DS 340W: Lab Assignment 1

1. Introduction

In this lab, I am aiming to build a decision tree classifier and a random forest classifier to predict the qualities of wines. A good prediction could help winemaker to have better products, and then increase the price of wines. The dataset we use with 4,898 samples were collected from the northwest region of Portugal (Cortez et al., 2009). This dataset includes seven different class labels, which represent the qualities from 3 to 9. The class labels from 3 to 9 have the following numbers of samples: 20, 163, 1457, 2198, 880, 175, 5, respectively. For each data sample, it has 11 features, and they are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol.

1. Method

In order to measure the goodness of a split, we have to involve the criterion to evaluate it. Two common criteria are the Gini index for the Gini impurity and Entropy for the information gain, and we are using the Gini to measure the goodness of a split in this lab. Since we are running the tree algorithms in this lab, this kind of algorithms needs stopping criteria for growing the tree; possible stopping criteria are max\_depth (the maximum depth of the tree), min\_samples\_split (the minimum number of samples required to split an internal node), min\_samples\_leaf (the minimum number of samples required to be at a leaf node), min\_impurity\_decrease (a threshold that a node will be split if this split induces a decrease of the impurity great than or equal to this threshold), and max\_leaf\_nodes (the maximum number of leaf nodes). In this lab, we only set the value of max\_depth, but min\_samples\_leaf, min\_samples\_split, and min\_impurity\_decrease are also being used as their default values, For example, min\_samples\_leaf = 1, min\_samples\_split = 2, min\_impurity\_decrease = 0.0.

In order to improve the diversity of the fitted tree, the random forest algorithms will generate a lot of decision trees, and each tree has a different combination of features; thus, this technique increased the diversity of the fitted tree. In Random Forest Classifier, the parameter n\_estimators used to control the level of randomness for this technique. Another technique used to improve the diversity of the fitted tree is the put randomness into in the classifier, which controls bootstrapping of the samples and the sampling of the features. In Random Forest Classifier, the parameter random\_state used to control this kind of randomness.

For each tree in a random forest, it used the following default stopping criterions for growing the tree: min\_samples\_leaf = 1, min\_samples\_split = 2, and min\_impurity\_decrease = 0.0.

1. Experiment

In this lab, we split the entire dataset randomly into 80% training and 20% testing, and then we have 3918 training samples and 980 testing samples.

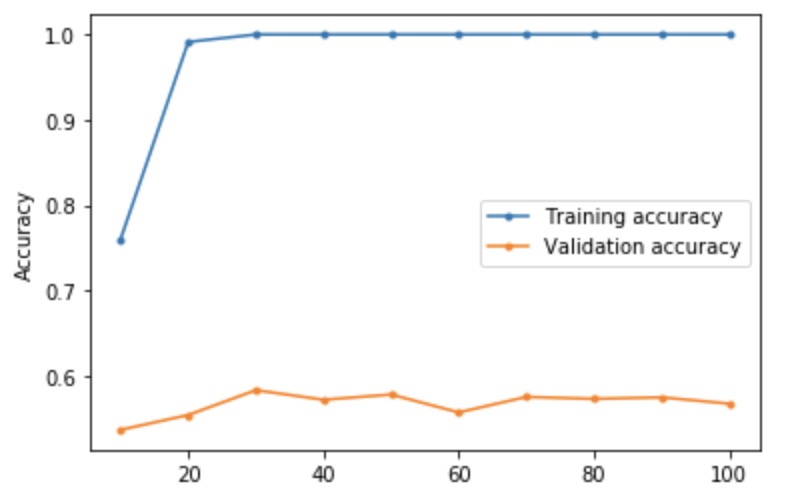


Figure 1: The Plot of Decision Tree Classifier

Based on the result of Figure 1, max\_depth = 30 had the highest validation accuracy. The insights I observed from this plot are (1) a value of max\_depth that has high train accuracy does not mean it has high validation accuracy at the same time. For example, max\_depth = 60 has 100% accuracy on training data, but its validation accuracy is relatively low. (2) There is a positive correlation between training accuracy and the values of max\_depth. In other words, if the values of the max\_depth increase, then the training accuracies would also increase.

