STAT451 Project Final Report (wine quality)

Shihao Yang
syang536@wisc.edu

Yifan Zhang
yzhang2258@wisc.edu

Kexiao Zhu
kzhu8@wisc.edu

Abstract

Given the great popularity of wine in the past decades, understanding wine quality has been mentioned more than ever. In this project, we sought to fit machine learning models which could accurately predict wine quality. Our dataset includes 6000+ 'Vinho Verde' wine data with 11 features based on physicochemical tests. Here, KNN, SVM, Decision Tree, and other ten models are used as Algorithms and been compared. We select the model with the greatest accuracy to make further predictions. Furthermore, analysis are made on feature importance between random forest and extra trees methods.

1. Introduction

Nowadays, people love to have a cup of wine when they are celebrating or simply resting. Wine is an alcoholic drink fermented from grape juice. It's welcome in many different areas around the world.Aside from it's good growth situation, Research suggests that it's good for your health if you drink a glass of wine occasionally. Because wine provides antioxidants which may promote longevity, and can help to protect you against heart disease.

In this project, we are investigating a specific wine type - 'Vinho Verde'. 'Vinho Verde' wine has been popular in the past decade due to its irresistible fruity flavor, festive fizziness, and freshness. 'Vinho Verde' take up about 16.8 percent of the wine made in the country of Portugal, and Germany and the United States imported over 53 percent of its wines production [1].

One important characteristic of 'Vinho Verde' wine which makes it attractive to customers is it goes well with most types of food. Pairing the wine with light dishes like salads and chicken, it can also be accompanied by more filling meals like potato and pork dishes. In addition, the white 'Vinho Verde' also goes great with seafood dishes, which are very popular in Portuguese cuisine like cod and monkfish.

In our project, we want to utilize those factors to predict the quality score of the wine which is given from 0(terrible) to 10(excellent). There were 11 factors measured: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides,free sulphur dioxide, total sulphur dioxide, density, pH, sul-phates and alcohol.And in our dataset, it contains 6498 rows and 13 columns.

'Vinho Verde' wine is always been known as 'cheep and cheerful', which retail price can lower to 4 dollar. But various 'Vinho Verde' wine do exist great differences in their qualities. So we hope our model can accurately present the quality of 'Vinho Verde' wine, using their physicochemical tests data. Having the ability to predict the wine quality based on physicochemical tests will not cause a qualitative change to the wine, but it will allow the manufacturers to clearly positioned 'Vinho Verde' with different brands or genres, set most suitable retail prices, and aim corresponding customers. This will even enable wine merchants to advertise accurately depending on the customers' favorites on wine quality. In the terminology of Economics, it may increase the total surplus of 'Vinho Verde' wine market.

2. Related Work

Several studies have been conducted on wine quality and physicochemical wine data. Back in 2009, Cortez et al.[3] propose a data mining approach to predict human wine taste preferences that is based on easily available analytical tests at the certification step. They presented a novel method that performs simultaneous variable and model selection for NN and SVM techniques. In order to show its impact in this domain, they tested such approach in a real-world application, the prediction of vinho verde wine taste preferences. Their dataset contain a total of 4898 white and 1599 red samples, which are later modeled under a regression approach to preserves the order of the grades. They believe that their integrated approach is valuable to support applications where ranked sensory preferences are required, including wine or meat quality assurance.

Another study which is conducted recently by Yogesh Gupta(2018) explored different machine learning techniques such as linear regression, neural networks (NN) and support vector machines (SVM) for wine quality assurance using white and red wine data.[2] He used linear regression to determine the dependency of target variable on independent variables. And important variables which make significant impact on dependent variable are selected based on the dependency. Further, predictions are made on the value of quality by using neural network and support vector machine. The study seeks to prove that feature selection can result in better predictions.

3. Proposed Method

Our project aims to build a model with an accuracy greater than 90 percent in order to make predictions on the Vinho Verde Wine's quality, using the data set posted by Shelvi Garg on Kaggle. This data set is a relatively small data set, which contains about 6500 observations and 12 attributes to the wine quality score. The features include wine type and various complex chemical compounds, such as fixed acidity, volatile acidity, and citric acid etc, which we will construct a model based on and make predictions.

