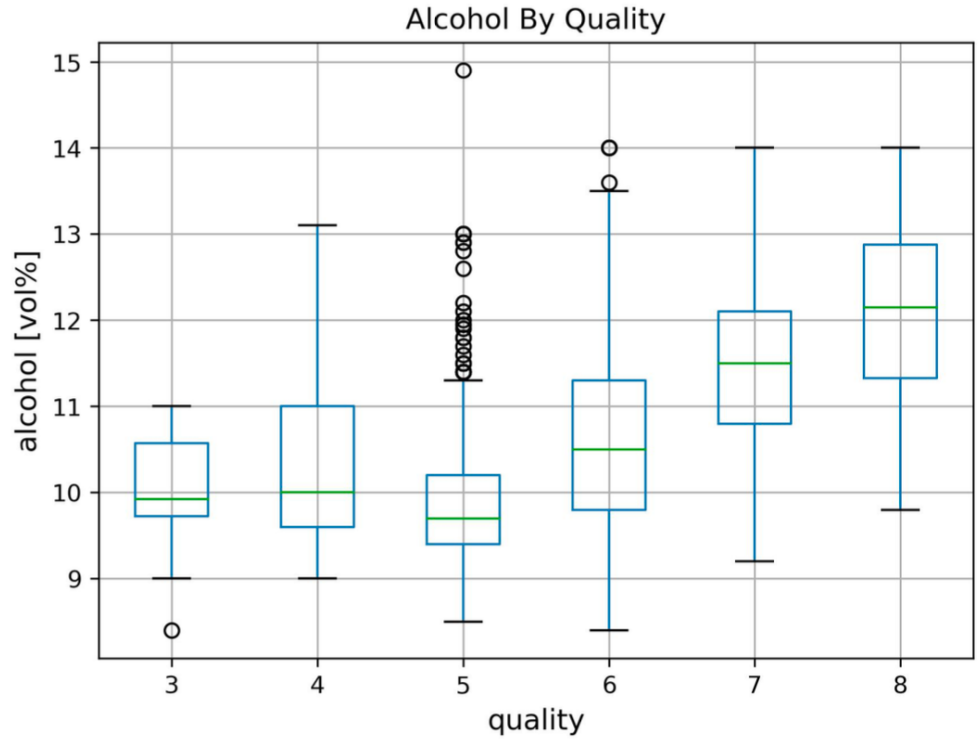

Fig. 2. The box plot for the quality and alcohol.

Volatile acidity is the second most linearly-correlated feature with the target variable. We can see their inverse relationship from this violin plot (Fig. 3).

