Wine Quality

Michelle Motley

Data1030

https://github.com/numbersix1103/Final.git

Dec. 9th 2022

Introduction:

The purpose of this project is to build a model, for myself and others who wish to use or replicate this model, with the intention of predicting the quality of a wine based on physicochemical tests. The dataset I have chosen to work with for this midterm is from Kaggle. The dataset, "winequality-red', refers to a CSV file composed of data pertaining to various instances of "vihno verde," a unique brand of wine from Mihno, Northeast Portugal. The data was compiled and donated by a team of researchers (Cortez et al. Nov 2009) who wished to provide a dataset that could be used by others to model human wine taste preferences by way of physicochemical properties of the wines. However, this team only provided the compiled dataset, they did not run models on the dataset themselves. I find this dataset interesting to work with because I am curious to see how well a model will perform when it comes to predicting the quality of a wine given various physicochemical properties of the wines.

This dataset has 12 features and 1599 data points. The target variable I have chosen to model and eventually predict is the feature "quality". Quality refers to the number of points the wine is rated on a scale of 0-10. The "quality" feature is an integer, therefore, this is a regression problem as the model will predict a continuous quantity.

The other 11 features include: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. Reference Table 1 for further information pertaining to these features.

Exploratory Data Analysis:

After reading in my dataset, I checked the number of rows and columns. Then I checked if there were any null values and the data types of each feature as well as creating histograms of features and their counts. The following findings are generated from the EDA process. Referencing Figure 1, the histogram of target variable (quality) versus the count appears to be normally distributed with no apparent outliers. In regard to the target variable, values only vary over one magnitude. Figure 2, the CDF of the distribution of points, is meant to reinforce the previous claims. Referencing Figure 3, this figure is a scatterplot of residual sugar (g/L) on the x-axis and the distribution of quality of a wine on y-axis. Referencing Figure 4, using the median value of the feature 'alcohol', I created two categories. One where the alcohol content of wine is less than or equal to 10.422983% and the other where the alcohol content of wine is greater than 10.422983%.

Methods:

My ML pipeline takes in the following arguments: X, y, preprocessor, model, and param_grid. My ML pipeline consists of separating out the feature matrix and the target variable. The X and y are the feature matrix and target variable. The dataset I have selected is iid, meaning the next observation is independent from the previous one. There are no overall trends, the distribution doesn't fluctuate and all wines in the sample are taken from the same probability distribution. Furthermore, knowledge of the value of one variable gives no information about the

value of the other and vice versa. Given that my dataset is iid and fairly large (>1.5k data points), I chose to perform a basic split on my dataset, implementing train_test_split from Sklearn. I set shuffle = True in order to ensure that data was mixed prior to splitting. There was no need to stratify as my target variable appears to be normally distributed. I performed a basic split where the weights of each set followed as such: train = 0.6, test = 0.4. When preprocessing I used StandardScalar on all continuous variables. As mentioned previously, my dataset had no null values. Therefore, I did not need to handle unknowns during the preprocessing. After preprocessing, the result was 11 total features. The ML algorithms I use include: lasso (linear), ridge (linear), elastic net (linear), random forest (non linear), SVR (non linear) and KNeighborRegressor (non linear). Each algorithm has their own parameter grids. I calculate the baseline using the test set and the RMSE metric. Root mean square error is used to measure the differences between values predicted by a model and the values observed. I use RMSE so I can interpret the error in the same units as my target variable. Then I implement GridSearchCV, which searches exhaustively through the manually specified set of the hyperparameters for each algorithm in order to return the mean and standard deviation of each algorithm's best models across the 10 random state iterations. Outside of this ML pipeline, I created another pipeline specifically for XGBoost. I iterated through 10 random states, in which I performed a basic split where the weights of each set followed as such: train = 0.6, validation = 0.2 and test = 0.2. I did not use the same splitting as the first pipeline because the early stopping of XGBoost was not working with GridSearchCV. However, I used the same preprocessor. I calculated the baseline with the test set using the RMSE metric. I return the mean and standard deviation of each 10 random state's combination of hyperparameters. Uncertainty is introduced in my ML pipeline due to data splitting and non deterministic ML methods (reference Table 2 and Figure 5). Uncertainty in the pipeline can occur with the computer itself. Each computer system has its own method for generating "randomness." As well, each random state splits the dataset differently. Therefore, it is important to loop over the same random states in order to ensure reproducibility. Different splitting could result in different test points in each test set therefore producing different feature importances. For random forest, the randomness is due to the selection of random subsets of the features to learn on. Furthermore, I provided a graph (Reference Figure 6) displaying the true versus predicted values. The true values are those of my test set (y_test) and the predicted values are those of XGBoost's best model's prediction (chosen_pred).

Results:

Reference Table 2 for a table pertaining to the algorithms used and their respective baseline and mean RMSE scores. Every one of the algorithms performs better than the baseline. However, XGBoost performed exceptionally well in comparison to the other algorithms. Since XGBoost was calculated using a different ML pipeline (different test points/test set size) than the other algorithms, it has a different baseline score of ~0.853 and a RMSE score of ~0.557, a score that is nearly 9 standard deviations better. This resulted from random state 8 using the following hyperparameters: learning_rate = 0.0001, max_depth = 3, and n_estimators = 10. I selected this

algorithm's random state, with the combination of hyperparameters that produced the best RMSE score, to produce the following global feature importances: cover (Figure 7), gain (Figure 8), total cover (Figure 9), total gain (Figure 10) and weight (Figure 11). The five global feature importance metrics can be summarized in Table 3. The top 3 features regardless of the metric used, include alcohol, sulphates, and volatile acidity. As well, I used it to produce 3 local SHAP force plots (Figures 12-14). The expected value for the local SHAP is 5.6376586. Some important features that positively or negatively influence local SHAP values include sulphates and volatile acidity which can act as an antimicrobial/antioxidant or lead to an "unpleasant" "vinegary" taste, respectively.

