The Best Wine

Kuan-Ying Wang
Computer Science
UC Santa Cruz
Santa Cruz, CA, US
kwang36@ucsc.edu

Yu-Shuo Li
Computer Science
UC Santa Cruz
Santa Cruz, CA, US
yli186@ucsc.edu

ABSTRACT

We propose a model that is able to classify wine quality with high accuracy given its physicochemical values as input.

We did a series of data preparation and feature engineering before we apply, compare, and optimize various models and algorithms to obtain the optimal classification accuracy. The models tested are linear regression, gradient descent, logistic regression, neural network, random forest, and gradient boosting classifier.

Our test results show that random forest and gradient boosting classifier outperform the other models and are able to achieve beyond 70% classification accuracy.

1. Motivation and Objective

Machine learning in the food and drink industry has been focused on producing the food and cutting the cost down. But when it comes to drinks and food like wine or cheese, it's more about the taste. Therefore, a useful application will be a model that could accurately classify food and drink qualities based on their physicochemical inputs, the model that people can rely on.

Our goal is to find out the best machine learning model and algorithm for classifying the quality of red and white wine given their physicochemical input values.

We are treating our dataset as a classification problem as opposed to a regression problem that other people may see it as. Unlike other researchers that uses single-model approach, we plan on achieving it differently. We did what other approaches lacked, that is, to compare different machine learning algorithms as well as implement them in various machine learning models. By optimizing the best model, we can achieve better classification accuracy.

During the process, we are also figuring out the physicochemical inputs that helped determine the quality of the wines. That is, which physicochemical inputs are better left out and which are good to keep when it comes to determining the quality of red and white wine.

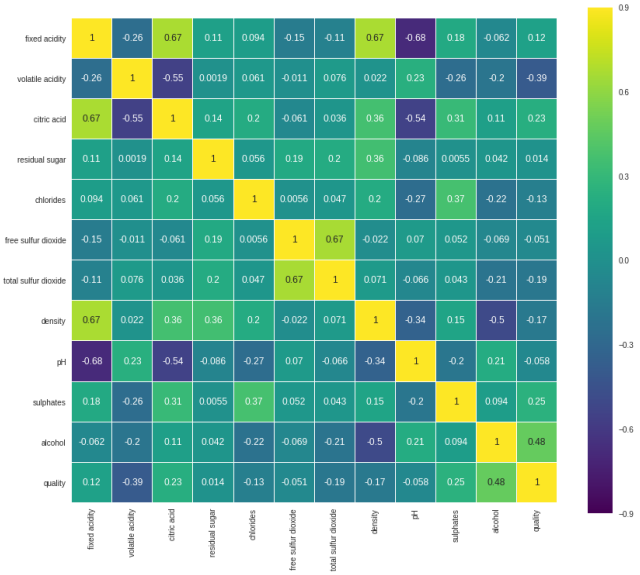
In addition to model implementation, by comparing different algorithms and models, we can observe the characteristic of each model and further understand their application and usage. Furthermore, we also plan on trying different methods that could decrease the noise in the data and lower the confusion the data may bring to our training model.

2. Dataset

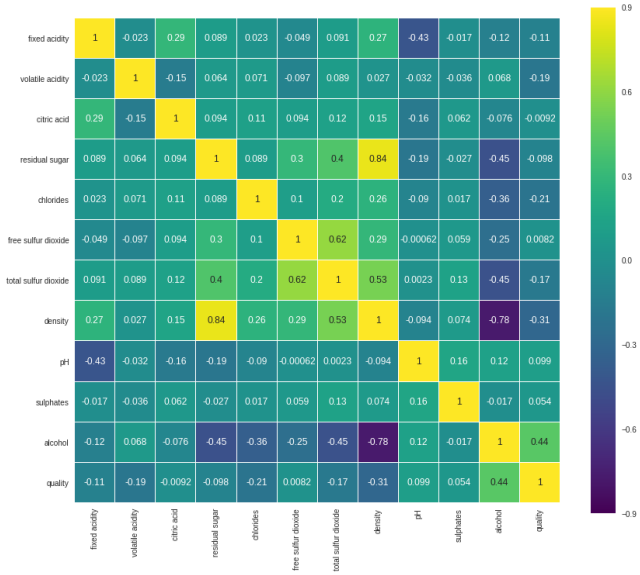
We obtain our dataset from the UCI machine learning repository. The Wine Quality Dataset contains 12 columns: 11 physicochemical values as input and 1 quality value as output. The physicochemical inputs are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates, and alcohol. The dataset is divided into Red Wine and White Wine. Red wine dataset contains 1599 samples. White wine dataset contains 4898 samples.