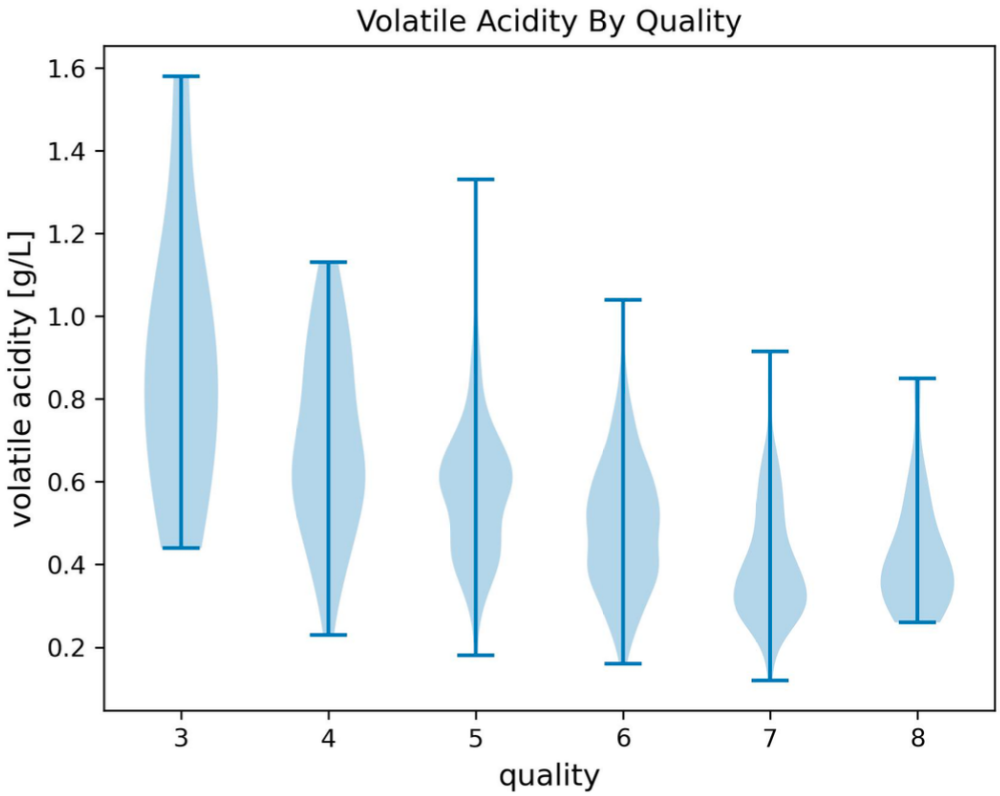

Fig. 3. The violin plot for the quality and volatile acidity.

This heatmap (Fig. 4) shows the Pearson correlation matrix of the features. If the color block is dark yellow, it stands for a highly positive correlation. If the color block is dark purple, it stands for a highly negative correlation. The correlations of several pairs of features are relatively high: fixed acidity and pH, fixed acidity and citric acid, fixed acidity and density, and free sulfur dioxide and total sulfur dioxide. But all the Pearson correlation coefficients are below 0.7, which means their correlations are not very high. Therefore, I kept all the features.