Outlook:

One approach I would take to improve this model is to increase my number of instances; a larger data set could prove to be more useful for prediction than this smaller dataset. As well, I would ensure that there are instances that actually receive a 0 or 10 in terms of quality, the target variable. Another small improvement would be to min-max encode certain features that may potentially have a clear minimum and maximum, such as pH. Finally, the last improvement, time and memory willing, would be to tune more parameters. For example, in XGBoost, it may be beneficial to tune subsample or colsample_bytree.

Feature Name	Description	Units	Type	Notes
fixed acidity	refers to most acids involved with wine or fixed or nonvolatile (do not evaporate easily)	pH	float64/continuous	
volatile acidity	refers to the amount of acetic acid in wine	g/L	float64/continuous	at too high of levels can lead to an unpleasant, vinegar taste
citric acid	refers to the colorless weak organic acid that occurs naturally in citrus fruits	pH	float64/continuous	found in small quantities, and can add 'freshness' and 'flavor' to wines
residual sugar	refers to the amount of sugar remaining after fermentation stops	g/L	float64/continuous	it's rare to find wines with less than 1 g/L, and wines with greater than 45 g/L are considered sweet
chlorides	refers the amount of salt in the wine	mmol/L	float64/continuous	
free sulfur dioxide	refers to the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion	ppm	float64/continuous	it prevents microbial growth and the oxidation of wine
total sulfur dioxide	refers to the ratio of free and bound forms of SO2	ppm	float64/continuous	in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine
density	refers to the density of water depending on the percent alcohol and sugar content	g/cm3	float64/continuous	
pH	describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic)	pH	float64/continuous	most wines are between 3-4 on the pH scale
sulphates	refers to a wine additive which can contribute to sulfur dioxide gas (SO2) levels	ppm	float64/continuous	acts as an antimicrobial and antioxidant
alcohol	refers to the percent alcohol content of the wine	%	float64/continuous	

Table 1) Each feature's (excluding the target variable): name, description, units and notes (if applicable)

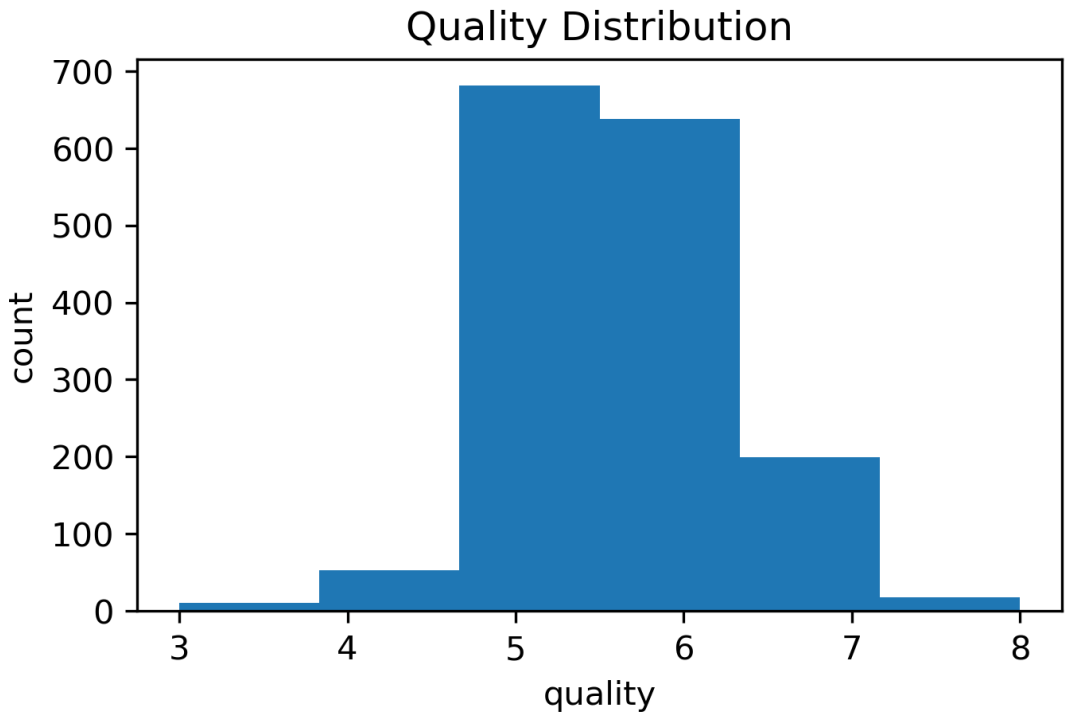

Figure 1) The target variable (quality) vs count. This figure is a histogram displaying points on the x-axis and the count of wines with that given rating on the y-axis. The figure appears to be normally distributed. Additionally, there are no apparent outliers.