Coincidentally, there is a similar data set on Kaggle posted by UCI Machine learning, which contains the about 5000 observations and similar attributes to our data set. This data set makes up the shortcoming of our small data set. Therefore, we intend to train and test our model with the original data set, and finally use our constructed model to make predictions on this new data set.

Firstly, we needed to clean our data set. We found that our data set include some 'NaN' values, which may hinder us from using some machine learning algorithms such as some tree-based methods and random forest algorithm. Therefore, before splitting our data set to training set and test set, our first task was to deal with those missing values. In order to have more data set, we did not choose to drop the rows containing those missing values. Instead, we imputed them with the column mean. Then, we needed to map the quality scores with different labels. Considering that the range of scores 1-10 is too large for our models to predict a specific score very accurately, we divided scores 1-10 into three different quality levels: low quality for score 1-4, medium quality for score 5-7, and high quality for score 8-10. After that, we labelled low quality as 0, medium quality as 1, and high quality as 2 to complete the following steps.

3.1. Feature Extraction and Selection

After cleaning the data set, we decided to do some feature extraction and selection to filter out some insignificant features before training our machine learning models. We found that there was a way for random forest classifier and extra tree classifier to extract the relative importance of different features of them. We planned to list these feature importance by firstly constructing these two types of models with the original clean data set. After extracting the feature relative importance, we found that there was only one feature 'type' having a much less importance, which was less than ≤ 0.01 than all other features, which had a importance around 0.1. Therefore, we decided to drop the column 'type' and construct our machine learning models with the other 11 attributes.

3.2. Model Construction

In our project, we intended to construct multiple types of machine learning models, including k Nearest Neighbors, Decision Tree, Random Forest, Super Vector Machine, Linear Regression, Naive Bayes, Extra Tree, AdaBoost, HistGradientBoosting, and some other ensemble methods such as Bagging and Stacking. Finally, we would like to choose the most accurate one to make predictions on the data set posted by UCI Machine Learning.

3.2.1 Linear Regression

Linear regression is one of the simplest algorithms in machine learning, and it will fit the data with a linear line. It is cheap and can be applied on almost all data sets, including our data set, so it becomes the first model we applied on our data set.

3.2.2 Super Vector Machine

Super vector machine (SVM) is a relatively new classifier in machine learning. One advantage of this method is that this SVM model can avoid the tendency of overfitting in some ways. Basically, there are four useful kernel functions in scikit-learn, which are linear function, polynomial function, radial basis function (RBF), and sigmoid function. In our project, we will directly import 'SVC' fucntion from the package 'sklearn.svm' and build our model use the default radial basis function.

3.2.3 Random Forest

Random forest is a very popular ensemble method in machine learning, which works by fitting a large number of decision tree classifiers on many subsets of the training data set. The two main advantages of random forest is the predictive accuracy and overfitting control. In our project, we import the function 'RandomForestClassifier' from the package 'sklearn.ensemble'. After constructing an initial model, we planned to use a grid search method to find a better parameter. Also, due to its high accuracy, random forest method plays an important role in our other ensemble methods.

3.2.4 K Nearest Neighbors

K Nearest Neighbors(KNN) is the most basic classification algorithm used for both classification and regression, which is cheap as well as easy to understand. This algorithm is relied on different types of distance, but in our project, we will primarily use the Euclidean distance. We construct our

KNN model by importing the function 'KNeighborsClassifier' from the package 'sklearn.neighbors'. After that, we used some grid search methods to figure out a better k for our model in a range between 1 to 15.

3.2.5 Decision Tree

Similar to the K Nearest Neighbors algorithm, decision tree is also a very essential non-parametric supervised learning method for classification and regression. It is always easy to interpret, and the trees can be visualized. Another important advantage of decision tree is that it can handle multiple types of data, no matter numerical or categorical. We use this method by importing the function 'DecisionTreeClassifer' from the package 'sklearn.tree'. After that, we used some grid search methods to tune three parameters, including 'criterion', 'max_depth', and 'min_samples_split'.

