# **Red Wine Quality**

# CISC684 Final Project

Group 5

Jie Ren, Long Chen, Yifan Wang, Yun Tang

**Dec 15 2019**

# 

# **Introduction**

The product quality certifications are the most important indicators for industries to promote their products [1]. The same is true of wine industry. However, the evaluation is usually carried out by human experts, which is both economically expensive and time consuming. To accelerate the assessing process, physicochemical laboratory tests are used routinely to characterize wine, including determination of density, alcohol or pH values, etc. Modeling wine preferences from the perspective of data is useful to improve wine production or support the oenologist wine tasting evaluations. [2] This project used a dataset of 1,599 red wine entries with quality levels varied between 0 and 10, as well as 11 physicochemical features, such as fixed acidity, chlorides, pH, alcohol, etc. This kind of work is necessary and useful, especially for small wine distributors who want to accelerate and improve the process of wine storing, classifying, organization, and fast deliver to liquor stores around the country. The goal of this project was to choose key attributes from the data that play the most important roles on the quality level of the red wine. The ultimate goal of the project is to find a model of the attributes to predict the quality levels. Therefore, the wine distributors will know which category a new wine belongs to immediately. The outcome of the project would accelerate both the evaluation process and the marketing process.

To clarify our dataset, the input and output of the model are:

**Ⅰ. Input:** 11 attributes of chemical properties that influence the quality of red wine. (Description and the specific detective chemicals involved) [2]

1. Fixed acidity: (tartaric acid-g/dm3) most acids involved with wine or fixed or nonvolatile (do not evaporate readily).

2. Volatile acidity: (acetic acid-g/dm3) the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste.

3. Citric acid: (g/dm3) found in small quantities, but can add 'freshness' and flavor to wines.

4. Residual sugar: (glucose-g/dm3) the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweets.

5. Chlorides: (sodium chloride-g/dm3) the amount of salt in the wine, which contributes the salty taste of a wine.

6. Free sulfur dioxide: (mg/dm3) the free form of SO2 that exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine.

7. Total sulfur dioxide: (mg/dm3) amount of free and bound forms of SO2; 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.

8. Density: (g/cm3) the density of water is close to that of water depending on the percent alcohol and sugar content.

9. pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale.

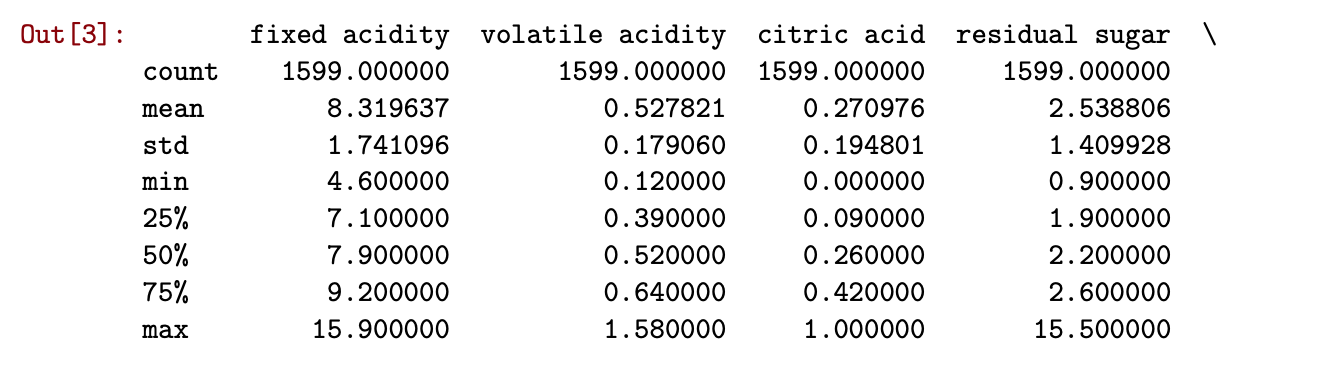
10. Sulphates: (potassium sulphate-g/dm3) a wine additive, which can contribute to sulfur dioxide gas (SO2) levels, which acts as an antimicrobial and antioxidant.