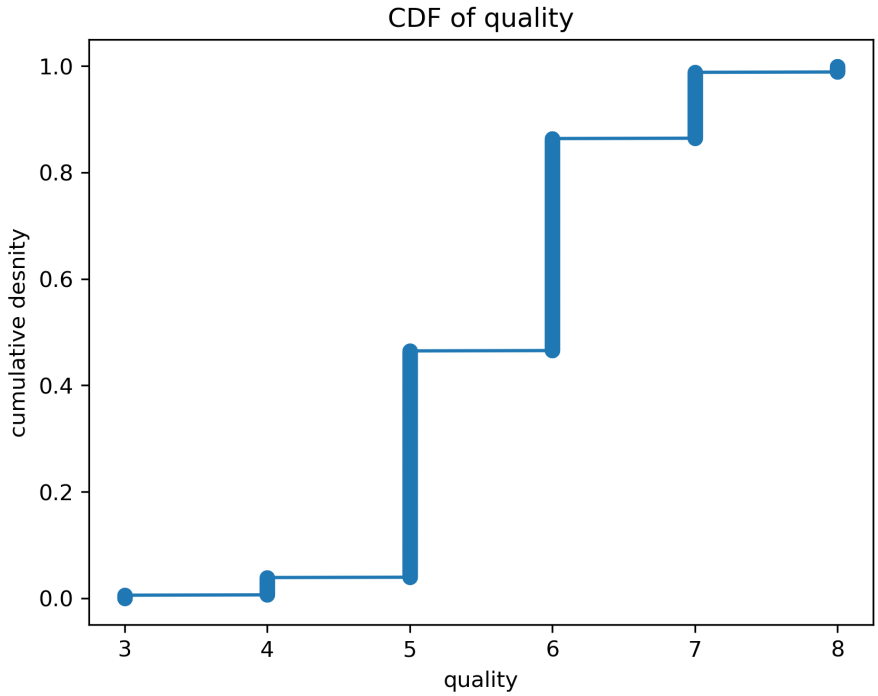

Figure 2) The CDF of the histogram displayed in Figure 1. This figure helps establish that the target variable (quality) is normally distributed. At a cumulative distribution of 0.5 (median/50th percentile), ~half of the data is accounted for (standard S curve).

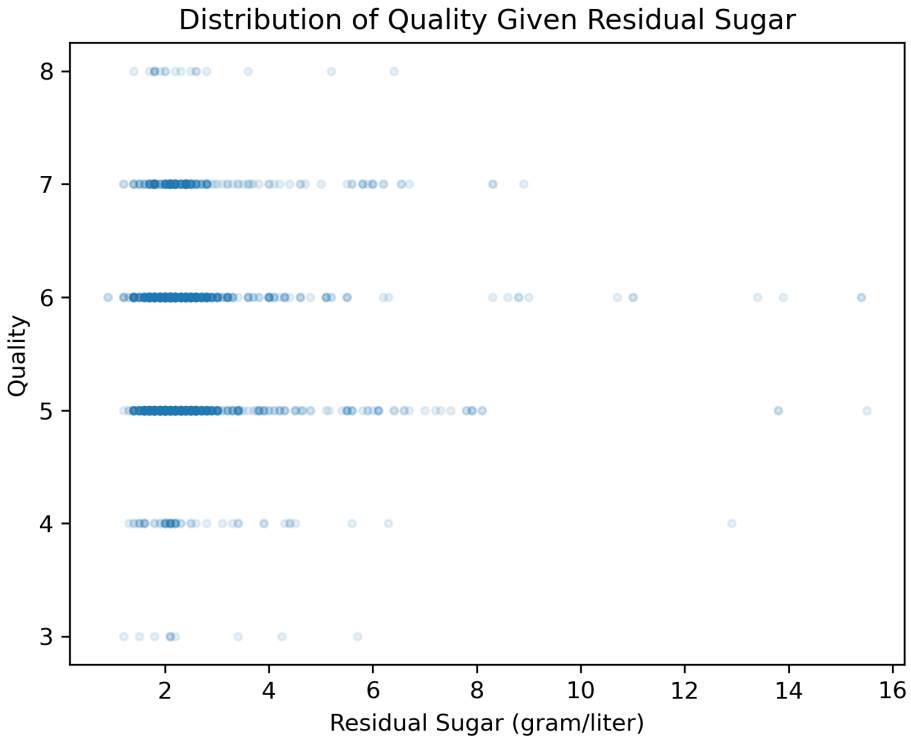

Figure 3: Residual sugar (continuous) vs Quality (continuous). This scatterplot displays residual sugar, on the x-axis, and the target variable, quality, on the y-axis. It is clear that the vast majority of wines have less than 4 g/L residual sugar.