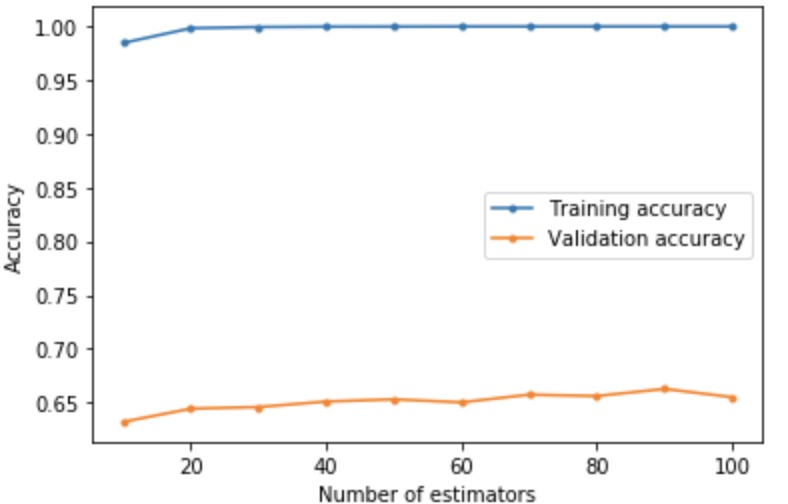


Figure 2: The Plot of Random Forest Classifier

Based on the result of Figure 2, n\_estimators = 90 had the highest validation accuracy. The insights I observed from this plot are (1) Unlike the Figure 1, the overall trends of validation accuracy in Figure 2 went up when the number of estimators was increasing (2) There is a positive correlation between training accuracy and the values of n\_estimators.

I found that max\_depth = 30 and n\_estimators = 90 are the best values for each classifier during parameter tuning. After this process, I trained both classifiers on the entire training set and applied them to the test set. Finally, I got 63% test accuracy from the decision tree classifier and 70% test accuracy from random forest classifier. Based on the above results, we can say that the random forest classifier has better performance on this task. Below are the plots of the confusion matrix of each classifier.

