DSC 540
HW1
CHAONAN SHI

Pima Diabetes

**Question #1: Run the code 5 times, record the accuracy and AUC scores of each run. What do you notice about the scores?*

Performance/Times	1	2	3	4	5
Accuracy	0.695	0.650	0.661	0.672	0.706
AUC	0.5	0.5	0.5	0.5	0.5

Analysis:

By running code 5 times, I found that Accuracy score has changed each time roughly in range 0.65~0.71. However, the AUC score remains stable at 0.5.

From my point of view, the changing of result score from Accuracy is because of data split. Since each time we run the code and processing of data split will select different data as Training & Testing. In this case, I am not surprise that fluctuation of Accuracy Score. The ROC score remaining in stable is because of AUC is calculating the possibility of positive sample over negative sample:

$$AUC = P(P_{postive} > P_{negative})$$

So For ROC score, 0.5 represents the positive sample= negative sample, whatever the how times of running.

For making our result stable, we need to either apply Cross-Validation or set up ‘seed’ number to make this result repeatable.

Next, let’s try changing one of the parameters of the Decision Tree.

On line 116, change the criterion option from ‘gini’ to ‘entropy’

**Question #2: Run the code again 5 times, record the accuracy and AUC scores of each run. What do you notice about the scores? How do they compare to scores above in question 1?*

Performance/Times	1	2	3	4	5
Accuracy	0.657	0.657	0.657	0.691	0.702
AUC	0.5	0.5	0.5	0.5	0.5

Analysis:

By running code 5 times, I found that Accuracy score has increased each time roughly from 0.65~0.71. However, the AUC score remains stable at 0.5.

By comparing the score in question 1, the results roughly same, especially for the AUC score.

From my point of view, there is no significant difference between GINI index and Entropy for calculation the Accuracy score, both of these two algorithms based on the impurity of data nodes. However, the score of accuracy from entropy increased each time I think it might due to the overfitting problem, either we over-calculated data from each nodes or get too many nodes in splitting.

Again, for making our result stable, we need to either apply Cross-Validation or Set up 'seed' number to make this result repeatable.

Now, let's setup scorers for the cross-validation split. This works a bit differently, we have to set up a dictionary of scorers first, then pass that into the cross_validate function call. The function will then return a dictionary of scores, which we can call by name.

**Question #3: Run the code 5 times, record the accuracy and AUC scores of each run. What do you notice about the scores? How do they compare to the simple test/train split scores in question 1?*

Performance/Times	1	2	3	4	5
Accuracy	0.65	0.65	0.65	0.65	0.65
AUC	0.5	0.5	0.5	0.5	0.5

Analysis:
By running code 5 times, I found both Accuracy score and AUC score remaining at same each time.

Comparing to the simple test/train scores in question 1, the Cross Validation definitely gives us more stable result for accuracy score. The accuracy score remaining on 0.65 and AUC stay on the 0.5 which same as what we got at question 1.

From my point of view, since the cross validation test performance for 20% (setup CV=5) from dataset. So the results should more stable and reliable by applying the cross validation.

**Question #4: Run the code once for each cv setting (3,8,10), record the accuracy and AUC scores. What do you notice about the scores? How do they compare to the CV performance above in question 3?*

Performance/CV	3	8	10
Accuracy	0.65	0.65	0.65
AUC	0.5	0.5	0.5

Analysis:
By running the code once for each cv setting (3,8,10), both accuracy and AUC score remaining at same each time;
Comparing to the CV performance score in question 3, the performance does not change. Surprisedly I suppose the performance be better at CV=10 than CV=3 (more folds more accuracy).