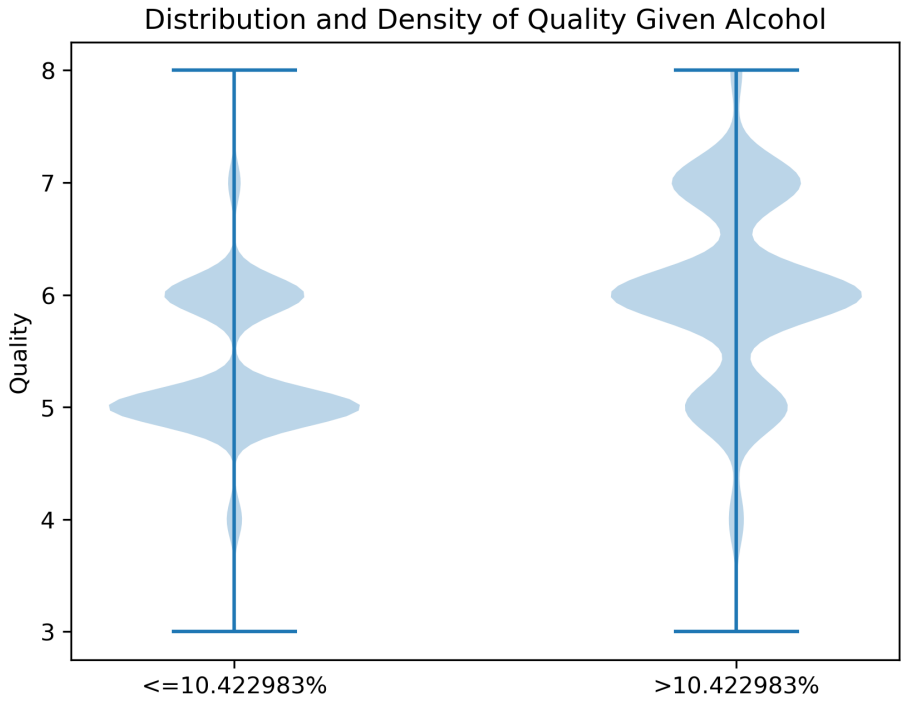

Figure 4: This violin plot depicts not only the distribution of data (summary statistics) but the density of wines with certain quality (peaks in the data). Wines that have greater than 10.422983% alcohol not only have a larger spread of quality, but the bulk of the density is rated higher than the bulk of the density of those wines that cost less than or equal to 10.422983%.

Algorithm	(Non) Linear	Baseline Score (RMSE)	RMSE Score	STD	Deviations from Baseline	Parameters Tuned
XGBoost	Non Linear	0.8533024083	0.55727	0.032991	8.973126255	Learning Rate = [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3] Max Depth = [1, 3, 10, 15] N Estimators = [10, 100, 300]
Random Forest	Non Linear	0.7930807591	0.6000968245	0.01230541373	15.68284812	Max Depth: [1, 2, 5, 7, 10] Max Features: [0.5, 0.75, 1.0]
SVR	Non Linear	0.7930807591	0.6290438361	0.013207026	12.42042857	Gamma: [1e-3, 1e-1, 1e1, 1e3, 1e5] C: [1e-1, 1e0, 1e1]
Ridge	Linear	0.7930807591	0.6418998226	0.01014589387	14.90070155	Alpha: np.logspace(-10, 0, 51)
Elastic Net	Linear	0.7930807591	0.6420801762	0.009995537998	15.10679894	Alpha: np.logspace(-2, 2, 21) L1 Ratio: np.linspace(0, 1, 21)[1:-1]
Lasso	Linear	0.7930807591	0.6422057623	0.009914002482	15.21837392	Alpha: np.logspace(-7, 0, 29)
KNeighborRegressor	Non Linear	0.7930807591	0.6646339121	0.01599046563	8.032714622	N Neighbors: np.arange(1,11)

Table 2) This table contains the algorithms used, whether they are linear or non linear, the baseline RMSE score, the standard deviation, how many standard deviations the RMSE score is from the baseline score and the parameters that were tuned.

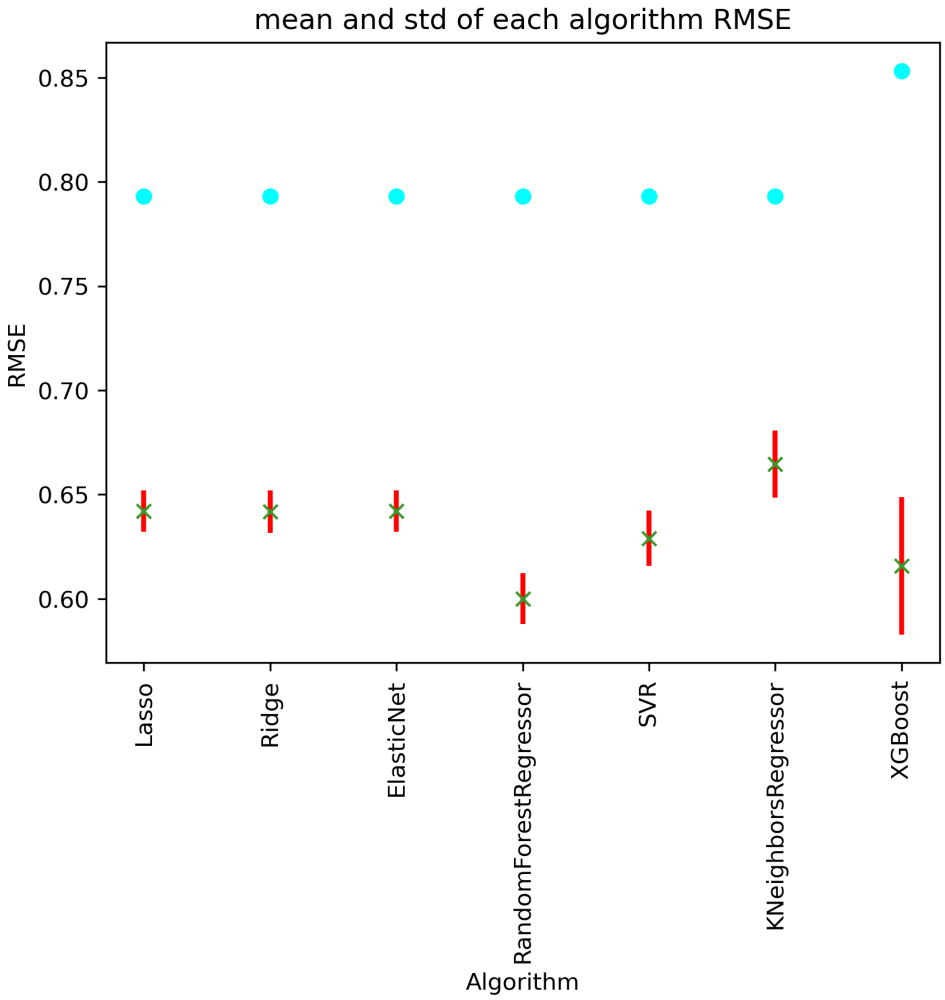

Figure 5) A plot to show different test scores under different random states

Figure 6) On the x-axis are the true values (y_test), on the y-axis are the predicted values (chosen_pred), where chosen_pred is the best XGBoost model's prediction