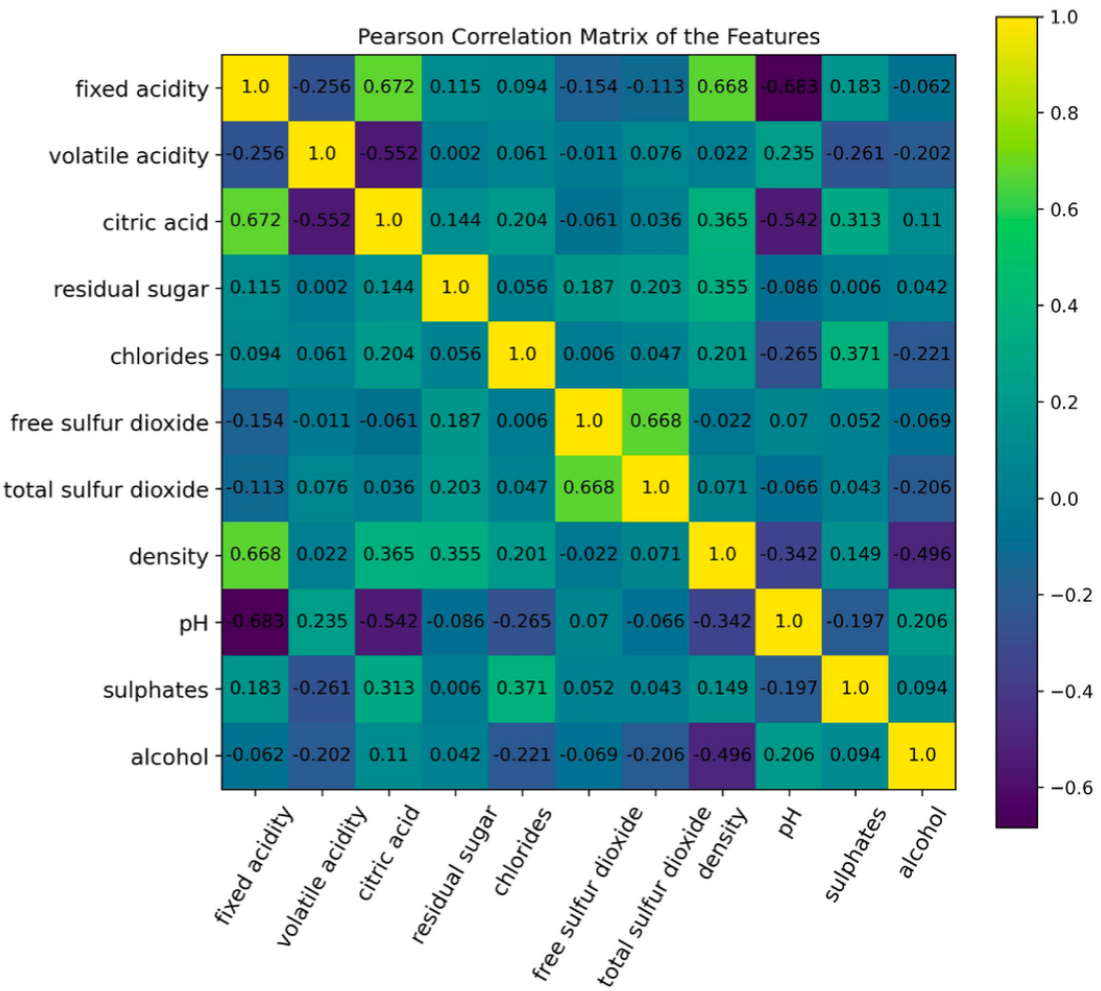

Fig. 4. The heatmap for the Pearson correlation matrix of the features.

This scatter plot (Fig. 5) shows a downward trend between fixed acidity and pH. The greater amount of fixed acidity a red wine has, the smaller the pH value it has.

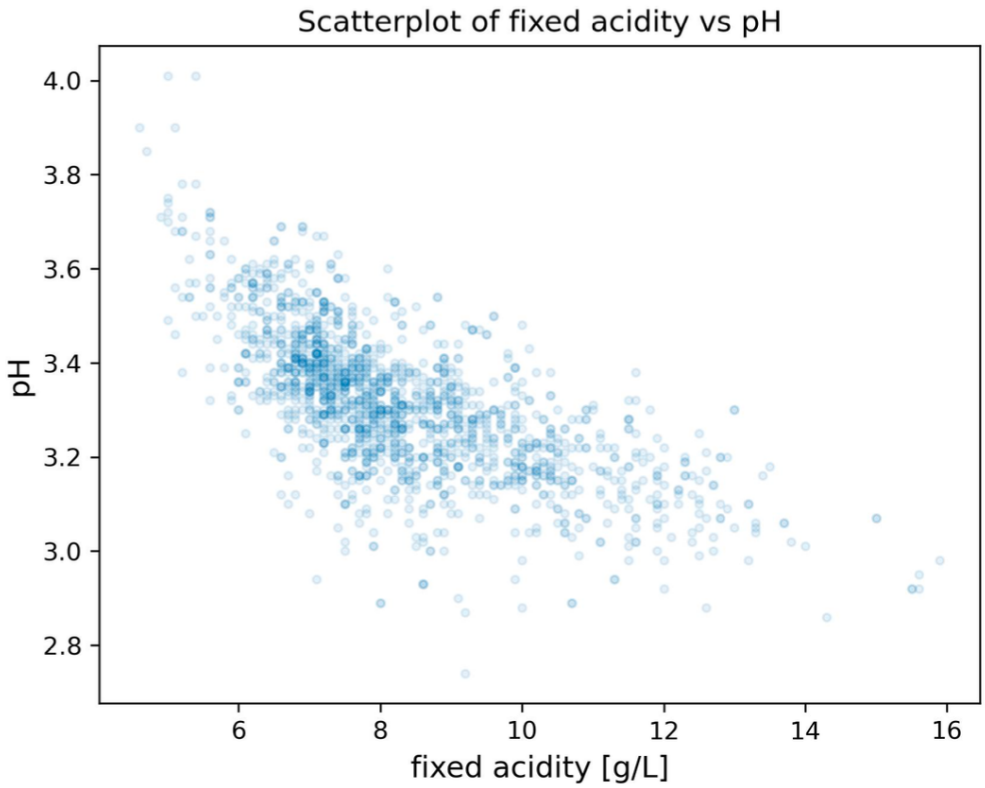

Fig. 5. The scatter plot for fixed acidity and pH.

3. Methods

The ML pipeline I built took the following aspects into consideration: splitting strategy, preprocessing method, evaluation metric, and ML algorithms and their parameter grids.

