Machine Learning based Wine Quality Prediction

Li Xinyue Erika
Data Science
BNU-HKBU United International College
m730026048@mail.uic.edu.hk

***Abstract*—The pursuit of wine has become increasing demanding, as a result, the standardized rating of wine is becoming important. In this project, to build a wine level prediction model, different algorithms have been used and compared. I divided the original problem into to two parts, binary classification problem and multi-class classification problem. Firstly, in the binary classification problem, Random Forest Classifier and Ensemble Learning give the best performance. Secondly, in the multi-class classification problem, I use classification approach and regression approach. I also use Keras framework to implement the neural network approach. The Random Forest Regressor and Bagging Regressor gives the highest result for multi-class classification problem. Moreover, I implement some wine type classification to train a classifier to distinguish red wine and white wine, which gives a good result.**

***Index Terms*—Multi-class Classification, Neural Network, Ensemble Learning, Random Forest, XGBoost**

I. Background and Motivation

Nowadays, people pursue higher and higher quality of life, and we like to drink wine in our daily lives, but we all know that the price of wine in the supermarket varies, and we often don't know how to distinguish a good wine from a bad one. And as we known to all, the value of the wine is highly related to its quality and the better the quality, the higher the value.

A good wine taster can judge the quality of a wine from its various properties. However, novices may not be able to analyze a wine's quality by tasting its sweet, sour, bitter, salty taste and aroma as experienced wine drinkers do. Moreover, the taste and rating of any food is highly subjective, and everyone has different preferences, so the standardized rating of wine is becoming more and more indicative. As a result, we need a model that can do a standard analysis of the quality of wine.

In this project, based on the dataset about red wine and white wine, I approach this problem by several methods. First of all, I divide the red and white wine into two classes, high quality and low quality, separately and apply binary classification algorithms to get a classifier to approximate the quality of the wine. For me, this is far from enough. I then apply multi-class classification algorithms. Innovatively, for multi-class classification problem, I also use some regression method. At the end, in addition to the prediction of wine quality, I also tried the prediction of wine types. To get a better score, I implement parameter selection. For each problem, I also use neural network based on deep learning framework Keras.

II. Related Work

There are several studies using some simple machine learning with the same dataset [1]. Due to the imbalance of the dataset, most of the existing work turns the problem into a binary problem, instead of getting a precise classifier, they simply divided the wine into two class, good wine and bad wine. [2]

Vishal Kumar proposed a binary classification method to roughly predict the quality using Random Forest Classifier. Mehmet KOMURCU also used Support Vector Classifier to approximate the quality of the wine. Moreover, Terence applied some feature selection method and XGBoost to get a classifier for good wine and bad wine.

There are also some studies using fuzzy logic tool [3], which also give good performances. Moreover, there are also a study that selected two class to do the classification and trained the classifier within two classes of wine.

III. Dataset, Features and EDA

A. Dataset, Features

The dataset for this project is downloaded from UCI machine learning repository, and it consists of two groups, the data about red wine and the data about white wine. For each group of data, it has 11 features, including **fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol**; and one output **quality**.

B. EDA

For the red wine dataset, it has 1599 tuples; and for white wine dataset, it has 4898 tuples. From the figure 1, we can see that the distribution is imbalance.

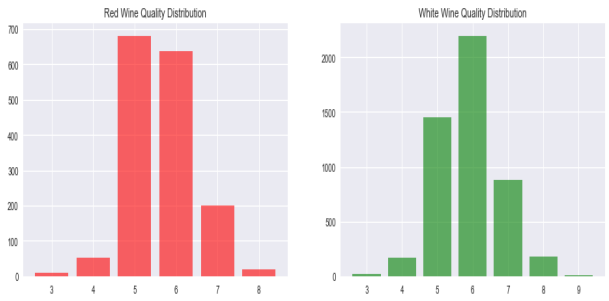

Fig. 1: Distribution of quality

The table 1 shows some statistics of the dataset. The quality for red wine is from level 3 to level 8; and the quality for white wine is from level 3 to level 9. From the description of the dataset, we can know that the level of the wine is between 1 to 10, as a result, if we use classification method, we can't get output 1, 2 and 10, because for classification approach, there is no training data for level 1, 2 and 10, as a result, the prediction value will not have these classes, which is a disadvantage of classification approach and it's not reasonable.

	red wine		white wine	
	min	max	min	max
fixed acidity	4.6	15.9	3.8	14.2
volatile acidity	0.12	1.58	0.08	1.1
citric acid	0	1	0	1.66
residual sugar	0.9	15.5	0.6	65.8
chlorides	0.012	0.611	0.009	0.346
free sulfur dioxide	1	72	2	289
total sulfur dioxide	6	289	9	440
density	0.99007	1.00369	0.98711	1.03898
pH	2.74	4.01	2.72	3.82
sulphates	0.33	2	0.22	1.08
alcohol	8.4	14.9	8	14.2
quality	3	8	3	9

TABLE I: Statistics for the dataset

Moreover, from the box-plot, we can see this dataset does not have too many outliers, we can use it safely.