3.2.6 Naive Bayes

Naive Bayes methods are supervised learning algorithm based on the Bayes' theorem as below.

$$P(y|x_1, x_2, ...x_n) = \frac{P(y)P(x_1, x_2, ..., x_n)}{P(x_1, x_2, ...x_n)} \tag{1}$$

In our project, we will import a 'GaussianNB' function from the package 'sklearn.naive_bayes', which uses Gaussion Naive Bayes for classification.

3.2.7 Extra Trees

Extra trees method is very similar to random forest, which uses some extra randomized decision trees on smaller subsets of our training data set to have a higher accuracy and control the overfit problems. In our project, we build our model with the function 'ExtraTreesClassifier' from the package 'sklearn.ensemble'. After that, we intended to use a grid search methods to find a better parameter which could maximize the model accuracy.

3.2.8 AdaBoost

The adaptive boosting method is one of the ensemble methods which is based on many other machine learning algorithms. This boosted classifier is adaptive because the misclassification of an earlier weak algorithms may be adjusted by other subsequent weak algorithms. As a result, the model constructed by AdaBoost will be a stronger algorithms finally. In our project, we use this algorithm with the function 'AdaBoostClassifier' from the package 'sklearn.ensemble'.

Method	Accuracy
Random Forest with Grid Search	94.5020 %
Extra Tree with Grid Search	94.4140 %
Random Forest	94.1026 %
Extra Tree	94.1026 %
Hist Gradient Boosting	93.9487 %
Decision Tree with Grid Search	93.2044 %
SVM	93.1795 %
KNN with Grid Search	93.1384 %
KNN	92.5641 %
Ada Boosting	89.4359 %
Decision Tree	89.3846 %
Naive Bayes	83.6410 %
Linear Regression	6.0389 %

Table 1. Our initial model accuracy table.

3.2.9 HistGradientBoosting

The histogram-based gradient boosting classifier is a tree based method and an improvement of gradient boosting classifier. Compared with gradient boosting classifier, it is much faster especially for those large data set. We will construct a hist gradient boosting classifier by importing the function 'HistGradientBoostingClassifier' from the package 'sklearn.ensemble'.

3.2.10 Other Ensemble Method

After constructing the models above, we made our first model accuracy table. We planned to pick up some high-accuracy models and use them to construct some more models with other ensemble methods such as stacking and bagging.

Bagging. Bagging is an ensemble learning method also known as bootstrap aggregation method. In our project, we applied this method by importing the function 'BaggingClassifier' from the package 'sklearn.ensemble'. We used random forest classifier as the base estimator, since it is one of the models with the highest accuracy (as shown in the Table 1).

Stacking. Stacking is an ensemble learning method which combines the predictions from multiple well-performed base estimators. In this way, models constructed by stacking can have a better performance and avoid overfitting in some way. In our project, we use two stacking classifier, which are stacking classifier and stacking CV classifier. We use logistic regression as a meta estimator because it is a relatively fast algorithm, and we use the top 4 models in our initial model accuracy table as base estimators, including

Decision Tree after grid search, Random Forest, HistGradientBoosting, as well as Extra Tree (as shown in the Table 1).

4. Experiments

4.1. Dataset

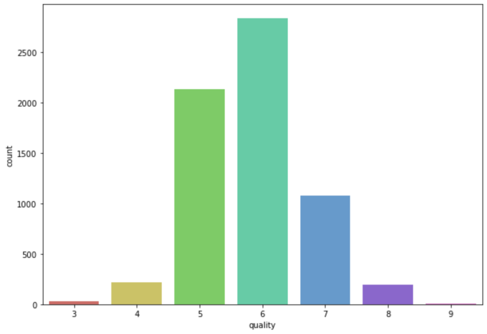

Figure 1. A visualization of our data set used from training and testing models.