3.1 splitting strategy

This dataset is IID. It does not have a group structure and is not time-series data. As mentioned earlier, this data is imbalanced. So, firstly I split the data into other and test sets, and secondly, I used StratifiedKFold to split the other set. As the dataset is not very large, I followed the conventional practice: assigning 60% to the train set, 20% to the val set, and 20% to the test set. After the splitting, each subset has the same proportions of different labels as the original dataset.

3.2 preprocessing method

For preprocessing, I applied MinMaxScaler to pH and alcohol as their feature values are clearly bounded. As we all know, the pH value must be between 0 and 14. And alcohol content must be between 0% and 100%. For all other nine features, I applied StandardScaler, as they are all continuous features, but the boundaries of their values are not very clear.

This dataset has no missing value. So, I did not need to exclude points or features due to missing-value issues. Also, the number of features it has is just 11, which is obviously smaller than the number of points. Besides, the dataset in general is not very large, so training an ML algorithm

on it would not be very computationally expensive using all the features. So, I kept all the features and points in the preprocessed data.

3.3 evaluation metric

I used RMSE to evaluate the models' performance, as it is one of the most commonly used regression metrics and it has the same unit as the target variable. Therefore, it is easier for other people to understand this model's performance intuitively.

3.4 ML algorithms and the parameter grid

The ML algorithms I tried on this dataset were: linear regression with l1 regularization (Lasso), linear regression with l2 regularization (Ridge), linear regression with an elastic net (ElasticNet), Random Forest Regressor, XGB Regressor, SVR, and K Neighbors Regressor.

For the Lasso and the Ridge, the parameter I tuned for both models was alpha. I tried 60 values spaced evenly on a log scale from 1e-7 to 1e7.

For the ElasticNet model, the two parameters I tuned were alpha and l1 ratio. I tried 20 values spaced evenly on a log scale from 1e-4 to 1e4 for alpha, and I tried 9 values spaced evenly from 0.1 to 0.9 for l1 ratio.

For Random Forest Regressor model, the two parameters I tuned were max depth and max features. I tried 5 values (1, 3, 10, 30, 100) for max depth, and I tried 4 values (0.25, 0.5, 0.75, 1.0) for max features.

For XGB Regressor model, the parameter I tuned was reg_alpha, and I tried 6 values (0e0, 1e-2, 1e-1, 1e0, 1e1, 1e2).

For SVR model, the two parameters I tuned were C and gamma. I tried 3 values (1e-1, 1e0, 1e1) for C, and I tried 5 values (1e-3, 1e-1, 1e1, 1e3, 1e5) for gamma.

For K Neighbors Regressor model, the two parameters I tuned were n_neighbors and weights. I tried 8 values (1, 3, 5, 7, 10, 30, 50, 100) for n_neighbors, and I tried 2 values ('uniform', 'distance') for weights.

Summary of algorithms' parameter grids

ML algorithm	Parameter	Values tried
Lasso	alpha	60 values spaced evenly on a log scale from 1e−7 to 1e7
Ridge		
ElasticNet	alpha	20 values spaced evenly on a log scale from 1e−4 to 1e4
	l1 ratio	9 values spaced evenly from 0.1 to 0.9
RandomForestRegressor	max depth	5 values: (1, 3, 10, 30, 100)
	max features	4 values: (0.25, 0.5, 0.75, 1.0)
XGBRegressor	reg_alpha	6 values: (0e0, 1e−2, 1e−1, 1e0, 1e1, 1e2)
SVR	C	3 values: (1e−1, 1e0, 1e1)
	gamma	5 values: (1e−3, 1e−1, 1e1, 1e3, 1e5)
KNeighborsRegressor	n_neighbors	8 values: (1, 3, 5, 7, 10, 30, 50, 100)
	weights	2 values: ('uniform', 'distance')

Fig. 6. The summary of algorithms' parameter grids

For each model, I ran 10 times with different random states and recorded each time's best model and corresponding test score. Ultimately, I measured the uncertainties due to splitting and due to non-deterministic ML methods by calculating the standard deviation of the 10 test scores.