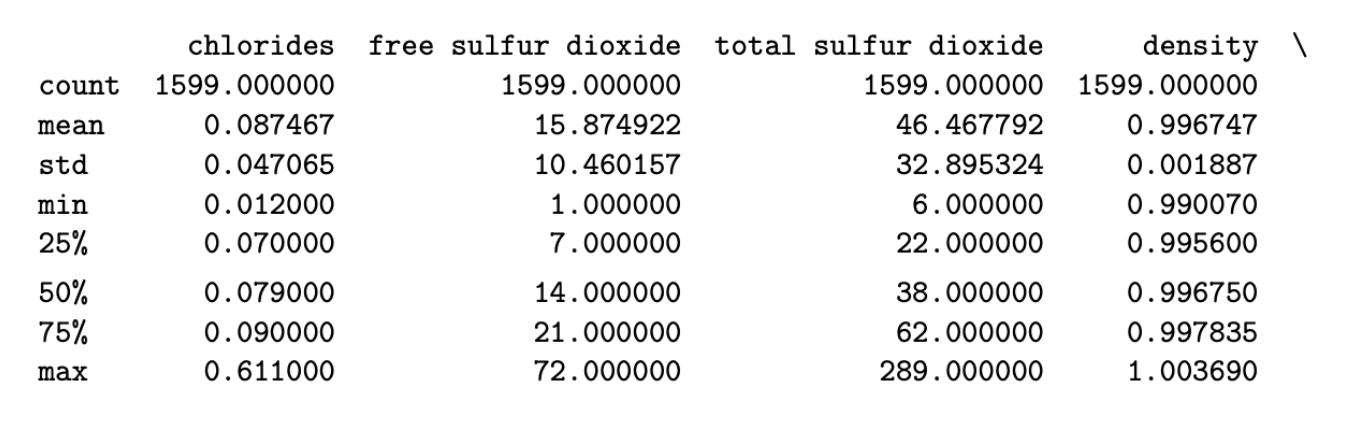
11. Alcohol: (% by volume). the percent alcohol content of the wine.

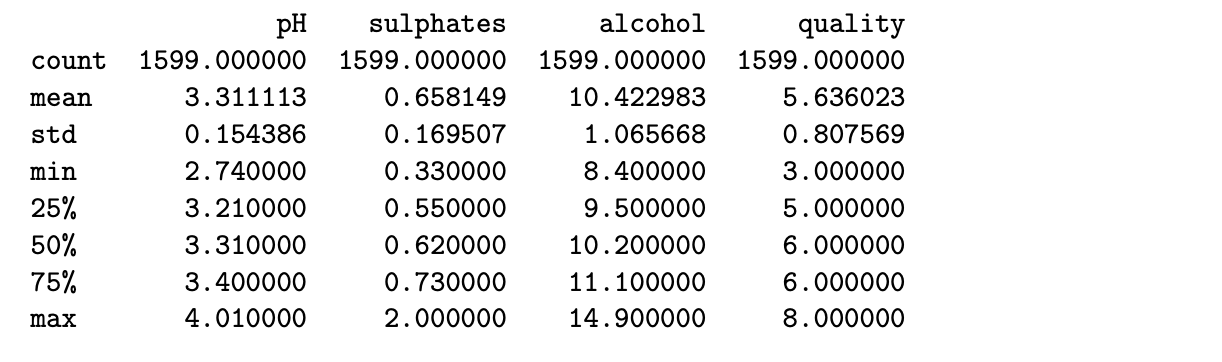
**Ⅱ. Output:** Quality level of red wine (score between 0 and 10) (discrete numerical variable).