2.1 Remove Unnecessary Features

We first look at the pairwise correlations between the variables.



(a) Pairwise correlations of red wine variables



(b) Pairwise correlations of white wine variables

Figure 1: (a) The red wine dataset does not have any pairs of variables that are highly correlated, but it does show that variables free sulfur dioxide, residual sugar, and pH's low correlation with quality. (b) The white wine dataset has a pair of highly correlated variables. It also shows citric acid and free sulfur dioxides' low correlation with quality.

As shown in figure 1(a), residual sugar, free sulfur dioxide, and pH have the following insignificant correlations with quality: 0.014, -0.051, and -0.054. White wine dataset, on the other hand, has a different correlation table. As seen in figure 1(b), citric acid and sulfur dioxide as <1% correlation with quality, -0.0092 and 0.0082 respectively.

These variables are best left out in the training process as they would most likely add noise to our models, which is proven to be true after testing. After eliminating these variables, most models' accuracy improved. However, surprisingly, removing one of the the highly correlated-pair in white wine, namely, density and residual sugar, did not improve the accuracies of our final models, thus both features are kept.

2.2 Eliminate Outliers

We then proceed to eliminate the outliers in our dataset by plotting variables that have higher correlations with quality against each other. For red wine, variables density, alcohol, and volatile acidity are plotted. For white wine, variables density, alcohol, and chlorides are plotted against each other to spot the outliers.

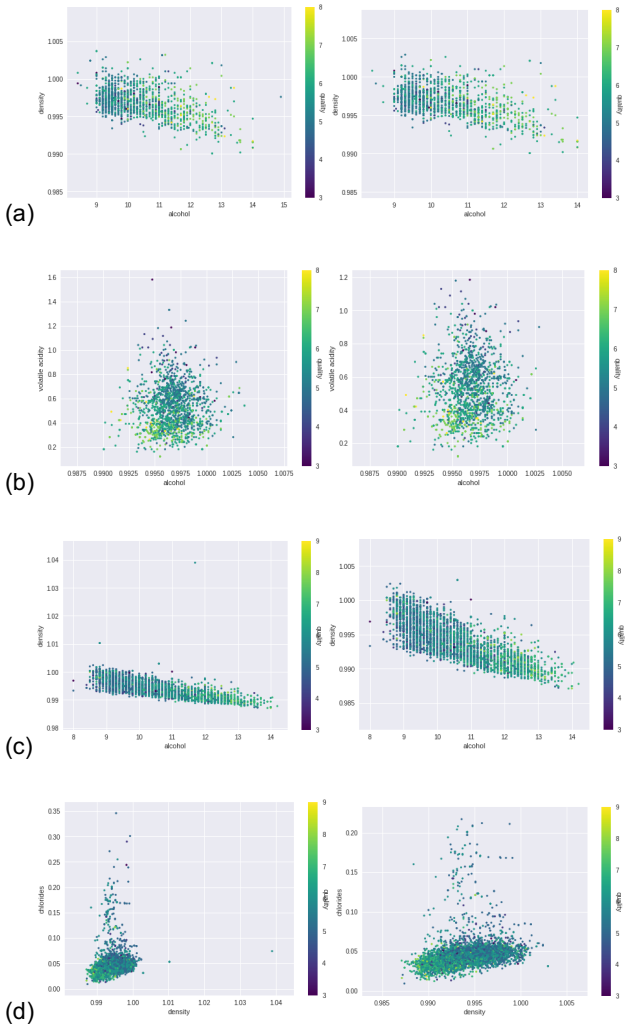


Figure 2: Plots of feature variables before and after deleting outliers. (a) Red Wine: Density vs alcohol plot (b) Red Wine: Volatile acidity vs alcohol plot. (c) White Wine: density vs alcohol plot. (d) White Wine: density vs chlorides

As seen in figure 2, the samples in both dataset are not easily separable, but the outliers are obvious as they are located far from mean value and could potentially add unwanted noise to our models. It is better to delete them before training the model. Even though the dataset could use more outlier deletion, but deleting too many samples could lead to our model losing generosity and should be avoided.