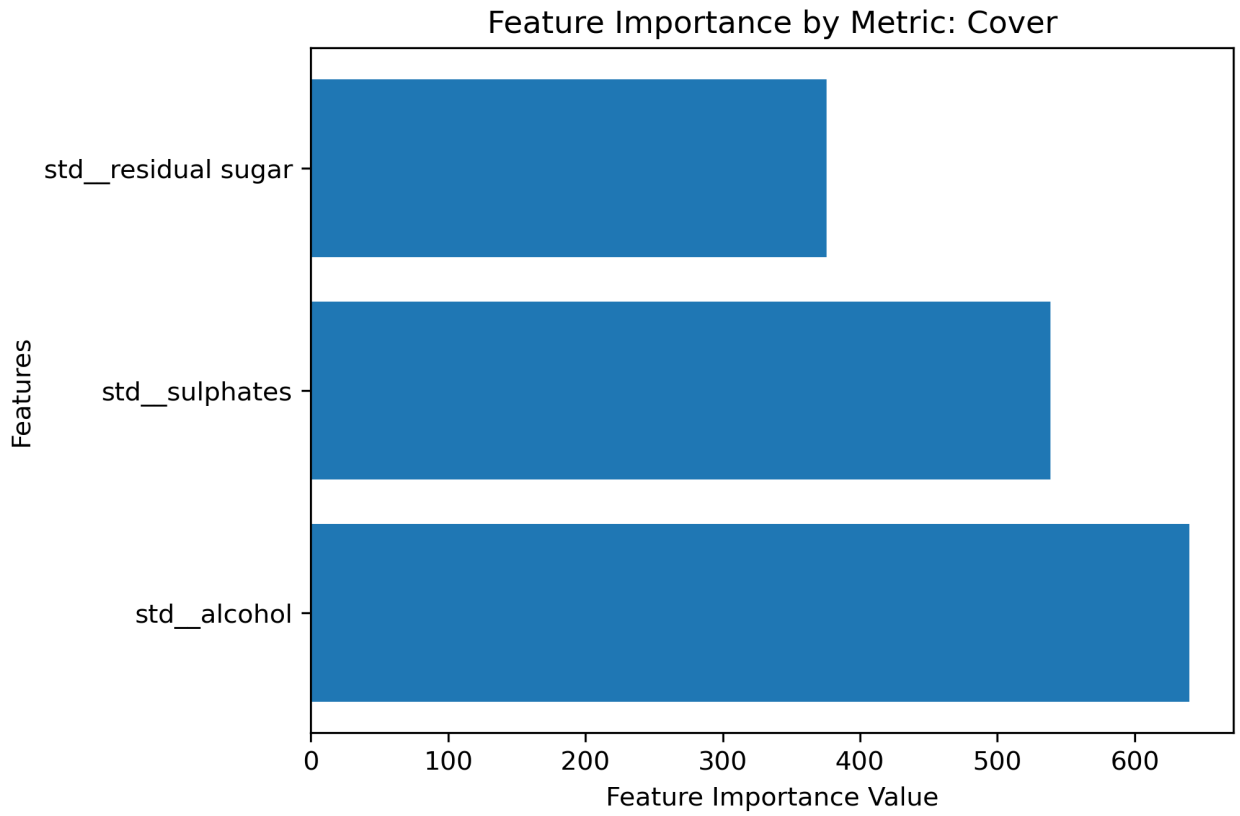

Figure 7) The top 3 important features by cover for XGBoost's random state with the combination of hyperparameters that produced the best score.

