Model Training and Evaluation Report

Chardonnay Or Merlot?

Name: Ruixie Fang
ID: 002422276

I. Task 1

This dataset is related with wine. Models are built to help to identify the wine category based on the wine components.

There are 13 descriptive attributes and 1 target attribute in the dataset *wineData.csv*. Each attribute has 118 observations. The feature *Class* is a categorical attribute with values *Chardonnay* and *Merlot*. These two classes are balanced. Each class has 59 observations. In order to be better used in later analysis, values in class are mapped as {0,1}. After class mapping, all features types are numeric now.

TABLE I. Attributes Description

Attributes Category	Type	Attributes Name
Descriptive	numeric	*'Alcohol', 'Malic acid', 'Ash', 'Proline', 'Flavanoids', 'Alcalinity of ash','Magnesium', 'Hue', 'Total phenols', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'OD280/OD315 of diluted wines',*
Target	categorical	*'Class'*

There might be no significant difference between normalizing data to [0, 1] vs. [0, 3] range, since the min-max normalization allows to maintain all relative differences between the values for the feature, and when the data is normalized, the original proportions between data points are guaranteed to be perfectly preserved. Although the range is different, the whole scale among instances is still similar.

II. Task 2

A. Stratified Holdout Sampling

To better train and evaluate the models, the original dataset is split into training and testing data sets by performing stratified holdout sampling. The performance of two subsets can help to understand the model. Each of these two sets has an even-out number of class labels. The detail information is as follows:

TABLE II. Summary Table for Training and Testing datasets

Datasets	Proportion	Training Testing
Training	**2/3**	13 descriptive features 1 target feature **78** observations
Testing	**1/3**	13 descriptive features 1 target feature **40** observations

B. Decision Tree

Decision trees are powerful and popular tools for classification and prediction. Here, we compare both *entropy* (information gain) and *gini* (Gini impurity) methods from depth-range 1 to 10.

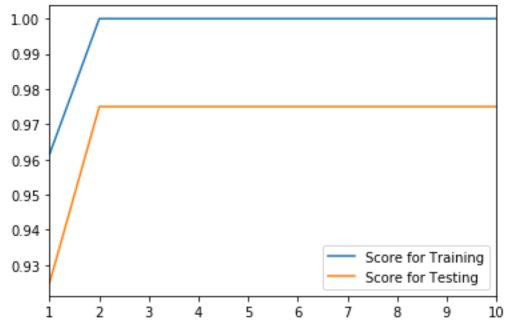

Fig. 1. Entropy Score

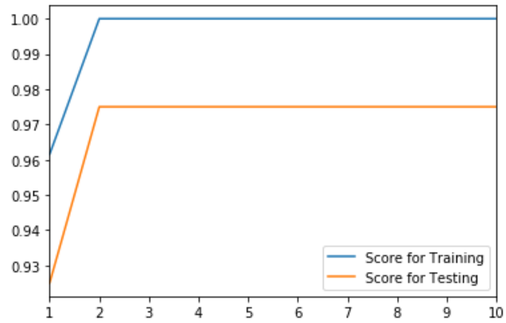

Fig. 2. Gini Score

Here is the summary table for two methods:

TABLE III. Decision Tree Scores for Training and Testing

Methods	Depth-Range	Score for Training	Score for Testing
Entropy	1	0.961538	0.925
	2-10	1.00	0.975
Gini	1	0.961538	0.925
	2-10	1.00	0.975

From the above performance plots and table, we can see that there is no difference between the two classification methods. Both training and testing datasets get their best scores at depth 2, and the performance of two methods begin to keep constant at depth 2, so the best decision tree is when tree depth = 2, with the best score = 97.5%.

Decision tree plots for *entropy* and *gini* with depth = 2 are as follows.

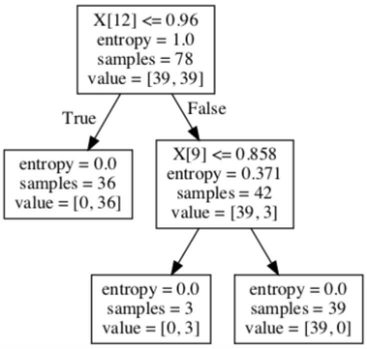

Fig. 3. Decision tree for *entropy*

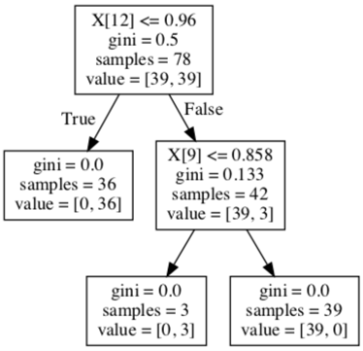

Fig. 4. Decision tree for *gini*

III. Task 3

A. KNN

The 2nd classifier used here is KNN. In this case, four different methods are used. The range of nearest neighbors is [1,10].