# **Prepare**

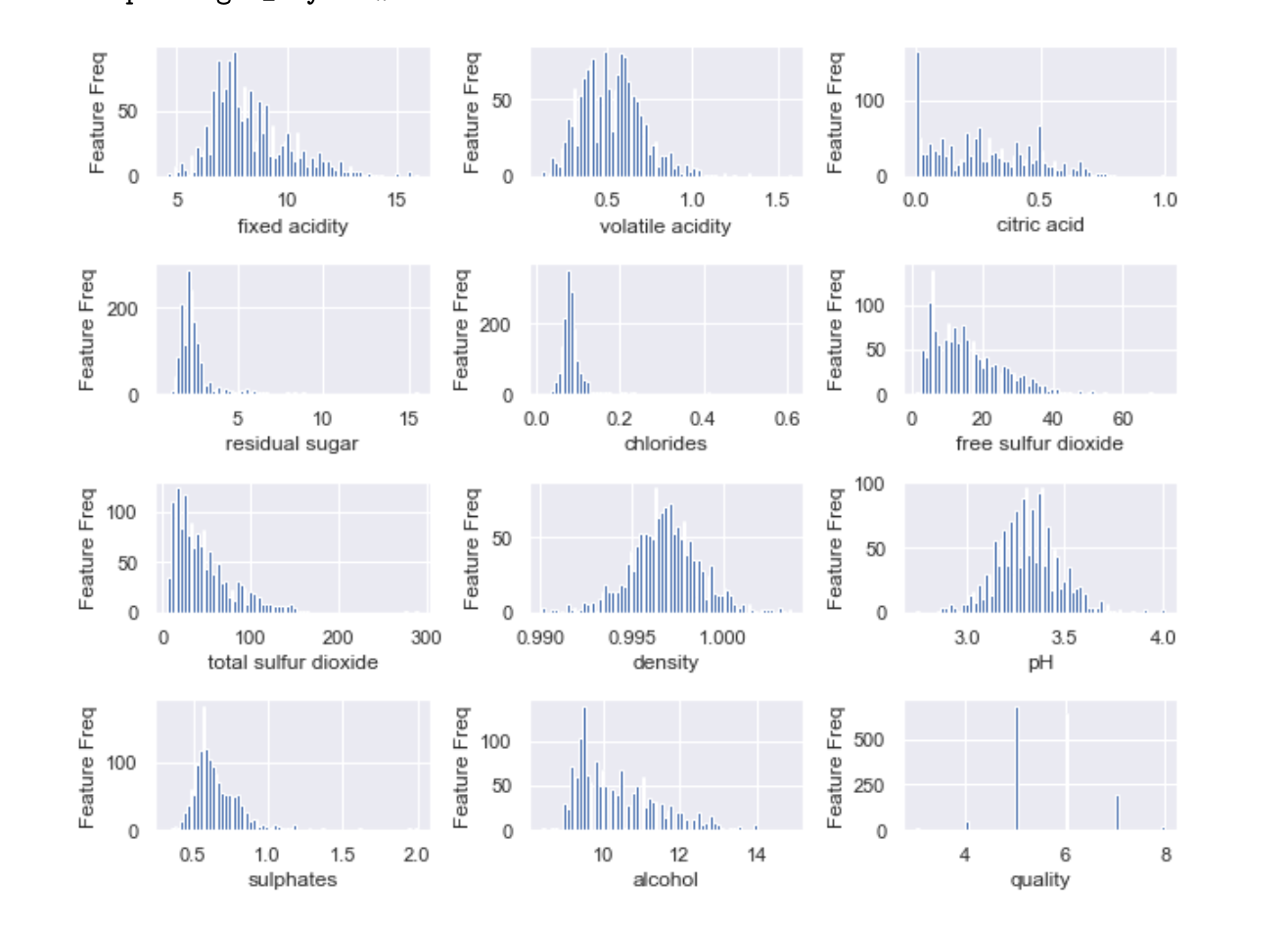
First, we examine the data. Some basic statistics of each feature is shown below.