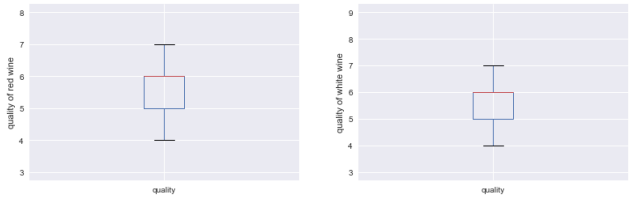

Fig. 2: Box-Plot for quality

From the following heat maps, we can know that the correlation between features, we can find that some of the features are highly related, and from my point of view, it is a nice idea to use regression model to fit it.

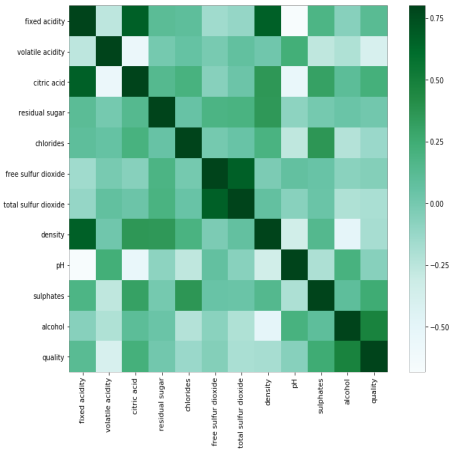

Fig. 3: Heat map for red wine

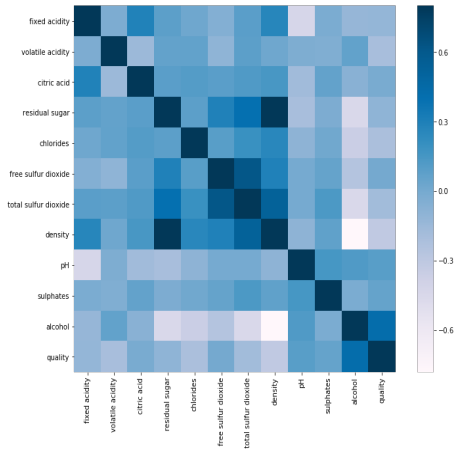

Fig. 4: Heat map for white wine

Moreover, we can know that some of the attributes are not normalize distributed, we may have to do the normalization before we applying models in order to get rid of the affect from some dominate features.

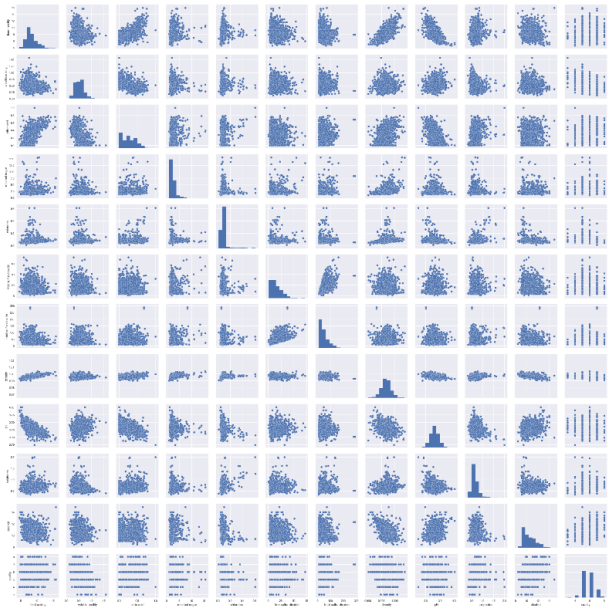

Fig. 5: Pair plots

IV. Methodology

A. Classification

As for the classification problem, I apply many different algorithms, for example, Logistic Regression Classifier, Random Forest Classifier, Support Vector Classifier, KNN, Decision Tree Classifier, Gradient Boosting Decision Tree, Ada Boost Classifier, Gaussian Naïve Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis and XGBoost. Among them, Random Forest Classifier, Support Vector Classifier, Gradient Boosting Decision Tree [4] and XGBoost gives a

better performance. What's more, I also use ensemble learning to improve the classification result.