- *Euclidean, Uniform*

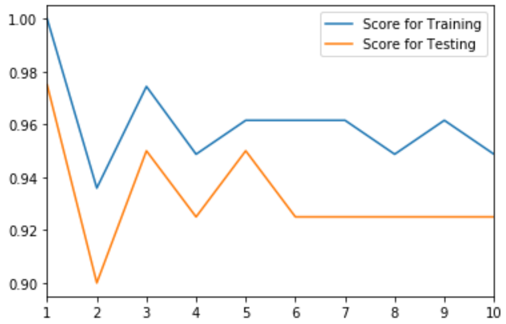

Fig. 5. KNN (Euclidean, Uniform)

- *Manhattan, Uniform*

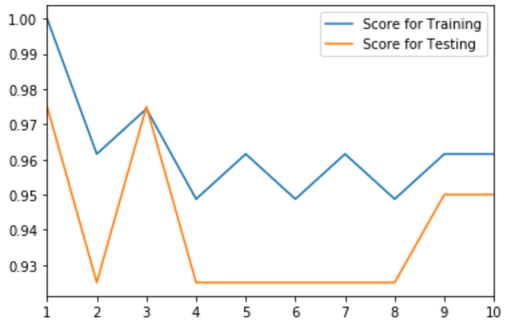

Fig. 6. KNN (Manhattan, Uniform)

- *Euclidean, Weighted*

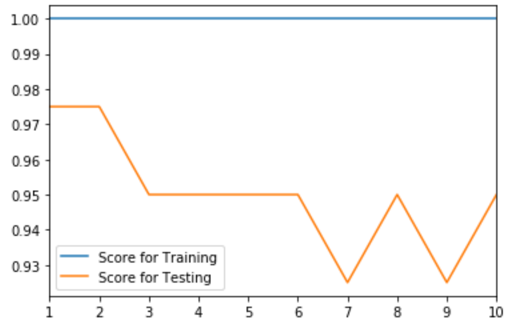

Fig. 7. KNN (Euclidean, Weighted)

- *Manhattan, Weighted*

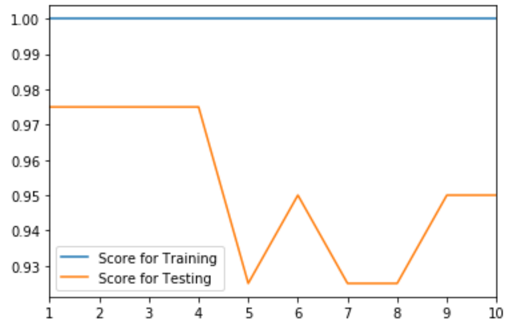

Fig. 8. KNN (Manhattan, Weighted)

The start point of the four methods has no surprise as expectation. Comparing with other methods, the KNN classifier has the ability to memorize the training set. When k = 1, the training score is 100% is because the closest neighbor point of any training dataset is itself.

TABLE IV. KNN(four methods) Score for Training and Testing

Method	Score for Training	Score for Testing	K (when testing score is best)
KNN(Euclidean,Uniform)	1.00	0.975	1
KNN(Manhattan,Uniform)	1.00	0.975	1,3
KNN(Euclidean,Weighted)	1.00	0.975	1,2
KNN(Manhattan,Weighted)	1.00	0.975	1,2,3,4

The above performance plots and table show that the best score is 97.5%. All four methods can provide best KNN classification result.

The most common part of the best score is when k = 1. However, if k = 1, the algorithm might be very sensitive to all sorts of distortions like noise, outliers and so on. Properly higher k value might be better and provide more robust results. And also weighted k nearest neighbor models usually make more accurate predictions as they take into account the fact that some of the nearest neighbors can actually be quite far away.

Based on the above reasons, in this dataset, the weighted method with properly higher k value might be the best:

KNN (Euclidean, Weighted) with k=2,
KNN (Manhattan, Weighted) with k=2,3,4

IV. Task 4

A. Random Forest

The 3rd classifier we used here is the random forest. The test range is [1,20]. Training and testing scores in the random forest are shown below.

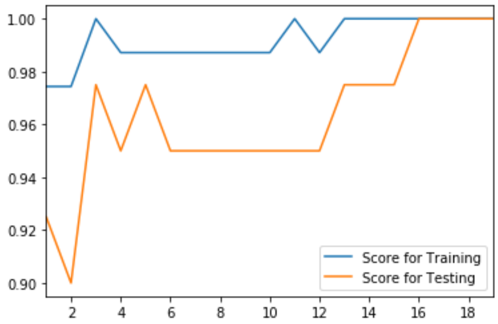

Fig. 9. Random Forest

The best score for both training and testing data is 100% after n = 16. There are several large fluctuations before n = 16, especially from 2 to 3.