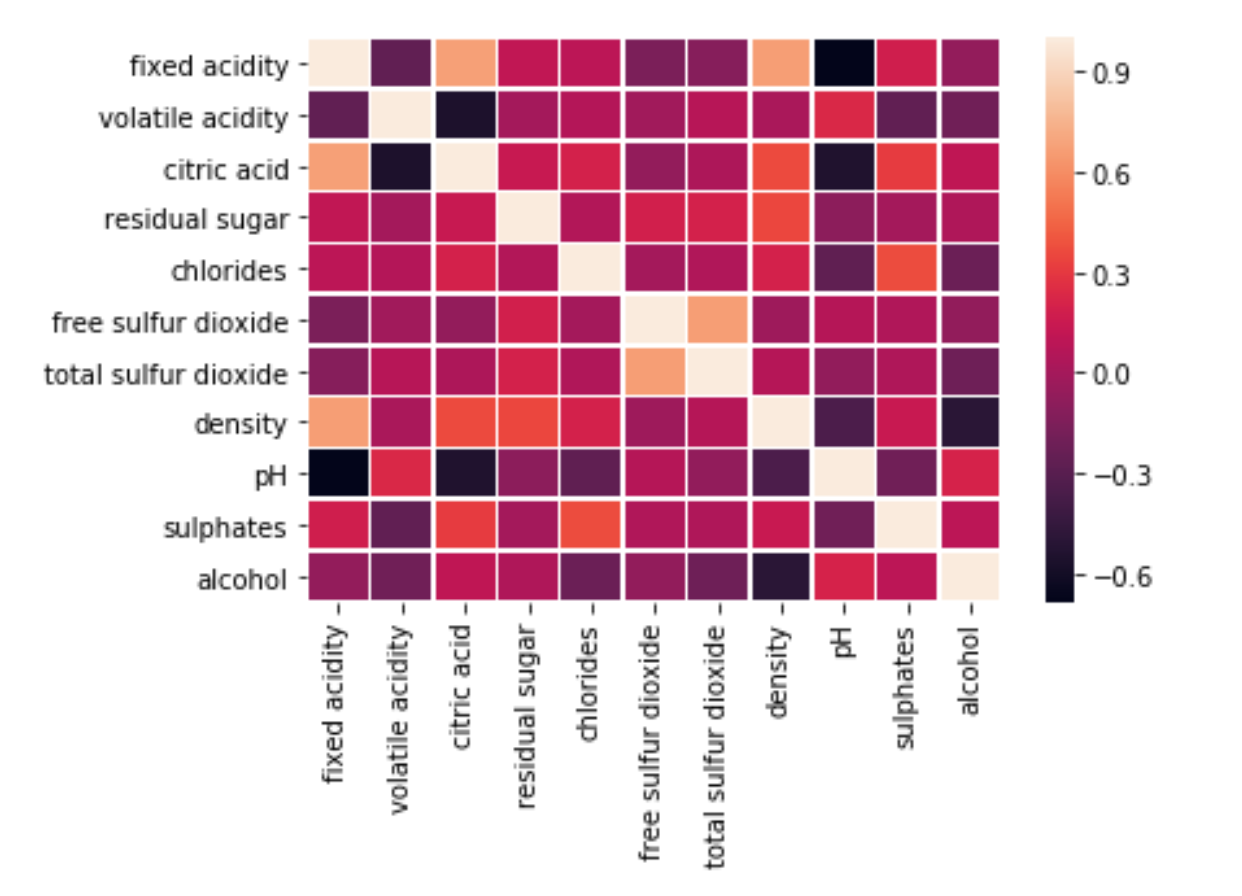




Then, we looked at the distribution of each feature. We can see that a number of features follow normal distribution and a few features have positive skewed distribution.



To discover the correlation between numerical attributes, we calculated the Pearson correlation coefficient and used heatmap to show it directly:



From the heat map, we could not identify a strong correlation between those variables, so the feature selection process needs to be further developed.

To evaluate the algorithms we use, we used the dataset splitting process. We used the sklearn package to do this:

**x\_train,x\_test,y\_train,y\_test= train\_test\_split(dataset.ix[:,0:11],dataset['quality'],random\_state=42)**

In this line of code, we set the random\_state to a fixed value so that we can get the same result in the second time. Therefore, we have 75% of the dataset as train set and the rest of the dataset as test set.

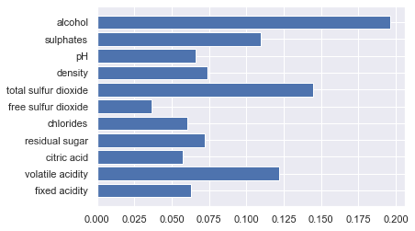
# **Algorithms**

In this project, we used several machine learning algorithms to achieve a model of predictions.