2.3 Data Imbalance Problem / Oversampling

Since we are classifying wine quality, we need to look at the distribution of quality in both datasets.

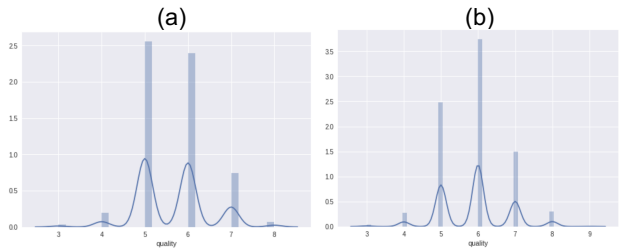


Figure 2: (a) The red wine dataset contains red wines of quality ranging from 3 to 8, with the majority of samples being quality 5 and 6. (b) The white wine dataset contains white wines of quality ranging from 3 to 9, with the majority of samples being quality 5 and 6.

As shown in figure 2, our dataset is strongly imbalanced. To address the problem, we need to oversample our data. But with only a handful of quality 3, 4, 8, and 9, the oversampled data would only add noise to our data because the wines with those qualities are too sparse, the oversampler won't be able to provide accurate, useful samples. Therefore our first step is to categorize and merge the wines into 3 quality classes: *Low*, *Medium*, and *High* so we have enough data of each class for the oversampler to be able to make the distinction between the classes. That said, wines with quality in the range of 1 to 5 are categorized as low-quality, 6 as medium-quality, and 7 to 10 as high-quality.

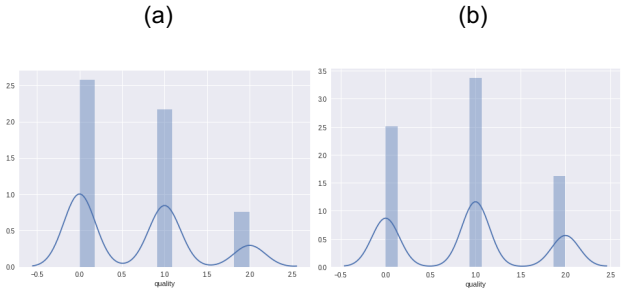


Figure 3: (a) Quality of red wine samples after the merge. After class merge on the red wine(b) Quality of white wine samples after the merge.

As shown in figure 3, we have denoted Low, Med, High quality as integer value 0, 1, and 2 for simplicity as well as making one-hot-encoding of our neural network model easier.

We now have enough data for the random oversampler. We then applied it to our training set so the model will not have biases over certain wine qualities. We avoided the mistake of oversampling the testing set, which could lead to leakage of testing set information.

We only applied oversampling to the minority classes. That way we don't create too much noise to our model as the non-minority classes already have a sufficient number of samples.

2.4 Feature Scaling

Our last step is to normalize our data without losing any information. We z-score transform all the features for a number of benefits: faster computation, better error shape, and to prevent certain features' domination and biases.

3. Models and Algorithms

We applied different regression and classification models to compare their accuracies and further optimize the models with better initial performance. All models are trained with 70/30 training and testing split. There's an increase of testing set from 80/20 split because of data imbalance and we want the test set to have sufficient minority-class samples.

3.1 Regression Models

The regression models tested are linear regression and gradient descent. Linear Regression is the model that we are first introduced to. Being the simplest regression model out there, we are curious to see its result on our training data.