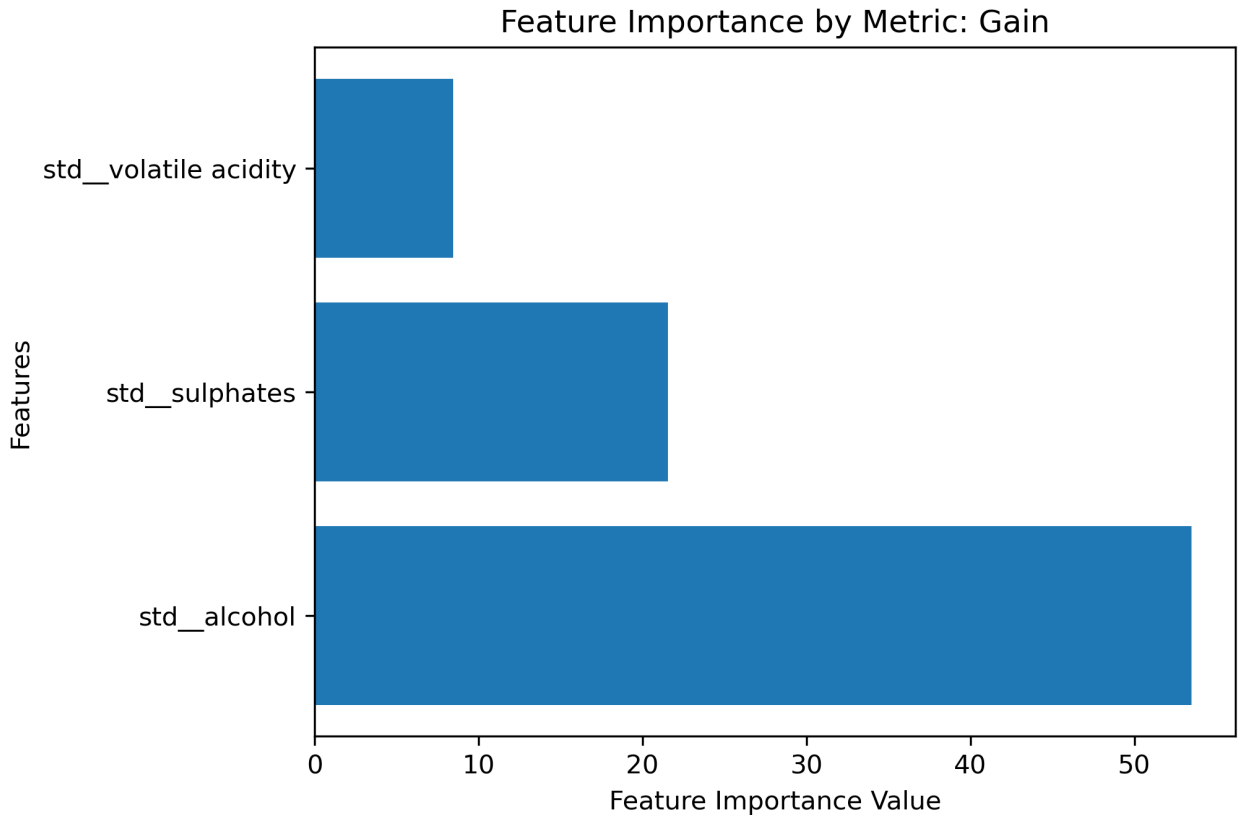

Figure 8) The top 3 important features by gain for XGBoost's random state with the combination of hyperparameters that produced the best score.

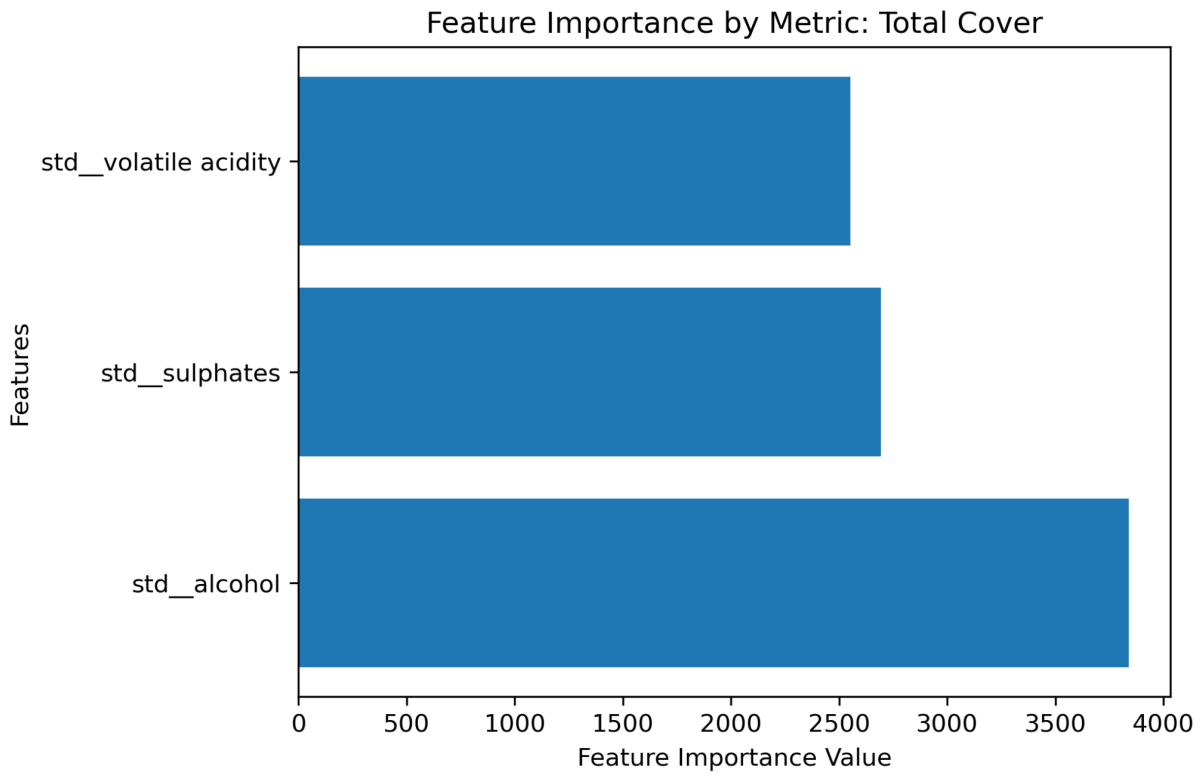

Figure 9) The top 3 important features by total cover for XGBoost's random state with the combination of hyperparameters that produced the best score.