The datasets we obtained are from UCI database, which is also published on Kaggle after some transforming. There are total 13 columns and 6497 rows in this data. For another dataset, which is very similar to our first one, only contain the data related to one type of wine-red wine. It has total 13 columns with 1598 rows. The variables included in the dataset are:

- fixed acidity: most acids involved with wine or fixed or nonvolatile(do not evaporate readily)
- volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste)
- citric acid: found in small quantities citric acid can add ’freshness’ and flavor to wines)
- residual sugar: the amount of sugar remaining after fermentation stops, it’s rare to find wines with less than 1 gram/liter)
- chlorides: the amount of salt in the wine)
- free sulfur dioxide: the free sorm of SO2 exists in equiliburium between molecular SO2(as a dissolved gas) and bisulfite ion)
- total sulfur dioxide: amount of free and bound forms of SO2)
- density: the density of water is close to that if water depending on the percent alcohol and sugar content)
- 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)
- sulphates: a wine additive which can contribute to sulfur dioxide gas(SO2)levels)
- alcohol: alcohol concentration of the wine..
- quality: Wine’s quality rank from 1 to 10. As we find that there are 10 levels in our response variable-quality. We decided to simple it to three level which are high, medium, and low.

4.2. Software

The computer software would be mainly Python through jupyter notebook and we used some packages like Scikit-Learn, XGBoost, some decision tree models, and so on.

4.3. Hardware

The computer hardware would be each group members’ laptop(CPU).

5. Results and Discussion

We mainly use the model’s built-in score funtion to measure our model’s accuracy. Our dataset has 12 columns after dropping ‘type‘ from it. For any missing data, we fill it with the mean value of that column to ensure the data size.

5.1. Overall Model Accuracy

Table 2 is a summary of the final results obtained from each model. The accuracy of models range from over 94 percent to 6 percent. Most of the test accuracy are above 90

Method	Accuracy
Random Forest with Grid Search	94.5020 %
Extra Tree with Grid Search	94.4140 %
Random Forest	94.1026 %
Extra Tree	94.1026 %
Stacking CV	94.0513 %
Hist Gradient Boosting	93.9487 %
Stacking	93.9487 %
Bagging	93.5897 %
Decision Tree with Grid Search	93.2044 %
SVM	93.1795 %
KNN with Grid Search	93.1384 %
KNN	92.5641 %
Ada Boosting	89.4359 %
Decision Tree	89.3846 %
Naive Bayes	83.6410 %
Linear Regression	6.0389 %

Table 2. This is a model selection table.

percent. This strongly represents physicochemical test data can predict wine quality with reliable results. It further implies there do have a physicochemical standard for judging the quality of wine. This result may provide wine manufacturers with better idea on wine production and inspire wine merchants of positioning wine based on their quality.

5.2. Random Forest

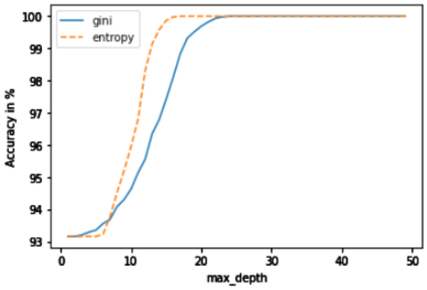

Figure 2. A figure visualize the parameter selection of Random Foerest Classifier.

Our initial Random Forest model gives us an accuracy around 94.10%. In this case, we did not adjust any parameter and just used the default. After seeing the high accuracy, we decided to adjust its parameter to achieve a better performance of our model. Then, we imported the function 'GridSearchCV' from the package 'sklearn.model_selection'. In the grid search, we defined 'criterion' as 'gini' and 'entropy' and limited 'max_depth' in a range between 1 and 50. Finally, we get a best parameter combination, which is 'gini' criterion and 50 max_depth. In this case, we get an accuracy around 94.41%. Here is a plot to better visualize the parameter selection of Random Forest Classifier. From this figure, we can find that 'gini' and 'entropy' tend to have similar effect after max_depth over 20, and criterion 'gini' and max_depth '50' can be one of the parameter combinations with the highest model accuracy.

5.3. Extra Tree

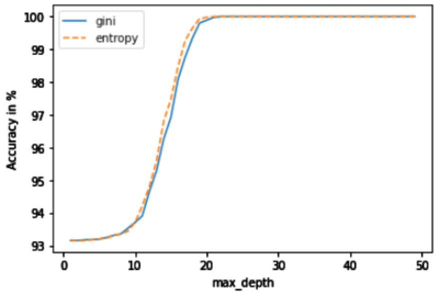

Figure 3. A figure visualize the parameter selection of Extra Tree Classifier.