4. Results

For each random state, I calculated the baseline RMSE. The corresponding baseline prediction is the mean of values of target variables in each test set. And the baseline RMSE was derived by calculating RMSE using the baseline prediction and the ground truth.

RMSE is a metric that shows the average distance between the predicted values from the model and the actual values in the dataset. Therefore, the lower the RMSE, the better a given model is able to fit a dataset.

The RMSE of all the ML models I used are significantly lower than the baseline RMSE, which means those models are all effective to some extent.

Below is a table (Fig. 7) summarizing the performance of the ML models. As we can see, the most predictive ML model was the Random Forest Regressor model, which has the lowest mean of the 10 RMSE test scores, 0.5596. The standard deviation of these test scores is 0.022. The mean of the baseline test scores is 0.7863 and its std is 0.0281. Therefore, this model's RMSE is 8.0726 standard deviations below the baseline's RMSE.

Summary of the algorithms' results

ML algorithm	mean of baseline test scores	mean of test scores	std of baseline test scores	std of test scores
RandomForestRegressor	0.7863	0.5596	0.0281	0.0220
XGBRegressor		0.5687		0.0190
KNeighborsRegressor		0.5927		0.0185
SVR		0.6263		0.0206
Ridge		0.6406		0.0183
Lasso		0.6408		0.0186
ElasticNet		0.6409		0.0185

Fig. 7. The summary of the algorithms' results.

The best value for the Random Forest Regressor model's parameter max_depth is 30, and the best value for max_features is 0.5.

The global feature importance metrics I used were permutation feature importance, SHAP feature importance, and XGB total_gain metrics.

When using the permutation feature importance metric, the top 3 important features are alcohol, sulphates and volatile acidity, and the top 3 least important features are density, critic acid and chlorides.

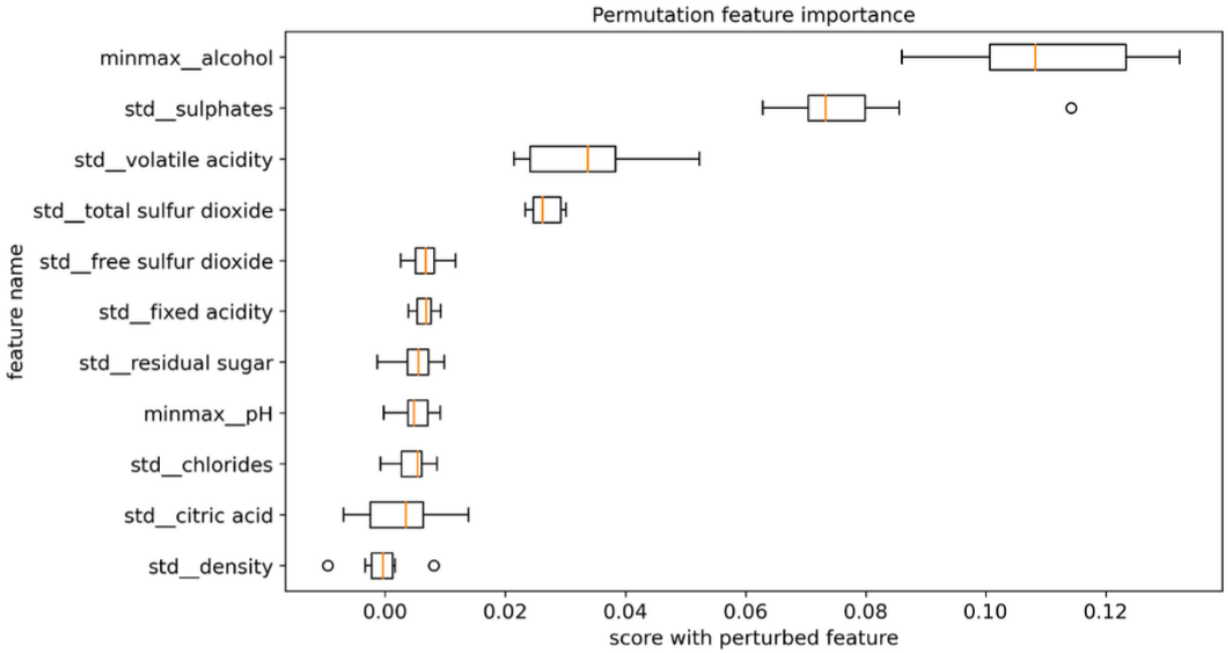

Fig. 8. Perturbation feature importance for the test set.