Gradient Descent is better for learning multi-feature problem when compared to linear regression. We use the loss function $\frac{1}{2n}\sum_{i=1}^{n}(y_i - c^T x_i + b)^2$ to calculate the gradient for each of the nine features. We used a learning rate of 0.03 to adjust feature weights in each iteration. We do it for 100 epochs to make sure the model converges to a local minimum.

3.2 Classification Models

Classification models are expected to work better with our data since our goal is to classify wine quality.

3.2.1 Logistic Regression

Logistic regression is the first classification model that we are introduced to.

3.2.2 Neural Network

Neural network is the model that we spent most of our time on. It takes the wine's physicochemical value as input and classifies its quality according to the physicochemical input values.

Our initial setup was three hidden layers with 400, 200, and 100 neurons. We used sigmoid as our initial activation functions, softmax as the output layer, and categorical cross-entropy as our loss function.

We observe the plot of training loss vs validation loss and the plot of training accuracy vs validation accuracy to analyze our model's performance as they are useful to see if the model is over fitting. As shown in figure 4, there is no sign of over fitting after adjusting to the right dropout rate.

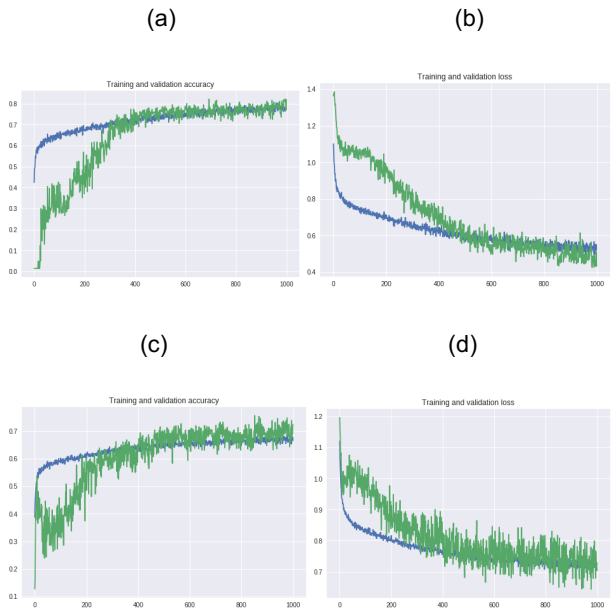


Figure 4: Blue = training, green = validation (a) Training and validation accuracy of the neural network for the red wine dataset. (b) Training and validation loss of the neural network for the red wine dataset. (c) Training and validation accuracy of the neural network for the white wine dataset. (d) Training and validation loss of the neural network for the white wine dataset.

Lots of modifications been made to the network. We have learned from trial and error that it is better to train a simpler network for more epochs than to train a complex neural network with fewer epochs. With a complex neural network that has hundreds of neurons in each layer, the model is prone to over fitting on the training set. To avoid vanishing gradient in our neural network, we have changed the hidden layers' activation function to reLU. We also added dropout to our model. Starting from 0.1 and slowly adjust it together with the number of neuron in each layer to obtain optimal accuracy. We also learned that increasing dropout rate may improve over fitting but it may also lead to a decrease in training accuracy as there may not be enough neurons to learn all the features.

We have learned from the lecture that mini-batching can improve performance. However, with our training data being oversampled, when the batch is too small, we cannot guarantee whether each batch contains an accurate representation of each sample classes. That is the tradeoff we have to make. For optimal accuracy, we test the model with several training sets and validation sets, from 50/50 to 90/10.

Our final neural network model consists of an 8-input input layer (9 for white wine), three (four for white wine) hidden layers, each with 100 neurons and 50% dropout rate. We used reLU as our activation function and softmax as the output layer. The validation split is 75% for training and 25% for validation. The loss function is the same as our initial model, categorical cross entropy, as it is best for calculating loss for classification problem in the sense that we need to adjust the probability of each class according to all softmax values of each category. Since we have a different number of samples in red and white wine, their batch size is different. We use 100-sample batch for red wine and 210-sample batch for white wine. Both models are set to iterate for 1000 epochs.