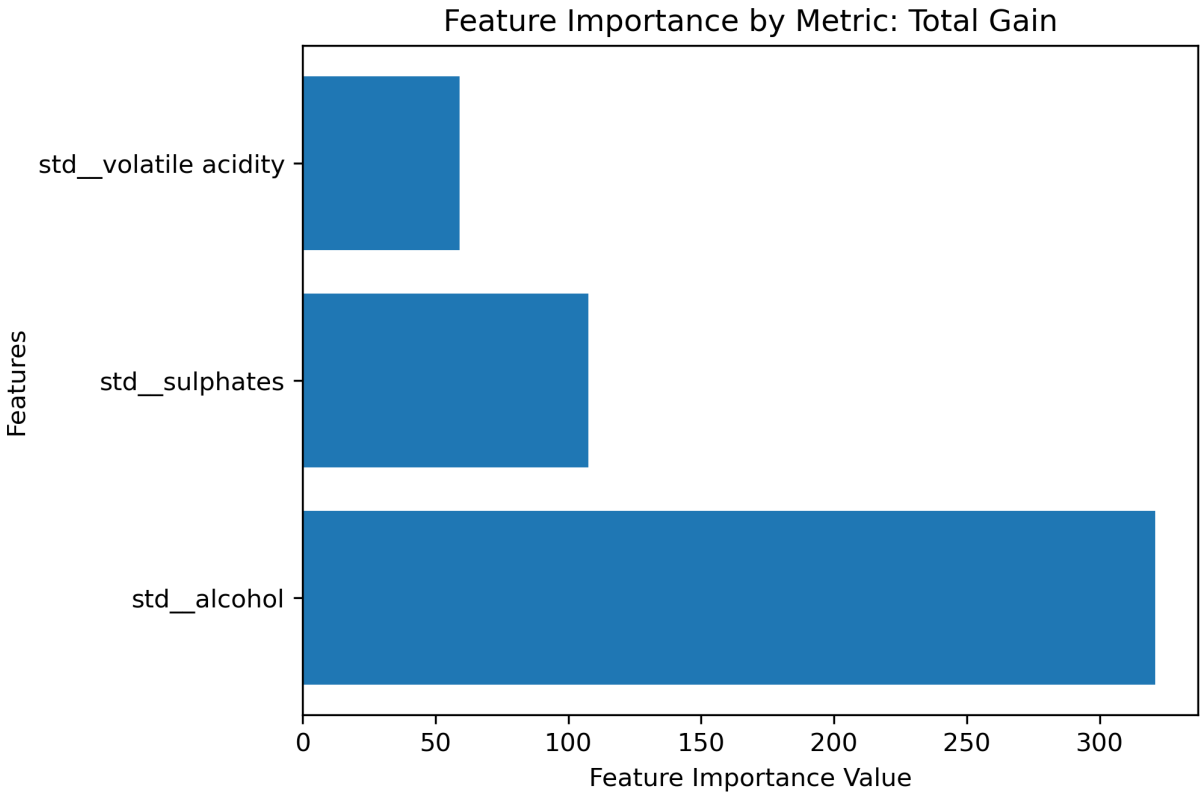

Figure 10) The top 3 important features by total gain for XGBoost's random state with the combination of hyperparameters that produced the best score.