1) Support Vector Classifier: Support Vector Classifier can also be called soft margin classifier, which is based on Support Vector Machine. SVM are proposed for binary classification problems and have been successfully applied to sub solution regression and a class of classification problems. Although support vector machines have achieved great success in solving binary classification problems, a large number of multi-valued classification problems in practical applications require how to extend SVM to multi-classification problems.

In the model with soft margin, it introduce positive slack variable ξ_i and the constrain is:

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

And for an error to occur, the corresponding ξ_i must exceed unity. And the optimization problem using soft margin becomes:

$$\min_{w,b} \frac{1}{2}||w||^2 + \sum(\xi_i)$$

$$s.t. y_i(w^T x_i \geq 1 - \xi_i, i = 1, 2, ..., m$$

$$\xi_i \geq 0, i = 1, 2, ..., m$$

Examples are now allowed to have margin less than 1, and if an example whose margin is $1 - \xi_i$, we will pay a cost of the objective function being increased by $C\xi_i$.

In later section, I use SVC and apply parameter selection, and it shows a good performance.

2) Random Forest Classifier: A random forest is a classifier that contains multiple decision trees, and its output categories are determined by the mode of the categories output by an individual tree [5].

Each tree is built according to the following algorithm:

1) Use N to represent the number of training cases (samples), and M to represent the number of features.
2) The number of input features M is used to determine the decision result of a node in the decision tree; where m should be much less than M.
3) A training set is formed by sampling N times from N training cases with fallback sampling, and the unselected use cases are used as prediction to evaluate the error.
4) For each node, m features are randomly selected, and the decision of each node in the decision tree is determined based on these features. According to these m characteristics, the optimal splitting mode is calculated.
5) Every tree will grow intact without pruning.

This algorithm is based on idea of ensemble learning and Decision Tree algorithm, it has many advantages:

- It can produce high accuracy classifiers
- Can handle a large number of input variables
- It is possible to generate internally unbiased estimates of generalized errors
- It can estimate missing data
- For an unbalanced set of classification data, it balances the error

We have already known that our data is imbalanced, as a result, it's a good idea for us to use Random Forest Algorithm to build the classifier. In later section, I use Random Forest Algorithm and apply parameter selection, and it really gives a good result.

3) Gradient Boosting Decision Tree: Gradient Boosting Decision Tree is also a kind of Boosting algorithm and it is similar to AdaBoost algorithm. However, there are still some differences, AdaBoost algorithm uses the error of the previous round of weak learners to update the sample weight value, and then iterates round by round. GBDT is also iterative, but GBDT requires that the weak learner must be a CART model, and GBDT requires that the sample loss predicted by the model should be as small as possible during model training.

GBDT model can be expressed as the addition model of decision tree:

$$f_M(x) = \sum T(x; \theta_m)$$

The feed-forward algorithm is used to determine the initial gredient tree:

$$f_m(x) = f_{m-1}(x) + T(x'\theta_m)$$

Determine the parameters of the next tree through empirical risk minimization:

$$\hat{\theta_m} = argmin \sum L(y_i, f_{m-1}(x_i) + T(x_i; \theta_m)$$

In later section, I use Gradient Boosting Decision Tree it also gives a good result.

4) XGBoost: XGBoost is an optimized distributed gradient enhancement library designed to be efficient, flexible, and portable. It realizes the machine learning algorithm under the Gradient Boosting framework. XGBoost provides GBDM that can quickly and accurately solve many data science problems [6].

XGBoost is an improvement of gradient boosting algorithm. Newton method is used to solve the extreme value of loss function. And the loss function is expanded to the second order using Taylor formula. In addition, regularization term is added into the loss function. The objective function is composed of two parts, the first part is the loss of gradient boosting algorithm, and the second part is the regularization term. The loss function is defined as:

$$L(\phi) = (y_i', y_i + \sum \Omega(f_k)$$

Where n is the number of training function samples and l is the loss to a single sample.

The regularization term defines the complexity of the model:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$$

and γ are self-defined parameters. And I use XGBoost in later section and it gives a good result.

5) Ensemble Learning-Voting: For the prediction of classification problem in Ensemble Learning, we usually use the voting method. The simplest voting method is the relative majority voting method. In other words, among the predicted results of sample x of T weak learners, the category with the largest voting number is the final classification category. If more than one category receives the highest number of votes, one is chosen at random as the final category.