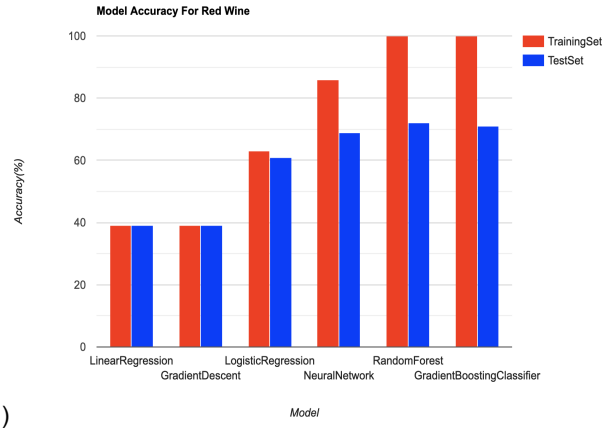
3.2.3 Random Forest Classifier

Random forest was suggested by our TA, Rafael when we can no longer improve accuracy on our neural network model. It is a decision tree based model. Surprisingly, the initial test output of our random forest model is better than the output of our 'optimized' neural network model. It passes 70% accuracy on both red and white wine dataset, which we have failed to do so with our neural network model. We set the number of estimator to 100 as adjusting it doesn't seem to have an effect on the accuracy.

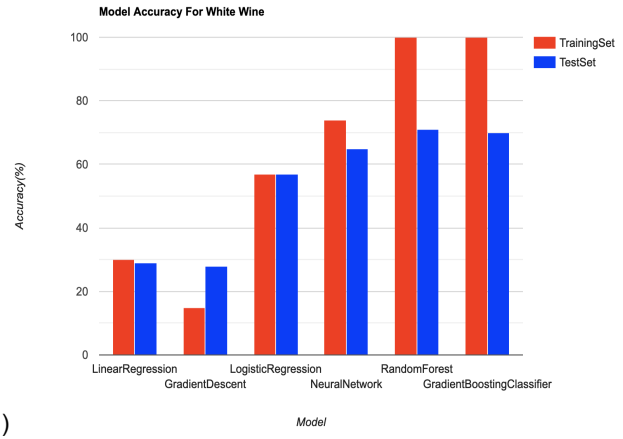
3.2.4 Gradient Boosting Classifier

Gradient Boosting classifier is a model that I came across while researching about the random forest. It is also a decision tree based model. The model surpasses 70% accuracy on both datasets. After trials and errors, the final number of estimator is set to 100, learning rate is set to 0.03, loss function defaults to logistic regression for classification probabilistic output. We also test the model with different values of max_depth to see how the features interact with each other and decided to settle with 11. This model also comes with feature_importance function that allow us to check each features' contribution to the classification result, details are in the analysis section.

4. Results and Analysis



(a)



(b)

Figure 5: (a) Test Set Accuracy for Red Wine: Linear Regression: 39%, Gradient Descent 39%, Logistic Regression:61%, Neural Network 69%, Random Forest: 71%, Gradient Boosting Classifier: 70% (b) Test Set Accuracy for White Wine: Linear Regression: 29%, Gradient Descent 28%, Logistic Regression:57%, Neural Network 65%, Random Forest: 71%, Gradient Boosting Classifier: 70%.

Oversampling did not increase accuracy on all models. In fact, using resampled training data to train regression models like linear regression and gradient descent decreases their accuracies, roughly around 5-10% for each model. In addition to regression models, it also decreases the accuracy of logistic regression. Therefore, we use the original un-sampled training data to train those regression models.

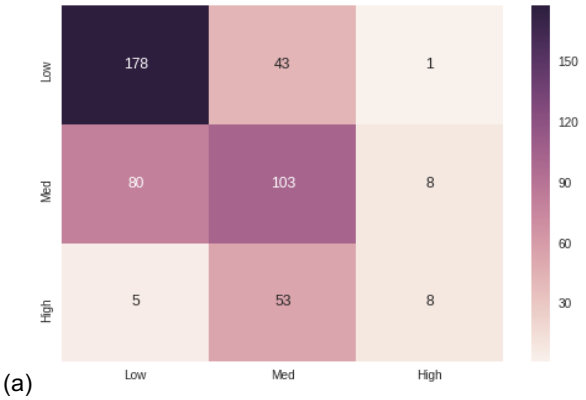
4.1 Results of Linear Regression and Gradient Descent

Linear regression has testing accuracy of 39% and 29% for the red and white wine dataset respectively. It is clear to us that this dataset required a more complex model than linear regression. It's the simplest regression model after all. This proves that both the red wine and white wine datasets are not linearly separable. The coefficient and intercept for the linear regression models did not provide much help due to the low accuracy of the models.