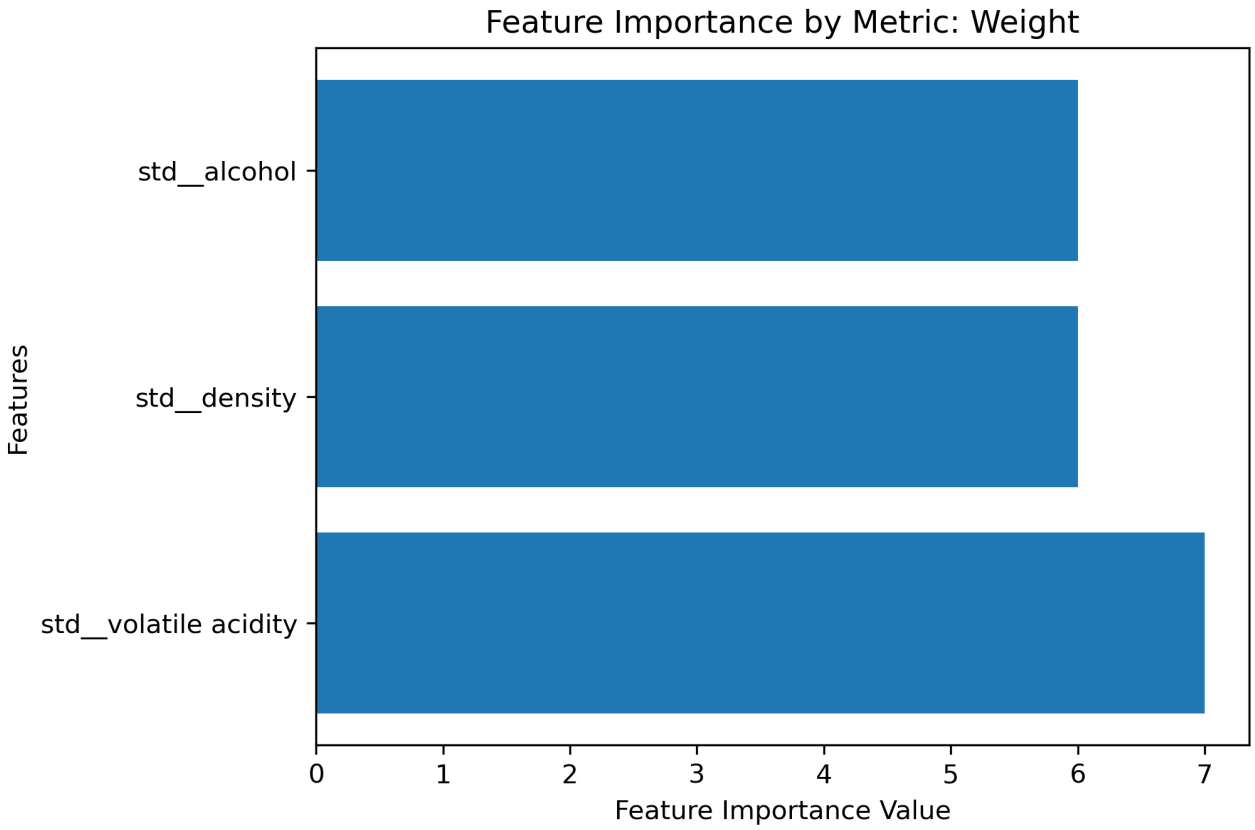

Figure 11) The top 3 important features by weight for XGBoost's random state with the combination of hyperparameters that produced the best score.

	Cover Importance	Gain Importance	Total Cover Importance	Total Gain Importance	Weight Importance
0	std__alcohol	std__alcohol	std__alcohol	std__alcohol	std__volatile acidity
1	std__sulphates	std__sulphates	std__sulphates	std__sulphates	std__density
2	std__residual sugar	std__volatile acidity	std__volatile acidity	std__volatile acidity	std__alcohol

Table 3) The top 3 features, regardless of the metric used, include: alcohol, sulphates, and volatile acidity.

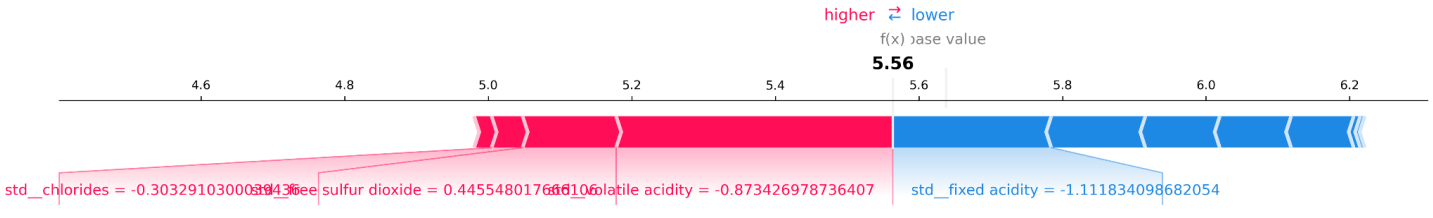

Figure 12) For this datapoint, the prediction is highly positively influenced by: fixed acidity and free sulfur dioxide while it is highly negatively influenced by: fixed acidity. This data point has a quality rating slightly less than the expected (5.6376586)

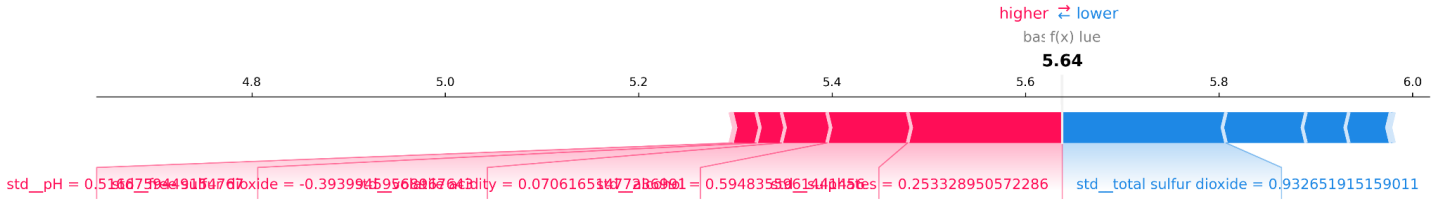

Figure 13) For this datapoint, the prediction is highly positively influenced by: sulphates and alcohol while it is highly negatively influenced by: volatile acidity and fixed acidity. This data point has a quality rating slightly greater than the expected (5.6376586)

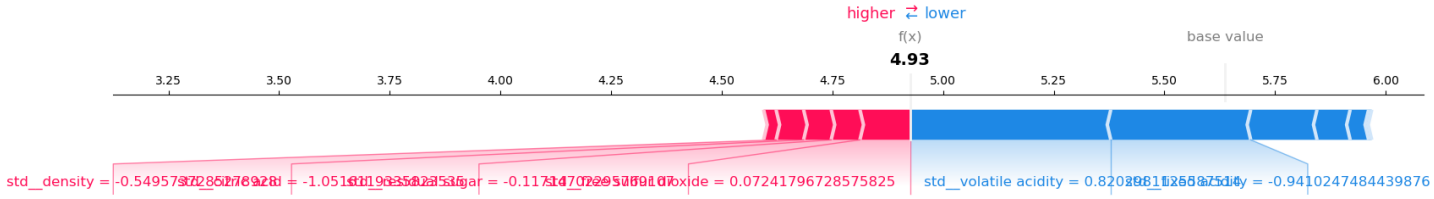

Figure 14) For this datapoint, the prediction is slightly positively influenced by: free sulfur dioxide and residual sugar while it is highly negatively influenced by: volatile acidity and fixed acidity. This data point has a quality rating less than the expected (5.6376586)

Works Cited:

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.

UCI Machine Learning. "Red Wine Quality." *Kaggle*, 2017, https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009.

UCI Machine Learning. "Wine Quality Data Set." *UCI Machine Learning Repository: Wine Quality Data Set*, 7 Oct. 2009, https://archive.ics.uci.edu/ml/datasets/wine+quality.