There is another method of voting called Weighted voting method. The votes for each weak learner are multiplied by a weight, and the weighted votes for each category are summed, with the largest value corresponding to the final category.

In our classification problem, I apply ensemble learning based on XGBoost, Random Forest Classifier and SVC and it gives high accuracy and performance.

B. Regression

All of the above advanced classifiers can be used as regression problems and can be called directly from the package of sklearn. In our project, I use Decision Tree Regressor, Support Vector Regressor, K Neighbors Regressor, Random Forest Regressor, AdaBoost Regressor, Gradient Boosting Regressor, Bagging Regressor, Extra Tree Regressor and Ridge Regression algorithm [4]. In this project, among these algorithm, Random Forest Regressor and Bagging Regressor give good result. The basic core of these algorithms is consistent with the above classification algorithm, so I will not repeat them here.

C. Neural Network and Deep Learning Framework

Keras is a very convenient deep learning framework with either TensorFlow or Theano as the back end. It can quickly build deep network and flexibly select training parameters for network training. Besides traditional machine learning

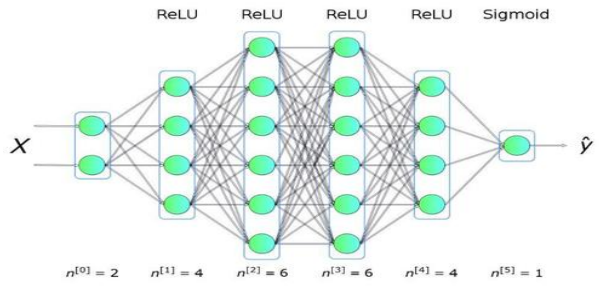

Fig. 6: Neural Network

algorithm, in this project, I also use Neural Network as a reference. Based on Keras, I compared different layer Neural Networks' performances and accept the one with highest score.

V. Experimental Study and Result Analysis

A. Binary Problem

Most of the existing studies divide the dataset of the wine into two class, good wine and bad wine, and transfer the problem into a looser one, binary classification problem. According to the description of the dataset, if the level of the wine is lower than 6.5, the wine is bad, otherwise, is good.

First of all, I follow previous studies, divide the dataset into two parts and apply classification algorithm based on this binary problem. The quality value higher than 6.5 will be assigned value 1 and lower than 6.5 will be assigned 0. Before applying any algorithm, I also do the normalization for the data. This step will reduce the effect of some features.

I split the dataset into two parts, 80 percent for training and 20 percent for testing. 10 folds cross validation is also used to get the mean accuracy score and get the more objective understanding of the model.

For the binary classification problem, I use Logistic Regression Classifier, Random Forest Classifier, Support Vector Classifier, KNN, Decision Tree Classifier, Gradient Boosting Decision Tree, Ada Boost Classifier, Gaussian Naïve Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis and XGBoost to build the model and compare the result. We use Logistic Regression as the base line, if the performance is worse than KNN, the algorithm is not suitable in this problem.

Algorithm	Red wine	White wine
Random Forest Classifier	**0.9281**	0.8531
Logistic Regression Classifier	0.8844	0.7990
SVC	0.9219	0.8223
KNN	0.8969	0.8163
Decision Tree Classifier	0.8906	0.7908
GBDT	0.9156	0.8357
AdaBoost Classifier	0.8906	0.7878
GaussianNB	0.8281	0.7337
Linear Discriminant Analysis	0.8781	0.8051
Quadratic Discriminant Analysis	0.8375	0.7571
XGBoost	0.9063	**0.8571**

TABLE II: Accuracy for Binary Classification Algorithm

The highest score in binary classification problem for red wine is produced by Random Forest Classifier and the accuracy score is around **0.9281**, and for white wine is XGBoost, the accuracy score is about **0.8571**. According to the base line, the algorithm Decision tree, AdaBoost Classifier, GaussianNB, Linear Discriminant Analysis and Quadratic Discriminant Analysis is not suitable for this part of classification problem.

After applying these algorithms, I do the parameter selection using grid search method, the best parameter for Random Forest for red wine is *max_depth* is 5, *min_samples_split* is 50, *min_samples_leaf* is 10 and *max_features* is 3, and for white wine, is *max_depth* is 11, *min_samples_split* is 50, *min_samples_leaf* is 10 and *max_features* is 3. As for SVC, the best parameter is *C* is 1, *gamma* is 0.9, *kernel* is 'rbf'.