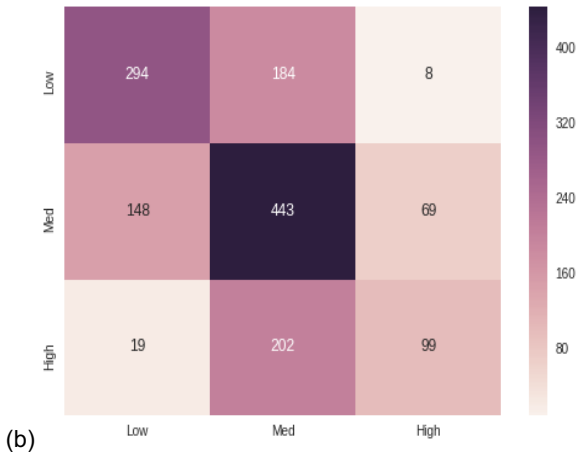
Gradient descent as shown in figure 5, has accuracies of 39% and 28% on the test set for red and white wine dataset respectively. Notice that the accuracy result of gradient descent is almost the same as the results from linear regression. We had initially expected it to perform better than linear regression but the result came clear when we recalled the number of features in the data, which is 9 dimensional. With the nature of our data not being easily separable, the gradient descent algorithm could not perform as well as expected.

4.2 Results of Logistic Regression

The accuracy of logistic regression on the test sets is 61% for the red wine dataset and 57% for the white wine dataset. This accuracy is significantly better than the results of linear regression and gradient descent. Since the outcome of logistic regression is bounded as opposed to linear regression and gradient descent that can go infinitely, it's more suited for classification problem like this one. The reason it is not performing as good as the models that will be mentioned later is that we have a multi-class classification. It would be harder for logistic regression to classify multi-class problems when the output of logistic function is binary.



(a)



(b)

Figure 6: (a) Confusion matrix of logistic regression for the red wine dataset. (b) Confusion matrix of logistic regression for the white wine dataset.

We can clearly see from figure 6 that logistic regression had trouble identify high-quality wine. In fact, for red wine dataset, it only correctly identified 8 high-quality wines out of 66 in the test set. For white wine, it's 99 out of 320. However, it did correctly classify the majority of low and

medium quality wines, false positive and false negative rates are too high.

4.3 Results of Neural Network

The accuracy of the neural network on the test set is 69% for the red wine dataset and 65% for the white wine dataset. The neural network has noticeable improvements on the accuracies when compared to logistic regression. Having implemented softmax as the output layer, the model is able to classify the quality of the wines by choosing the quality class that the wine has a higher percentage of being in.