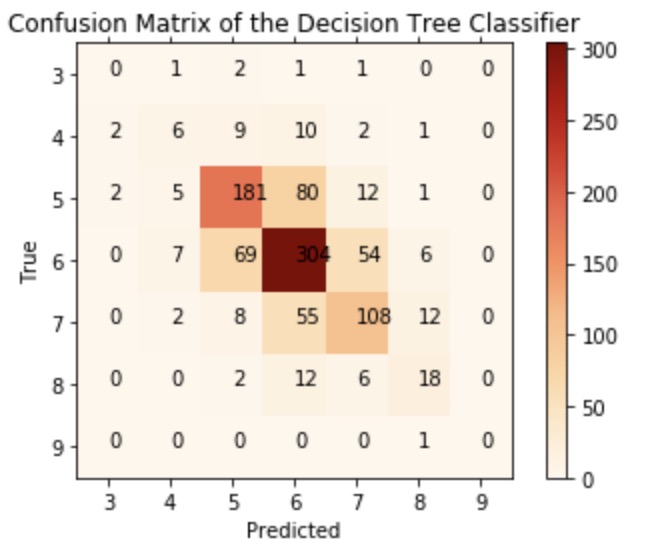


Figure 3: The Plot of Confusion Matrix of the Decision Tree Classifier

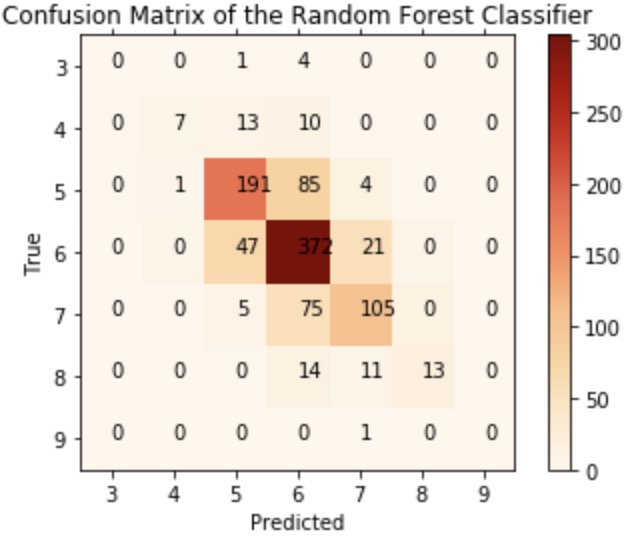


Figure 4: The Plot of Confusion Matrix of the Random Forest Classifier

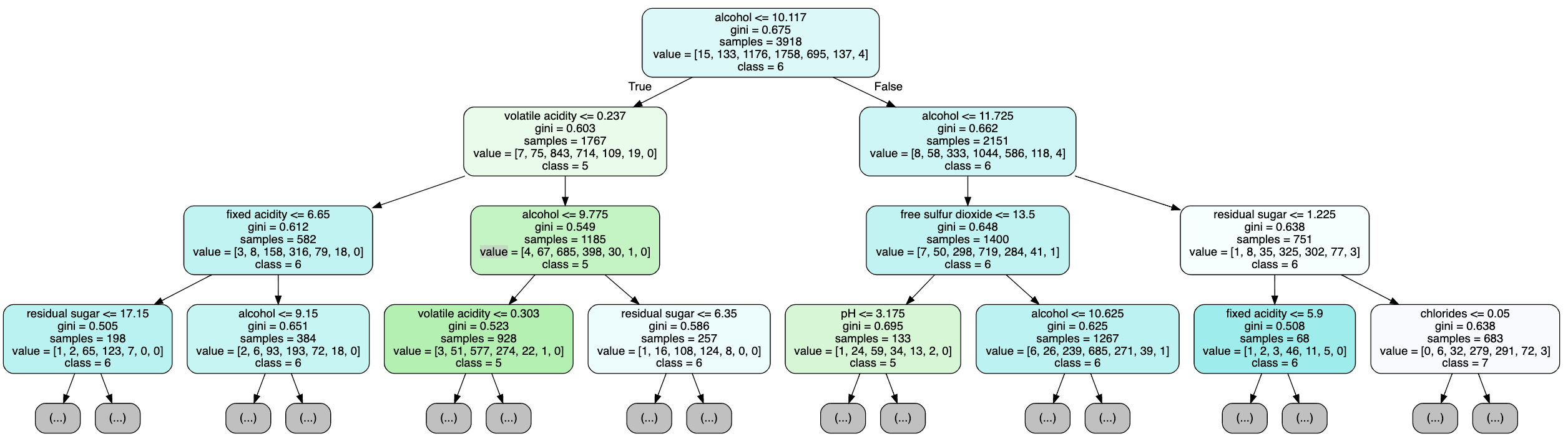


Figure 5: The Plot of the Decision Tree from the Decision Tree Classifier

This figure is the Decision Tree from the Decision Tree Classifier. The feature name of the beginning node is alcohol, and the split rule is <= 10.117 goes True, and otherwise. The feature name of the left node on layer one is volatile acidity, and the split rule is <= 0.237 goes True, and otherwise. The feature name of the right node on layer one is alcohol, and the split rule is <= 11.725 goes True, and otherwise. The feature name of the first left node on layer two is fixed acidity, and the split rule is <= 6.65 goes True, and otherwise. The feature name of the second left node on layer two is alcohol, and the split rule is <= 9.775 goes True, and otherwise. The feature name of the third left node on layer two is free sulfur dioxide, and the split rule is <= 13.5 goes True, and otherwise. The feature name of the last node on layer two is residual sugar, and the split rule is <= 1.225 goes True, and otherwise. For each leaf node, their class labels are 6, 6, 5, 6, 5, 6, 6, 7 from left to right, respectively.