Surprisingly, we found the Extra Tree model gives us the same accuracy as the Random Forest one at first, which is around 94.10%. We believed this is because the core principal of both algorithms are decision trees on multiple subsets of the training set. We also conducted a grid search for Extra Tree classifier, defining 'criterion' as 'gini' and 'entropy' and limiting 'max_depth' within the range between 1 and 50. In this time, we found the best parameter combination is the criterion 'gini' and the max_depth 20. The result, which is around 94.50%, is slightly greater than the Random Forest Classifier. As in the figure above, though the overall trend is similar to the Random Forest classifier and the accuracy will reach the highest when max_depth is around 20, the two types of criterion do not differ a lot from each other. It is plausible that criterion 'gini' and max_depth '20' can be one of the parameter combinations with the highest model accuracy.

5.4. K Nearest Neighbors

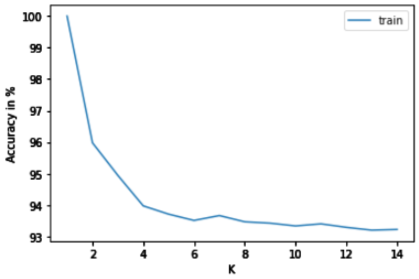

Figure 4. A figure visualize the parameter selection of KNN Classifier.

For the KNN classifier, the only parameter that we needed to tune is the number of neighbors. Initially, we defined the parameter as 'n_neighbors' is 5, and we got a model with an accuracy around 92.56%. Then, we conducted a grid search by defining this number of neighbors within the range between 1 and 15, and finally we have the best accuracy when 'n_neighbors' is 9, which is around 93.13%. In the figure above, there is a decreasing trend when k is increasing. We believed that though a small k-value may increase the model accuracy, it tends to have an overfit problem. Therefore, we got a best test accuracy when 'n_neighbors' is 9.

5.5. Decision Tree

For our initial decision tree model, we have 89.3846% accuracy on the test data. After we did the grid search on it, we find the best parameters are {'criterion': 'gini', 'max_depth': 3, 'min_samples_split': 10}. And we got a quite improved accuracy: 93.2044%. As we can see from above figures, as parameter max_depth increase, the accuracy also increase. And entropy criterion increase more

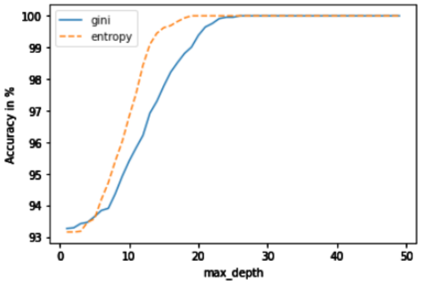

Figure 5. A figure visualize the parameter selection of Decision Tree Classifier.

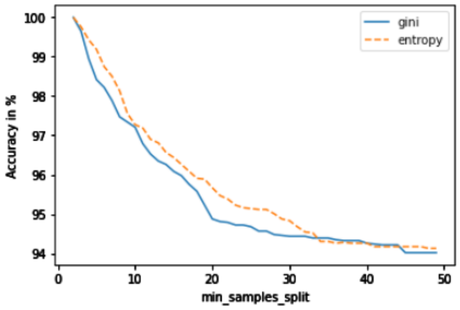

Figure 6. A figure visualize the parameter selection of Decision Tree Classifier.

rapid than gini. For min_sample_split, the accuracy decrease as it increase. And the trend for both gini and entropy criterion is quite the same.

5.6. Linear Regression

Linear Regression has the lowest accuracy as 6.0389 %. As one of the simplest algorithms in machine-learning, linear regression assumes a linear relationship between the input and output varaibles. Therefore, it fails to fit complex datasets properly. Here, we are searching how the quality of wine are connected with other eleven variables. Hence a straight line doesn't fit the data properly.

5.7. Further prediction

As we know random forest model and extra tree model both have the highest score among other methods. We apply this two models we trained to test on the smaller dataset which we mentioned in the dataset section: it has only one type of wine with around 1598 rows. The test accuracy is 98.31 % for randomforest and 98.50 % for extra tree method. In that case, we can say that extra tree model is the best model to predict the wine quality.