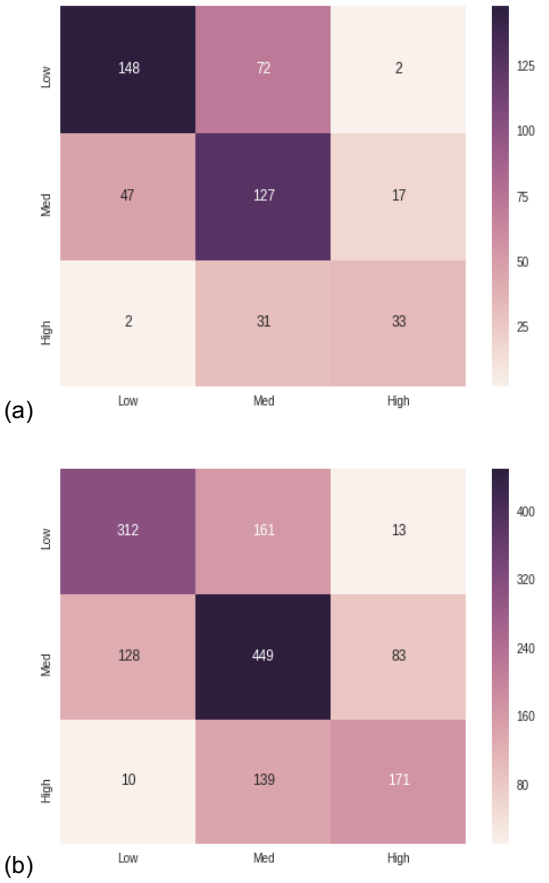


Figure 7: (a) Confusion matrix of the neural network for the red wine dataset. (b) Confusion matrix of the neural network for the white wine dataset.

As shown in figure 7, our neural network model was able to predict >50% of high-quality wines for both datasets. An improvement from logistic regression. However, with oversampled data, there is an increase in the number of low-quality wines being predicted as high quality when compared to logistic regression, which uses original, un-sampled training data. Overall, neural network has better performance with the oversampled data and outperforms logistic regression.

4.4 Results of Random Forest

As shown in figure 8, the number of correct classifications in each class is greater than that of the neural network. Even with 70% accuracy, some samples are still categorized incorrectly as samples in our dataset are not easily distinguishable.

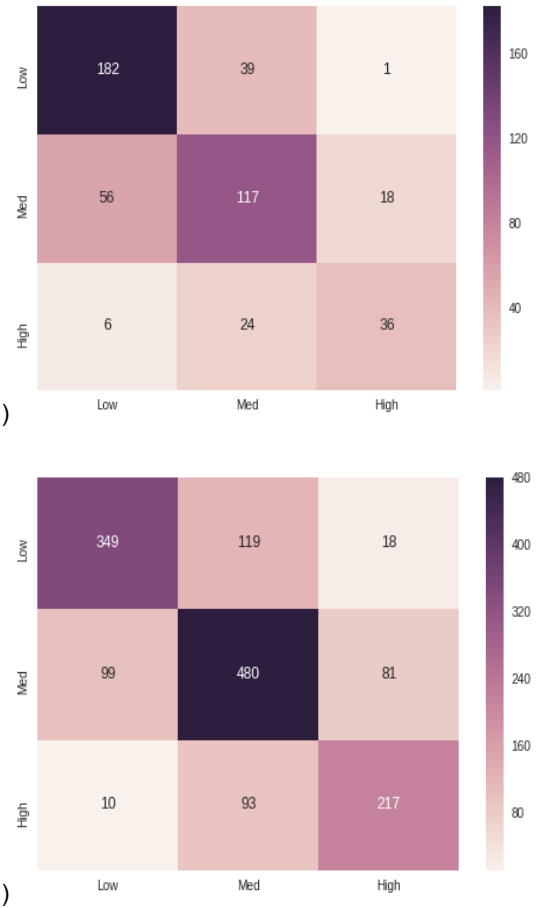


Figure 8: (a) Confusion matrix of random forest for the red wine dataset. (b) Confusion matrix of random forest for the white wine dataset.

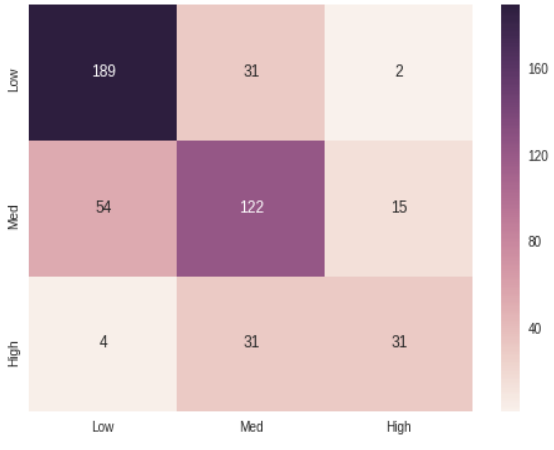
4.5 Results of Gradient Boosting Classifier

The results are similar to the results of random forest, with 70% accuracy and the majority of classification output being correct. The reason of similar classification output could be the fact that both models are decision tree based ensemble models.