When using SHAP values, the top 3 important features are the same as those calculated with the first metric, but the top 3 least important features (residual sugar, free sulfur dioxide and fixed acidity) are different.

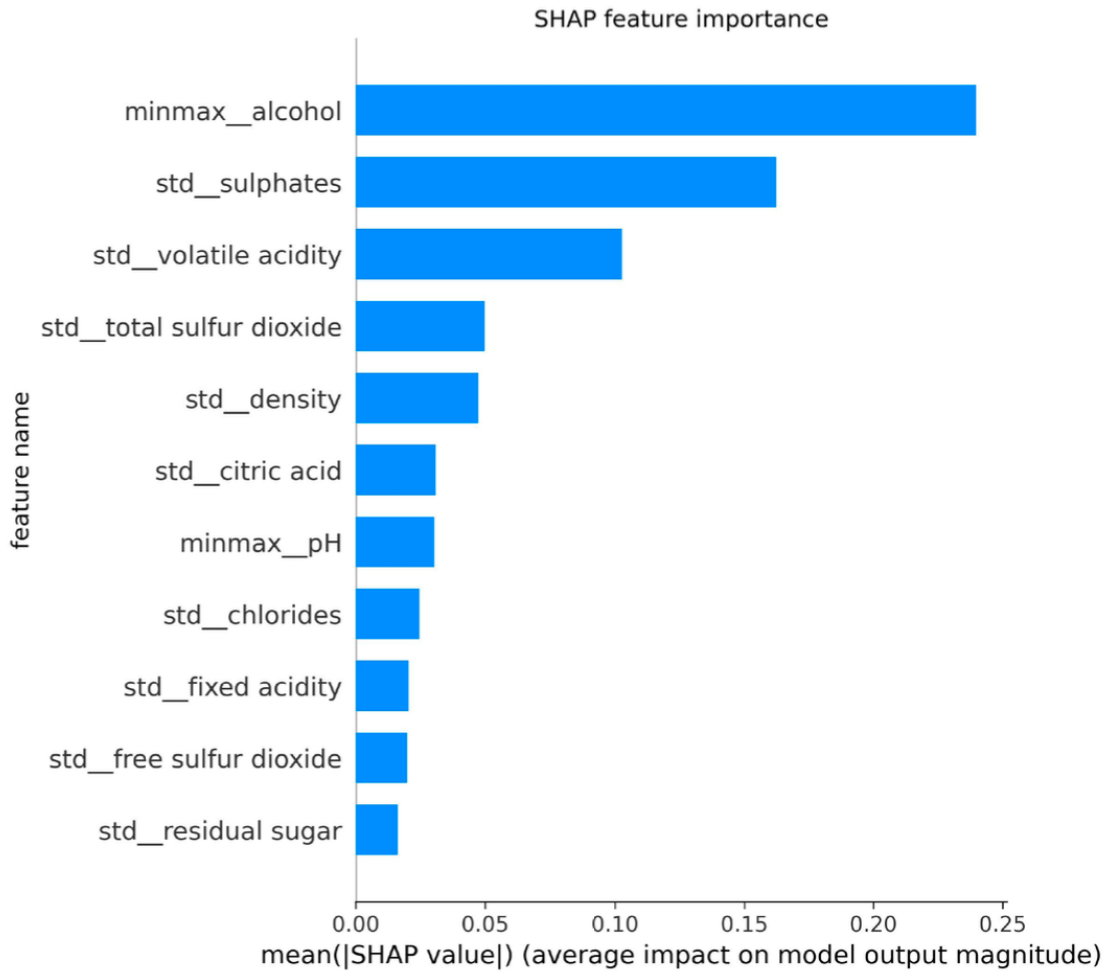

Fig. 9. SHAP feature importance for the test set.

As we can see in figure 7, XGB is the second best model with the mean of the test scores (0.5687) slightly worse than RandomForestRegressor's but the std (0.0190) slightly better.