The following table shows the rank of all descriptive features based on their importance.

TABLE V. Importance Rank of Descreiptive Features

Rank	Variable Name	Importance
1	Proline	0.323692
2	Alcohol	0.288800
3	Magnesium	0.118603
4	Color intensity	0.088492
5	Flavanoids	0.071114
6	Total phenols	0.032225
7	Nonflavanoid phenols	0.021390
8	OD280/OD315 of diluted wines	0.015870
9	Alcalinity of ash	0.014716
10	Ash	0.012231
11	Hue	0.012028
12	Malic acid	0.000840
13	Proanthocyanins	0.000000

Since the random forest model is not good enough, we can try to improve the model efficiency based on importance rank table. Taking good advantage of several important factors or removing some unimportant factors might be a good way.

B. Comparision

It would be fair to compare the random forest classifier against the classifiers from the decision tree and KNN. The random forest model is a combination of bagging, subspace sampling, and decision trees. And these classifiers are widely used classification techniques in machine learning.

Out of all these three classifiers investigated in Tasks 2 - 4, the random forest is better than others. Since random forest models build multiple decision trees and combine them to get a more accurate and stable prediction, the predictions could be closer to the true value on average. A fair comparative test is based on the same datasets.

V. Task 5

A. Model improvement

In order to improve the model efficiency and quality, several ideas below are used in the dataset to get better model results.

- *Remove unimportant features*

Malic acid and *Proanthocyanins* are removed from the original dataset based on the importance rank table.

- *Derive new features based on correlation*

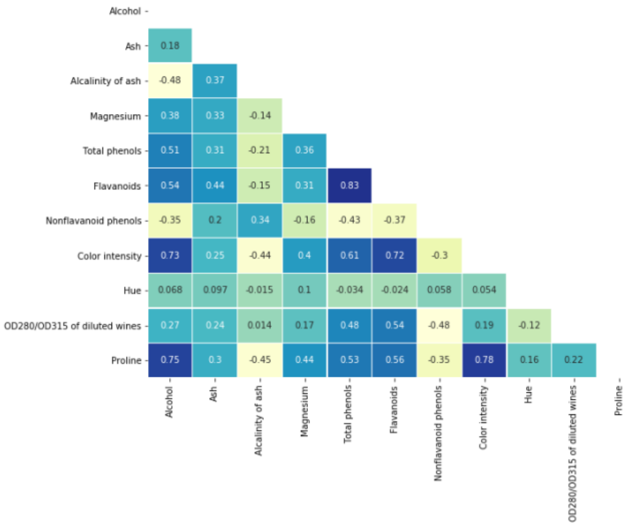

Fig. 10. Correlation heatmap

Based on the correlation heatmap plot, *Alcohol* and *Proline*, *Flavanoids* and *Total phenols* have a high positive correlation. Two new features are derived from the raw features: *Proline+Alcohol* and *Flavanoids+Total phenols*.

- *Remove unimportant features again*

After adding new features to the dataset, all processes are run again to remove the unimportant feature, and also remove features we used to derive new features since the new feature is more powerful.

B. Conclusion

Yes, New results can beat the best results.

After model improvement, although there is no significant change for the best score, several models become more stable. For example, the random forest model is much better than before, as shown below. Although there is a potential overfitting point around n = 11, both training and testing datasets show stable and constant improve when the estimator number becomes greater.

The reason might because potential outliers might be removed, and the new derived features might more stress the difference between two classes, which can help the model to make a more accurate prediction.

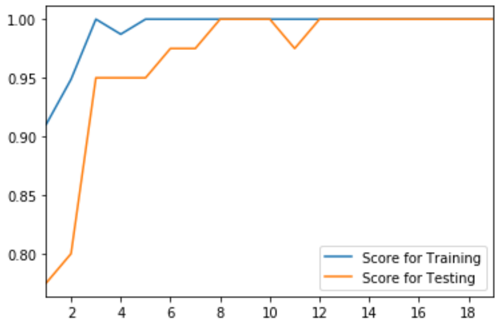

Fig. 11. Random Forest After improvement

C. New ideas

Feature selection (derive feature /drop column) is a good way to improve the model, but since the sample size is small, some limitation is still hard to erase. For example, when we do stratified holdout sampling, it will be much better if we have very large datasets. This ensures that the training set and test set are sufficiently large to train an accurate model and fully evaluate the performance of that model.

If possible, maybe we can try to expand the sample size by adding new collected observations or using Markov Chain to simulate a large size sample based on proper assumption.