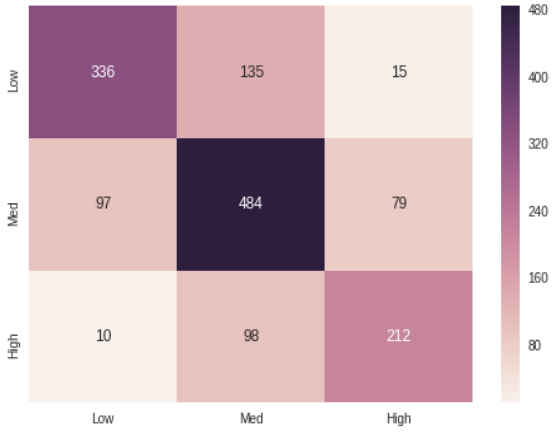
Feature importance of red wine: 0.09270747, 0.13422102, 0.09881626, 0.09312347, 0.12029542, 0.10621629, 0.13582678, 0.21879328.

Feature importance of white wine: 0.08356038, 0.1193922, 0.10465261, 0.09532569, 0.10787762, 0.14150165, 0.09611529, 0.08632171, 0.16525285.

We can see that each feature contributed a fair percentage to the model's classification result, which is good and we don't need to further eliminate any feature.



(a)



(b)

Figure 9: (a) Confusion matrix of gradient boosting classifier for the red wine dataset. (b) Confusion matrix of gradient boosting classifier for the white wine dataset.

5. Conclusion

The samples in our dataset are not easily distinguishable as each wine contains a unique set of physicochemical inputs and that's why wine tasting can be considered as an art.

In this paper, we have compared and optimized several machine learning algorithms and models to classify wine quality. The effort we put into data preparation including data cleaning and feature engineering has a significant impact on our models' accuracy. As a result, we are able to reach 70% accuracy for classifying both red and white wine quality given their physicochemical inputs.

The models' performance on our wine dataset is ranked as follows: Linear Regression ~= Gradient Descent < Logistic Regression < Neural Network < Random Forest ~= Gradient Boosting Classifier. It is certain that there's a correlation between the machine learning model's performance and the types of the dataset. In this case of wine physicochemical inputs and quality, decision tree based ensemble models outperform the other models.

6. Contribution

Kuan-Ying Wang: Data cleaning, feature engineering, implement, test, and analyze linear regression, gradient descent, neural network, and gradient boosting classifier, and work on neural network optimization.

Yu-Shuo Li: Project idea, dataset finding, environment set up, data loading, implement, test, and analyze logistic regression and random forest model, help test and analyze neural network, and work on neural network optimization.

7. Future Work

The dataset we have is relatively small and fairly outdated when we found them in the Irvine database. Therefore, it would be practical to have a more recent and bigger database. Having a bigger database will benefit our accuracy greatly, especially when it comes to training model like neural network. One other concern is the nature of biases in the quality parameter in our dataset. Different wine may appeal to people differently, so it is important to control the biases and quality of the dataset we have. In sum, to obtain a larger and bias-free dataset may greatly increase the accuracy of our model in the future.

Ensemble learning is an area that is not taught in our lecture. Different models learn data features differently, so it makes sense that combining them could result in higher accuracy. In the future, we would like to further explore the concept of ensemble learning as well as its application in different machine learning models.

ACKNOWLEDGMENTS

We thank professor Norouzi for introducing useful machine learning algorithms, models, and techniques, TA Rafael for suggesting the random forest algorithm.

REFERENCES

[1] Eulogio, R. (2017). *Introduction to Random Forests*. [online] Datascience.com.

[2] G. Lemaitre, F. Nogueira, D. Oliveira, C. Aridas (2017). *imblearn.over_sampling.RandomOverSampler — imbalanced-learn 0.4.3 documentation*.

[3] Scikit-learn.org. (2018). *3.2.4.3.5. sklearn.ensemble.GradientBoostingClassifier — scikit-learn 0.20.3 documentation*.