Based on the good performance models, Random Forest Classifier, SVC and XGBoost, I apply ensemble learning. Every classifier has the same weight, the result is the most voted class. And the ensemble score for red wine is **0.9437**, and for white wine is **0.8459**.

Based on Keras package, I also use Neural Network as a reference. I want to check if deep learning model can improve the performance or not. As our problem is a simple

binary classification problem, I directly use three-layer Neural Network, and apply the ReLU as the activation, the optimizer I choose is Adam and the loss function is binary cross entropy. For each model, I apply 20 epochs and the following are the loss and accuracy versus epochs.

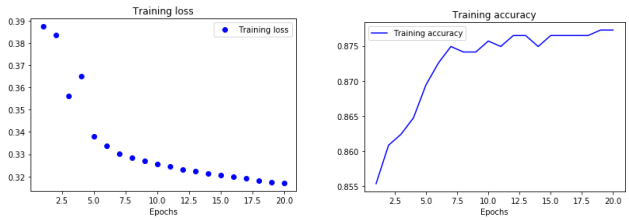

Fig. 7: Training for Red Wine

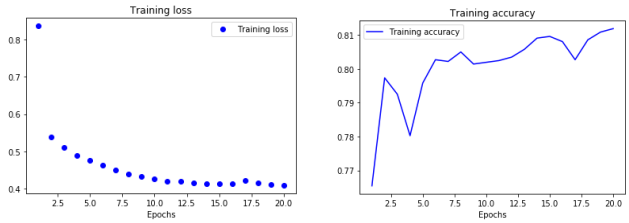

Fig. 8: Training for White Wine

B. Multi-class Problem

Approximate prediction for good wine and bad wine cannot satisfy the real-world condition. To get a better classifier, we have to solve the classification problem for 10 levels of wine.

Similar to the binary problem, I also do the normalization for the data and split the dataset into two part: training set and testing set.

1) Classification approach: For this multi-classification problem, the most direct and simple way is to use some multi-classification algorithm. Compared with the binary classification problem, we can use the similar algorithm to get the result. I also apply Ensemble Learning based on Random Forest Classifier, XGBoost and SVC. Here we also use Logistic Regression Classifier as the base line.

Algorithm	Red wine	White wine
Random Forest Classifier	**0.6969**	**0.6531**
Logistic Regression Classifier	0.6250	0.5000
SVC	0.6469	0.5418
KNN	0.6000	0.5429
Decision Tree Classifier	0.6928	0.5367
AdaBoost Classifier	0.5562	0.4388
XGBoost	0.6656	0.6000
Ensemble learning	0.6906	0.6051

TABLE III: Accuracy for Multi-Class Classification Algorithm

The highest accuracy is provided by Random Forest Classifier. And according to the base line, KNN and AdaBoost Classifier is not suitable for this problem.

2) Regression approach: From the EDA, we know that the distribution of the quality is closed to normal distribution, which means that there is no enough data for the training in low level and high level. It even has no data in level 1, 2 and 10, which is impossible for a classifier algorithm to work in these levels.

I tried to use regression algorithm to build the model. As the prediction value for a regression model is continuous, I just assign the output to its nearest integer. In this part, I use Decision Tree Regressor, Support Vector Regressor, K Neighbors Regressor, Random Forest Regressor, AdaBoost Regressor, Gradient Boosting Regressor, Bagging Regressor, Extra Tree Regressor and Ridge Regression algorithm to build different regression model and compare the accuracy.

Algorithm	Red wine	White wine
Decision Tree Regressor	0.6031	0.5428
SVM	0.6375	0.5480
KNN	0.5938	0.5224
Random Forest Regressor	**0.7188**	0.6081
Ada Boost Regressor	0.6250	0.4643
Gradient Boosting Regressor	0.6438	0.5286
Bagging Regressor	0.6938	**0.6092**
Extra Tree Regressor	0.6375	0.6061
Ridge Regression	0.6375	0.4776

TABLE IV: Accuracy for Multi-Class Regression Algorithm

Here I use Decision Tree Regressor as the base line and get rid of KNN. The highest accuracy is provided by Random Forest Regressor and Bagging Regressor.

Compared with classification approach and regression approach, it is easy to treat this problem as a classification problem, however, the accuracy for regression approach is higher.