Wine Quality Dataset

**Question #5: Run the code 5 times, record the RMSE and Expl Variance scores of each run. What do you notice about the scores?*

Performance/Times	1	2	3	4	5
RMSE	0.88	0.802	0.78	0.81	0.82
Explained_Variance_Score	-0.18	-0.06	0.03	0.03	-0.02

Analysis:
By running code 5 times, I found that RMSE score has changed each time roughly in range 0.77~0.9 and Explained_Variance_Score has changed each time roughly in range -0.2~0.05

From my point of view, the changing of result score from RMSE and Explained_Variance_Score are because of data split. Since each time we run the code and processing of data split will select different data as Training & Testing. In this case, I am not surprise that fluctuation of RMSE and Explained_Variance_Score.

Next, let's try changing one of the parameters of the Decision Tree.

On line 191, change the criterion option from 'mse' to 'friedman_mse'

**Question #6: Run the code again 5 times, record the RMSE and Expl Variance of each run. What do you notice about the scores? How do they compare to scores above in question 5?*

Performance/Times	1	2	3	4	5
Friedman_mse	0.81	0.78	0.75	0.75	0.8
Explained_Variance_Score	0.02	0.08	0.12	-0.02	0.07

Analysis:
By running code 5 times, I found that Friedman_mse score has changed each time roughly in range 0.75~0.82 and Explained_Variance_Score has changed each time roughly in range -0.2~0.1

From my point of view, the changing of result score from Friedman_mse and Explained_Variance_Score are because of data split. Since each time we run the code and processing of data split will select different data as Training & Testing. In this case, I am not surprise that fluctuation of RMSE and Explained_Variance_Score.

Basically there are not significant difference between Friedman_mse and RMSE.

**Question #7: Run the code 5 times, record the RMSE and Expl Variance scores of each run. What do you notice about the scores? How do they compare to the simple test/train split scores in question 5?*

Performance/Times	1	2	3	4	5
RMSE	0.9	0.9	0.9	0.9	0.9
Explained_Variance_Score	-0.31	-0.31	-0.31	-0.31	-0.31

Analysis:

Comparing to the simple test/train scores in question 5, the Cross Validation definitely gives us more stable result for both RMSE and Explained_Variance_Score score. The RMSE score remaining on 0.9 and Explained_Variance_Score stay on the -0.31.

From my point of view, since the cross validation test performance for 20% (setup CV=5) from dataset. So the results should more stable and reliable by applying the cross validation.

**Question #8: Run the code once for each cv setting (3,8,10), record the RMSE and Expl Variance. What do you notice about the scores? How do they compare to the CV performance above in question 7?*

Performance/CV	3	8	10
RMSE	0.98	0.93	0.91
Explained_Variance_Score	-0.46	-0.5	-0.47

Analysis:

By running the code once for each cv setting (3,8,10), I found that RMSE score has changed each time roughly in range 0.90~0.99 and Explained_Variance_Score has changed each time roughly in range -0.5~0.45

Comparing to the CV performance score in question 7, the performance not stable.
Basically there are not significant difference between Friedman_mse and RMSE.

From my point of view, the results are not surprise to me, since I suppose the performance be better at CV=10 than CV=3 (more folds meaning more accuracy).

**Question #9: Run the code once, record the RMSE and Expl Variance. What do you notice about the scores? How do they compare to the CV performance above in question 7? What features were selected, and which were removed?*

RMSE	0.96
Explained_Variance_Score	-0.48

Analysis:

By running the code once, I found that RMSE score is 0.96 and Explained_Variance_Score is -0.48
The selection for this methondoly are : ['fixed acidity', 'residual sugar', 'free sulfur dioxide', 'total sulfur dioxide', 'alcohol']
The selection for this methondoly removing are : Features (total/selected): Features (total, selected): 11 5; 6 removed

**Question #10: Run the code once, record the RMSE and Expl Variance. What do you notice about the scores? How do they compare to the CV performance above in question 7? What features were selected, and which were removed?*

RMSE	0.89
Explained_Variance_Score	-0.39

Analysis:

By running the code once, I found that RMSE score is 0.89 and Explained_Variance_Score is -0.39
The selection for this methondoly is ['volatile acidity', 'sulphates', 'alcohol']
The selection for this methondoly removing are : Features (total/selected): 11 3 ;[8] removed