When using XGB total_gain metric, the top 3 important features are also the same as those calculated with the first and second metrics, and the top 3 least important features are the same as those calculated with the second metric.

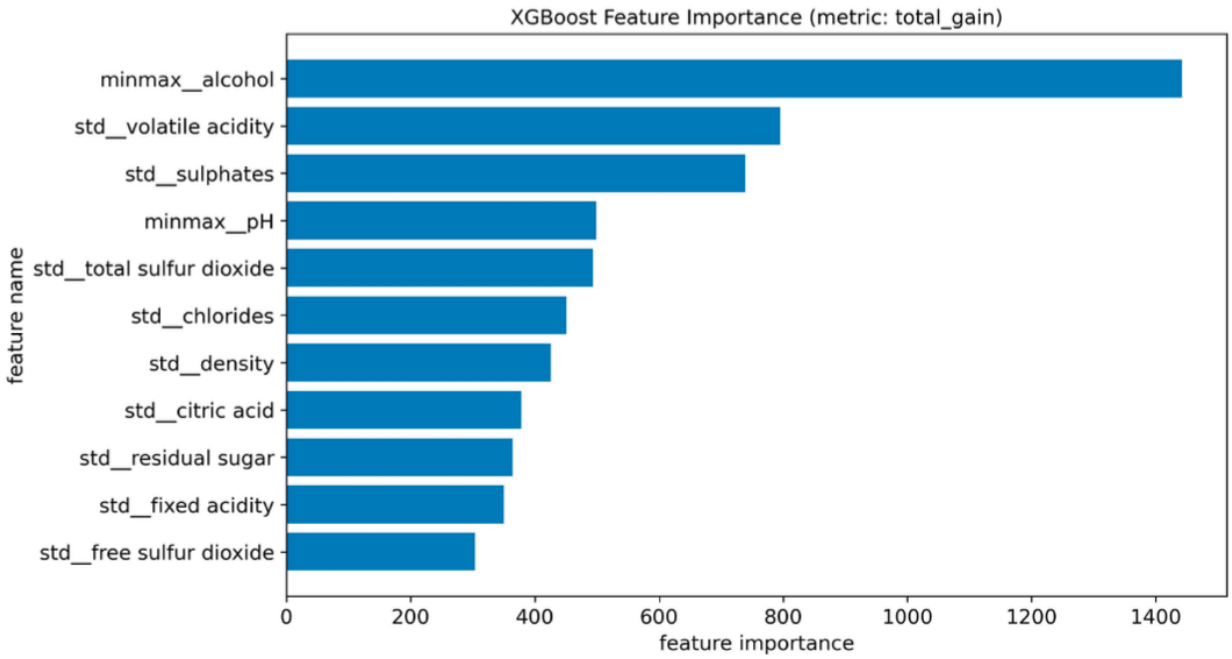

Fig. 10. XGBoost Feature Importance metric: total_gain.

To sum up, alcohol, sulphates and volatile acidity are the top 3 most important features. The top 3 least important features would vary with different global feature importance metrics, but residual sugar, free sulfur dioxide and fixed acidity appear more frequently in the top 3 least important features list. Based on this conclusion, wine producers can assign more resources to these most important factors when producing red wines. Similarly, wine lovers can pay more attention to these features when selecting wines.

Summary of three global feature importance metrics

Global feature importance metric	Top 3 most important features			Top 3 least important features		
Permutation	alcohol	sulphates	volatile acidity	density	critic acid	chlorides
SHAP	alcohol	sulphates	volatile acidity	residual sugar	free sulfur dioxide	fixed acidity
total_gain	alcohol	volatile acidity	sulphates	free sulfur dioxide	fixed acidity	residual sugar

Fig. 11. The summary of three global feature importance metrics.

The fact that residual sugar is one of the least important features is very unexpected. I think human beings are very sensitive to sweetness. Therefore, I expected that residual sugar played an important role when assessing the red wine's quality, whether its effect is positive or negative.