**Ⅰ.** First, we used ***decision tree*** as the baseline. It is a tree structure, in which each internal node represents a test on an attribute, each branch represents a test output, and each leaf node represents a category. Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map nonlinear relationships as well. [3] Therefore, the first model we tried was the decision tree. In the model, entropy was used as the evaluation standard, and the top-down recursive method was used to build a tree with the fastest entropy decline. The entropy of random variable X can be defined as follows:

It's easy to build a decision tree in practice, we just need to import the class from scikit-learn.

Decision Tree can help us to explore the dataset, at the same time it will perform the feature selection process by using the entropy criterion. In that case, we can discard redundant features and reduce the complexity of the algorithm, so that we don’t need to use all attributes. The feature importance chart can be shown as follows:



The accuracy of decision tree is somewhat low, the test score is just 0.570 while the train accuracy is 1.000! We have good reasons to think that the algorithm is over-fitted. But we can select some variables for the next step analysis. In fact, we only choose the features whose feature importance is above average. The attributes are: 'volatile acidity', 'total sulfur dioxide', 'sulphates', and ‘alcohol’.

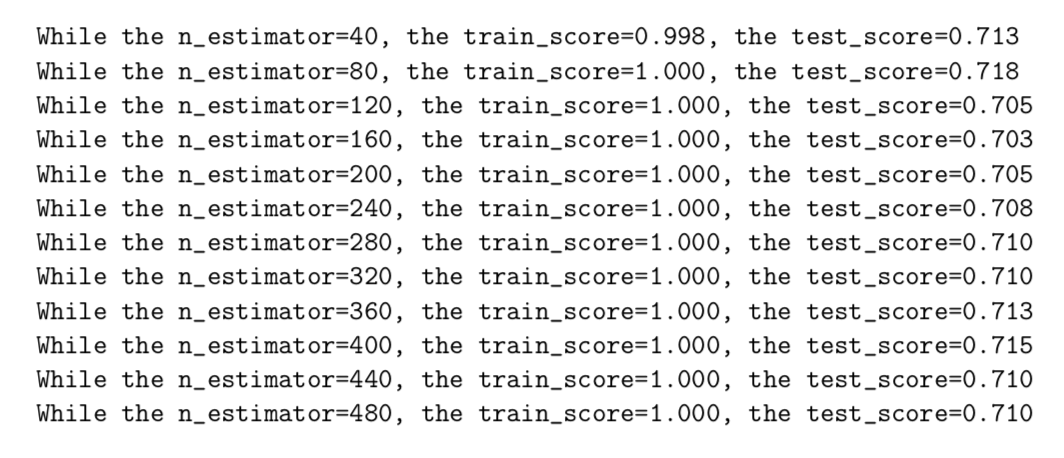
**Ⅱ.** Then we tried to make classification by a famous algorithm: Support Vector Machine. “Support Vector Machine” (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane (MMH) that best divides the dataset into classes. [1][4] Our problem is to classify the quality level for a specific wine that is based on several properties, so SVM is an appropriate model we can use in the dataset. Here, we choose the Radial Basis Function Kernel (RBF kernel), since the RBF kernel is a popular kernel function commonly used in support vector machine classification. RBF can map an input space in infinite dimensional space. First, we import the SVM module and create support vector classifier object by passing argument kernel as the linear kernel in SVC() function. Then, fit our model on train set using fit() and perform prediction on the test set using predict(). At the end, we estimate how accurately the classifier can predict the quality. Accuracy can be computed by comparing actual test set values and predicted values.

When we used the SVM algorithm on all the features, accuracy score of the train set was 0.756 and of the test set was 0.540. The difference of the accuracy between two data sets suggests overfitting. So, we decided to combine the SVM algorithm with feature selection by decision tree. We applied SVM again on only the attributes, 'volatile acidity', 'total sulfur dioxide', 'sulphates', and ‘alcohol’, which were selected in a decision tree model in part I, according to their feature importance. The accuracy of the training data set was 0.626 and of the test data set was 0.580. Therefore, the overfitting was decreased and we got a more robust model with less variance.

|  |  |  |
| --- | --- | --- |
|  | Before feature selection | After feature selection |
| Accuracy of train set | 0.756 | 0.540 |
| Accuracy of test set | 0.626 | 0.580 |