Algorithm	Red wine	White wine
Random Forest Classifier	**0.6969**	**0.6531**
XGBoost	0.6656	0.6000
Ensemble learning	0.6906	0.6051

TABLE V: Top 3 Multi-Class Classification Algorithm

Algorithm	Red wine	White wine
Random Forest Regressor	**0.7188**	0.6081
Bagging Regressor	0.6938	**0.6592**
Extra Tree Regressor	0.6375	0.6061

TABLE VI: Top 3 Multi-Class Regression Algorithm

3) Neural Network approach: Similar to the binary problem, I also use Deep Learning framework Keras to apply some Neural Network models. Differently, as the multi-class classification problem is much more complex than the binary classification problem, as a result, we can use more complex neural network model. Besides training the model with more epochs, I also try to find the number of layers that gives the best accuracy score.

The best accuracy appears in 3-layer Neural Network and the best score is **0.6219** for red wine and **0.5234** for white wine.

	Red wine	White wine
3 layers	**0.6219**	**0.5234**
4 layers	0.6031	0.5021
5 layers	0.6093	0.5082
6 layers	0.5625	0.5122

TABLE VII: Accuracy for Different Layers of Neural Network

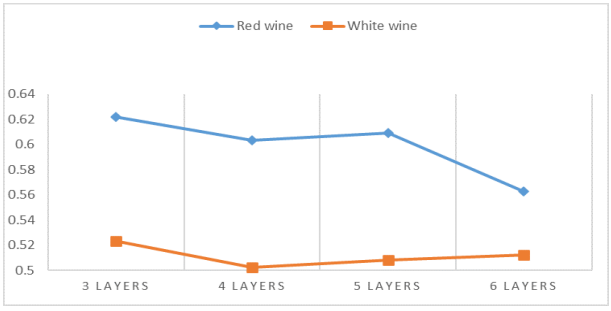

Fig. 9: Accuracy for Different Layers of Neural Network

Although I use complex Neural Network model to build the model, it does not improve the accuracy significantly. On the contrary, the prediction result for regression model is better than classification model and Neural Network because regression can train a model for low levels and high levels data, and as for classification model, it does not have data for level 1, 2 and 10, and the distribution of the data is imbalanced, which is difficult for classification algorithm to perform well.

VI. Further Study

A. Wine classification

Besides quality prediction, this dataset can also used to train a classifier to predict the type of the wine. I combine red wine data and the white wine data, as a result, the quality will be a feature and the type will be the output. Take the binary classification as reference, I build the model based on different classifiers and compare the accuracy. Take Logistic Regression Classifier as the base line.

Algorithm	Accuracy
Random Forest Classifier	0.9962
Logistic Regression Classifier	0.9938
SVC	0.9977
KNN	0.9954
Gradient Boosting Classifier	0.9938
AdaBoost Classifier	0.9877
XGBoost	0.9954
3-layer Neural Network	**0.9969**

TABLE VIII: Accuracy for Wine Type Prediction

The 3- layer Neural Network with 100 epochs gives the highest accuracy **0.9969**.

VII. Conclusion and Future work

In this project, different algorithms have been used and compared. I divided the original problem into to two parts, binary classification problem and multi-class classification problem, and in the multi-class classification problem, I use classification approach, regression approach and neural network approach and find that the regression approach gives a higher result. Moreover, I also implement some wine type classification to train a classifier to distinguish red wine and white wine, which gives a good result.

Due to the limitation of dataset, the multi-class classification accuracy is not as high as the binary classification one. In the future, I will try some other approach to solve this problem, for example, I can delete the low level and high level, train the classifier for the middle level wine. I also can do the sampling and make the distribution more balance.

VIII. Data Source and Code

The source for Data: https://archive.ics.uci.edu/ml/datasets/wine+quality

The implementation code: https://github.com/Erika1012/WineQualityPrediction

References

[1] P. Appalasamy, A. Mustapha, N. Rizal, F. Johari, and A. Mansor, "Classification-based data mining approach for quality control in wine production," *Journal of Applied Sciences*, vol. 12, no. 6, pp. 598–601, 2012.

[2] G. Hu, T. Xi, F. Mohammed, and H. Miao, "Classification of wine quality with imbalanced data," in *2016 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2016, pp. 1712–1217.

[3] S. Petropoulos, C. S. Karavas, A. T. Balafoutis, I. Paraskevopoulos, S. Kallithraka, and Y. Kotseridis, "Fuzzy logic tool for wine quality classification," *Computers and Electronics in Agriculture*, vol. 142, pp. 552–562, 2017.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[5] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[6] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.