I also calculate SHAP values to inspect local feature importance for three data points. Take No. 125 datapoint as an example, the red arrows in its force plot represent feature effects (SHAP values) that drive the prediction value higher, while blue arrows are those effects that drive the prediction value lower. Each arrow's size represents the magnitude of the corresponding feature's effect. Therefore, the top 3 features that contribute positively to the prediction are alcohol, volatile acidity and sulphates. The top 3 features that contribute negatively to the

prediction are fixed acidity, chlorides, and pH. It is generally consistent with the results calculated with the aforementioned global feature importance metrics.

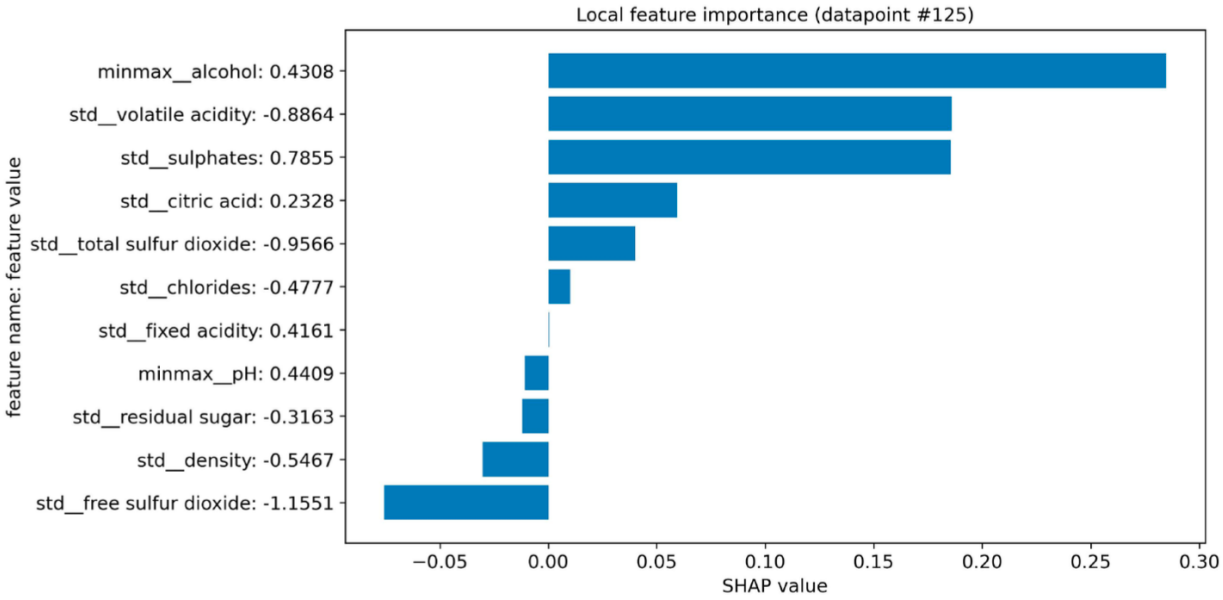

Fig. 12. Local feature importance based on SHAP values for datapoint No. 125.

5. Outlook

There are some ways that might further improve the model's performance.

To begin with, if I have stronger computational resources, I could search over more values when tuning the models' hyperparameters.

Also, I tried seven different ML algorithms on this dataset, but I could try more algorithms given more time, since maybe these other models work even better than the current best model.

Besides, as this data is imbalanced and it is in general not very large, maybe I could upsample classes with smaller points to balance the dataset.

In addition, the number of features now is 11, I could add more features through feature engineering, which may improve the predictive power.

References

1. [1] Conway, Jan. "Wine consumption worldwide in 2021, by country (in million hectoliters)." statista, July 2022, URL:https://www.statista.com/statistics/858743/ global- wine-consumption-by-country/

2. [2] Cortez, Paulo., et al. "Modeling wine preferences by data mining from physicochemical properties." Decision Support Systems, vol. 47, no. 4, November 2009, pp. 547-553, URL:https://www.sciencedirect.com/science/article/pii/ S0167923609001377?via%3Dihub#aep-section-id20
3. [3] Sivalenka, Madhuri. "Basic Machine Learning with Red Wine Quality data." Kaggle, URL:https://www.kaggle.com/code/madhurisivalenka/basic-machine- learning-with-red- wine-quality-data