**Ⅲ.** Bootstrap Aggregation (or Bagging for short) is a simple and very powerful ensemble method. An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. Bagging is a general procedure that can be used to reduce the variance for those algorithms that have high variance. The overfitting problem appeared in our first model because that decision trees algorithm is a greedy algorithm. It chooses which variable to split on using a greedy algorithm that minimizes error. Because of the high variance and too deep search of the individual trees, overfitting is a common problem. Therefore, Random Forests is an improvement over our decision trees algorithm. Since combining predictions from multiple models in ensembles works better if the predictions from the sub-models are uncorrelated or at best weakly correlated. Random Forest changes the algorithm in a way that the sub-trees are learned so that the resulting predictions from all of the subtrees have less correlation.[5] For the above reasons, we used the random forest model, and chose the number of trees as 300 (n-estimate=300). The accuracy score of the train set was 1.000 and of test set was 0.695.

The number of basic estimators is an important hyper-parameter. The value of this parameter will significantly affect the accuracy of the model. So we have to adjust the hyper-parameter to improve our model to better performance. We chose different n-estimator from 40 to 560 and calculated both train score and test score and the results are as following screenshot.



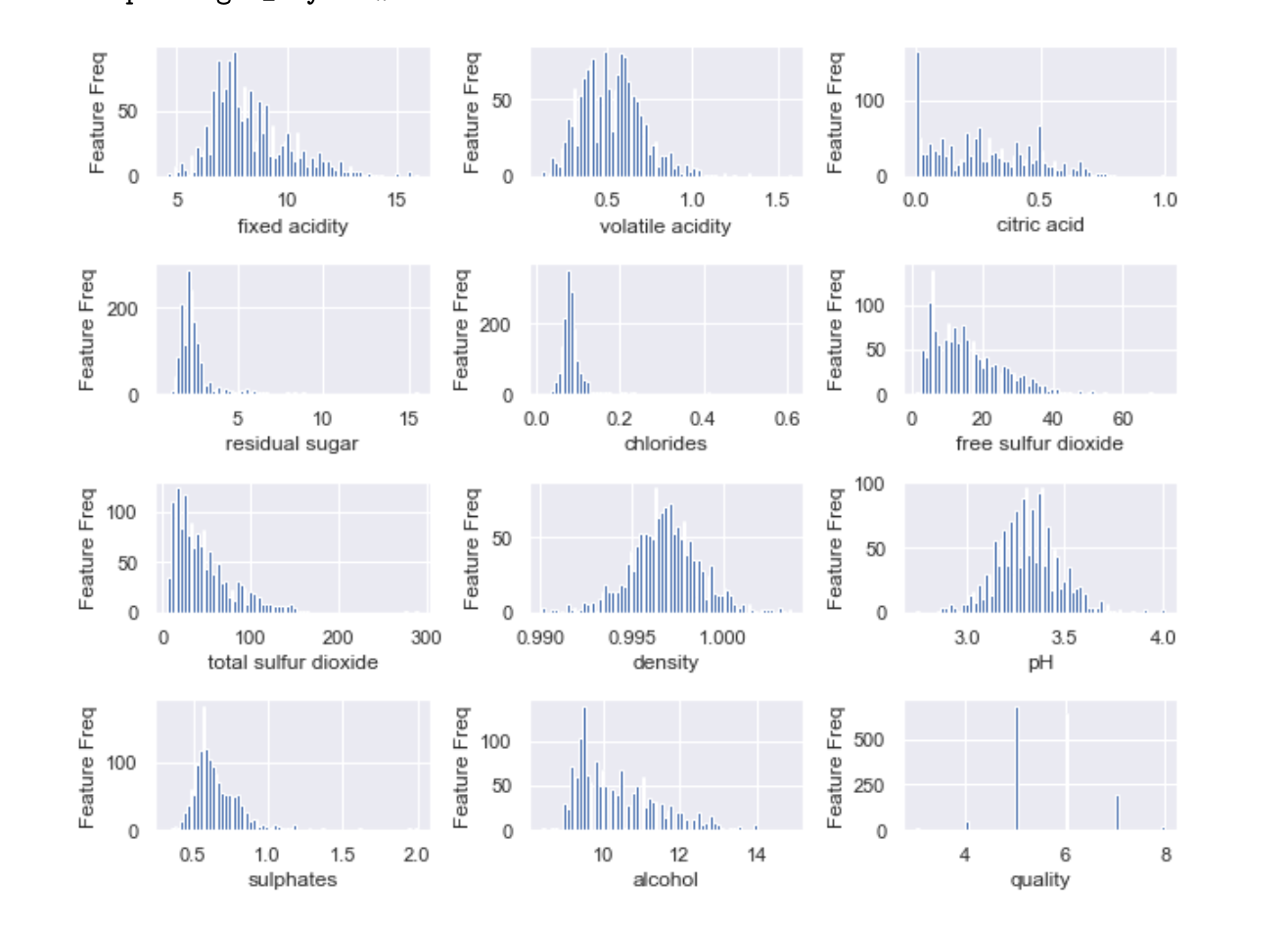
We could see that when the hyper-parameter value was 80, it performed best on the test set, so we chose 80 as the final parameter.