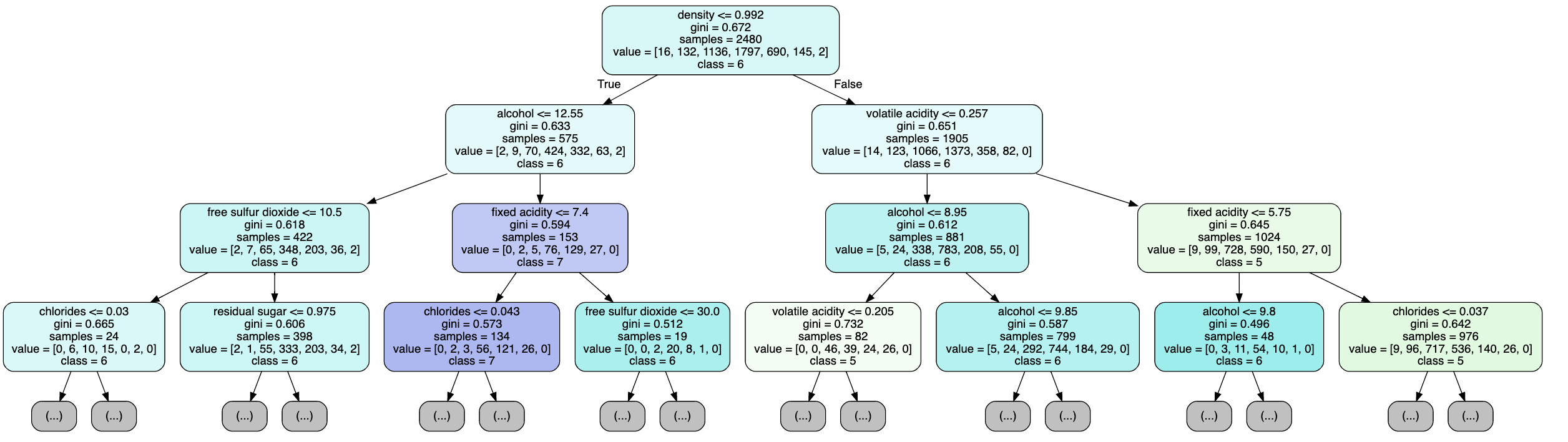


Figure 6: The Plot of a Decision Tree from the Random Forest Classifier

This figure is a Decision Tree from the Random Forest Classifier. The feature name of the beginning node is density, and the split rule is <= 0.992 goes True, and otherwise. The feature name of the left node on layer one is alcohol, and the split rule is <= 12.55 goes True, and otherwise. The feature name of the right node on layer one is volatile acidity, and the split rule is <= 0.257 goes True, and otherwise. The feature name of the first left node on layer two is free sulfur dioxide, and the split rule is <= 10.5 goes True, and otherwise. The feature name of the second left node on layer two is fixed acidity, and the split rule is <= 7.4 goes True, and otherwise. The feature name of the third left node on layer two is alcohol, and the split rule is <= 8.95 goes True, and otherwise. The feature name of the last node on layer two is fixed acidity, and the rule is <= 5.75 goes True, and otherwise. For each leaf node, their class labels are 6, 6, 7, 6, 5, 6, 6, 5 from left to right, respectively.

Reference

Learn. (n.d.). Retrieved February 11, 2020, from https://scikit-learn.org/stable/

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Reference on Code

I used the Sklearn, pandas, matplotlib, graphviz, numpy packages to complete the code, so some of the code may very similar.

I also used https://www.tarekatwan.com/index.php/2017/12/how-to-plot-a-confusion-matrix-in-python/ as a reference to know how to plot the confusion matrix, and I used the exercise code from class as well. Thanks.