5.8. Feature Importance

In addition to make better predictions, we also want to find out which feature is relative more important in our predicting. We conduct feature importance analysis with both random forest and extra tree model which result in the highest test accuracy before.

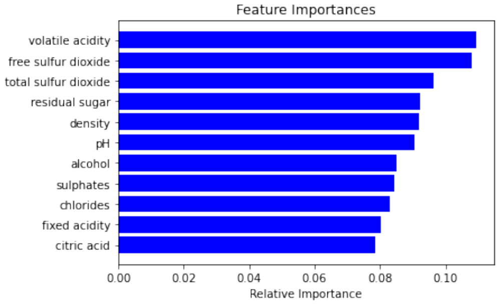

Figure 7. Feature importance by using random forest

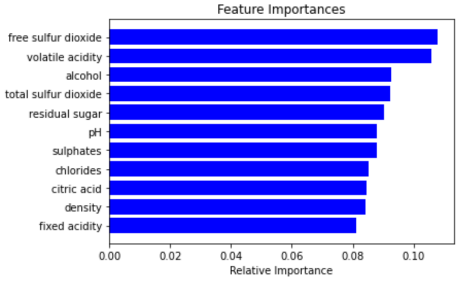

Figure 8. Feature importance by using extra tree

Figure 7 is the feature importance by using random forest. In this plot, we can find the most important feature in predicting the wine quality are volatile acidity, free sulfur dioxide, total sulfur dioxide, residual sugar, density, and pH. All these features may be reasonable for being important in predicting wine quality except density. As we know, the density of wine also depends on the alcohol and sugar content. But why density plays a more important role than other features here.

We can check feature importance with extra tree to strengthen our analysis and solve our doubts. Figure 8 is the feature importance by using extra tree. In this plot, we can find the most important feature in predicting the wine quality are free sulfur dioxide, volatile acidity, alcohol, total sulfur dioxide, residual sugar, and pH. Different from previous analysis with random forest, here alcohol becomes the one of the most important features, and density drops to one of lowest places. Given density is correlated to alcohol, this somehow represents these features are not as important as other features like free sulfur dioxide.

By comparing to the feature importance with random forest, we find out free sulfur dioxide, volatile acidity, total sulfur dioxide, and pH play the most important role in both models. It can imply that the free sorm of SO2 exists in equiliburium between molecular SO2, the amount of free and bound forms of SO2, the amount of acetic acid in wine, and the pH value of wine indicate its specific quality.

6. Conclusions

In our wine quality prediction problem, we mainly conduct the task to predict the wine's quality. We utilize 14 methods to fit the data and we find two of them have the highest teat accuracy among others: Random Forest and Extra Tree.

Based on the result from our second test on the smaller dataset, Extra Tree has the best overall performance. Besides, method Stacking CV also had a quite decent accuracy which is expected. We foresee there would be a quite low score for linear regression method, but it's still quite surprise when we find that the linear regression model had such a low accuracy.

One thing we could improve in the future is that to find more data, since the wine quality data point are unequally distributed. Also, since we apply that much method in our project, we relatively ignore the hyperparameters tuning step. So maybe we can do more hyperparameters tuning in the future.

7. Acknowledgements

We gratefully appreciate the help and guidance from our professor Sebastian Raschka.

8. Contributions

Shihao found the original dataset from the web and Yifan found the second data which we used to test on model. Shihao also did the data cleaning and feature selection steps. Kexiao and Yifan together implemented the different method to fit a model on data. Besides, Kexiao and Yifan also conduct several grid search on different models to improve the performance of the model. All team members, Shihao, Yifan, and Kexiao, worked together to finished the project report.

References

[1] Vinho verde – grape, wine, wine style or wine region? 2019.

[2] Y. Gupta. Selection of important features and predicting wine quality using machine learning techniques. *Procedia Computer Science*, 125:305–312, 2017.

[3] F. A. T. M. J. R. Paulo Cortez, Cerdeira António. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

9. Github link

https://github.com/Hawk9808/STAT-451-project-group