**Future Work**

There are two points we need to mention here.

First, when we did this project, we found even though we tried three different models, overfitting happened on most of the models and only decision tree+SVM had less overfitting. We think this may be because the nonparametric models we chose have more flexibility when learning a target function. The next step we need to do is to try to find some parameters or techniques to limit and constrain the models.

Second, as a matter of fact, we can get more information from the distribution of the features.(Figure shown below) We can see that the distribution of quality is unbalanced. So, our prediction results have some defects. The same phenomenon can be shown in a confusion matrix plot (not shown here), too. We have no way to truly verify if it is correct due to the lack of additional data. The next step we need to do is to randomly split dataset and iterate 10 to 30 times without using any model, and run evaluation of the correctness of our dataset. We can also take measures such as down sampling, stratified sampling.



**Conclusion**

In this project, we tested three algorithms, Decision Tree, SVM, and Bagging to classify the quality of 1599 entries of wine data with 11 physicochemical features. Our final model first used Decision Tree for feature selection ( and selected 'volatile acidity', 'total sulfur dioxide', 'sulphates', and ‘alcohol’ as the key features). Then, we ran SVM on the selected features, using the Radial Basis Function Kernel (RBF kernel) to get the final model. The accuracy of our final model on the test dataset was 0.580, which was not impressive but good enough since it is much higher than a random guess, which is 10% (one out of 10 levels). The lessons we learned from the model is that, although there were 11 features involved in the quality certification of red wine, there were only 4 features that play a crucial role in the quality level classification: 'volatile acidity', 'total sulfur dioxide', 'sulphates', and ‘alcohol’. We noted that these four features are distinct from each other in their physico-chemical properties. Recall that in the original dataset, there were a number of features related to acidity, dioxide, sulphates, etc. However, only one of these groups of features were selected. Therefore, our model suggests the four most important features in the quality certification. This could provide a truncated and more convenient method for the certification, which will save both economic and time cost. Wine distributors now can use our model to quickly predict the quality level of the wine they receive and accelerate the marketing service.

**References:**

[1] Gupta, Y. (2017, December). Selection of Important Features and Predicting Wine Quality Using Machine Learning Techniques. *​6th International Conference on Smart Computing and Communications*, *ICSCC 2017, 7-8*.

[2] 1. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. ​*Decision Support Systems,​ ​47(​ 4), 547-553.*

2. Marina S. C., Angelo G. R., Elizangela S. S., Margareth F. D., Ana Cristina N. C. Chloride concentration in red wines: influence of terroir and grape type. *Food Sci. Technol, Campinas, 35(1): 95-99, Jan.-Mar. 2015*

*3.* [https://en.wikipedia.org/wiki/Acids\_in\_wine#In\_wine\_tasting](https://en.wikipedia.org/wiki/Acids_in_wine" \l "In_wine_tasting)

4. <https://pt.wikipedia.org/wiki/PH>

5. <https://en.wikipedia.org/wiki/Feature_selection>

[3] <https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb>

[4] <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>

[5] <https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/>