Machine Learning Homework 2

2020047029

JunYeong Ahn

Q1.

	RF Acc	RF AUC
Trial1	**0.8178438661710037**	**0.8415270018621974**
Trial2	**0.7806691449814126**	**0.8353494623655914**
Trial3	**0.7769516728624535**	**0.8093230694037146**
Trial4	**0.7472118959107806**	**0.8272036474164133**
Trial5	**0.7546468401486989**	**0.8336644990366089**

#1a)

In this question, the decision tree used to predict the presence/absence of diabetes (target class) used entropy(measure of how much "information" each feature contains relative to the target) for its separation criterion. * In this case 'Accuracy' is simply the rate of correct classifications and AUC(Area Under the (ROC) Curve) is a measure of how well the classifier distinguishes between classes – how true positive rate and false positive rate trade off. Both are for classification problem.

Accuracy and AUC range from 0 to 1. In general case well distinguished classes lead to high accuracy of model, which seems to mean that Acc and AUC would have the same trend of increase/decrease. Seeing the five trials above, they somewhat follow such trend but not completely. When Acc increases from 0.7472118959107806 (Trial4) to 0.8178438661710037 (Trial1) AUC increases from 0.8093230694037146 (Trial3) to 0.8415270018621974 (Trial1) The figures were maximized both in Trial1 but minimized not in the same Trial – in Trial4 and Trial3 respectively, which means the increase/decrease of Acc doesn't always mean the increase/decrease of AUC. This implies that a single performance score cannot be the answer for deciding a good model so we should comprehensively take multiple scores into consideration (this is associated with 'gold standard' in Q12.), although both indicates the performance of classifiers. And given five trials AUC has the smaller range of fluctuation than Acc. Also AUC always showed higher figure than Acc in each trial.

#1b)

fit(*X*, *y*, *sample_weight=None)* - Build a forest of trees from the training set (X, y).

-Three Possible Parameters

1. X / This is the training input samples.

Shape : ***{array-like, sparse matrix} of shape (n_samples, n_features)***

2. y / This is the target values in real numbers**.**

Shape : ***array-like of shape (n_samples,) or (n_samples, n_outputs)***

3. sample_weight / This is the sample weights, assigning an explicit weight to each example used for training.

Shape : ***array-like of shape (n_samples,), default=None***

Q2.

Random Forest Acc : 0.77 (+/- 0.08)

Rnadom Forest AUC : 0.83 (+/- 0.07)

CV Runtime : 1.5378458499908447

Though the code has been run only once, the result of Acc and AUC will be the same whenever I run the code because it is the result of 5-fold CV. Unlike k-fold CV, in train/test split the train set is randomly and newly sampled in each trial and the test set also changes as a result, which results in different outcomes of Acc and AUC. In k-fold cross validation (Here, k is 5 but I'll use k to generalize the CV steps), however, the data is divided into k parts (folds). And k-1 of the parts are used for training, and 1 is used for testing. This procedure is repeated k times rotating the test set then those results are averaged. These folds are divided as they are now even whenever CV is conducted again. Thus, the outcome will not change whenever I try it. Thus 0.77 of Acc and 0.83 of AUC are averaged version of k (5) cases, which means they give an insight on how the model will generalize to an independent dataset.

Also (+/- '0.08') and (+/- 0.07) on the back of Acc(0.77) and AUC(0.83) mean they guarantee each 0.69~0.85(Acc) and 0.76~0.80(AUC) of error range with 95% confidence level – for example, '0.08' is 2 (originally 1.96 that stands for 95% confidence level) * standard deviation. As a result, standard deviation of Accs and AUCs for each fold case of CV would be 0.04 and 0.035.

Q3.

(No. of Trees)	RF Acc	RF AUC	CV Runtime
5	**0.74 (+/- 0.06)**	**0.77 (+/- 0.07)**	**0.12760233879089355**
10	**0.74 (+/- 0.05)**	**0.80 (+/- 0.08)**	**0.24483704566955566**
20	**0.76 (+/- 0.06)**	**0.81 (+/- 0.07)**	**0.5206584930419922**
50	**0.77 (+/- 0.06)**	**0.83 (+/- 0.07)**	**0.773944616317749**
200	**0.77 (+/- 0.07)**	**0.83 (+/- 0.06)**	**3.0807113647460938**
500	**0.78 (+/- 0.07)**	**0.83 (+/- 0.06)**	**7.890007257461548**
1000	**0.77 (+/- 0.07)**	**0.83 (+/- 0.07)**	**15.257170915603638**

Based on the result on the table Acc and AUC tend to almost increase from 0.74 (+/- 0.06) to 0.77 (+/- 0.07) and 0.77 (+/- 0.07) to 0.83 (+/- 0.07) respectively as No. of trees increases, although there are some exceptions; Acc has slightly decreased by 0.01 when No. of trees changed from 500 to 1000 with the error range being the same; AUC didn't increase (and even decrease) although No. of trees changed from 200 to 500. Also, it is hard to say that Acc and AUC are completely proportional to No. of trees since we cannot identify some noticeable changes in scores after No. of trees reach a certain level (N=50 seeing the table). But still we can say that there's a trend that performance gets better in terms of Acc and AUC as No. of trees increases. This makes sense since our model is a Random Forest, which yield nice results in terms of performance by taking many idiots(weak trees) and averaging them.

Q4.

#4a)

Random Forest Acc: 0.76 (+/- 0.05)

Random Forest AUC: 0.82 (+/- 0.06)

CV Runtime: 1.8610718250274658

In Q2.

Random Forest Acc : 0.77 (+/- 0.08)

Rnadom Forest AUC : 0.83 (+/- 0.07)

CV Runtime : 1.5378458499908447

The mean of Acc and error range (calculated in Q4. by 5-fold CV) are respectively 0.76 and 0.71~0.81. And the mean of AUC and error range (calculated in Q4. by 5-fold CV) are respectively 0.82 and 0.76~-0.88. Compared to the record in Q2, Acc has decreased by 0.01 and AUC also did by the same figure, which intuitively means the performance of our classification model become worse after Wrapper-Based feature selection. However, the scores of Q4 have almost no difference with those of Q2, with perhaps such slight reduced performance if any, but not outside the standard deviations in Q2. This implies that we can get comparable performance with fewer features, which is called *parsimonious* model that is always preferable.

But if the performance noticeably changes (greatly deviates the expected error range of previous model) after feature selection, there can still be a tradeoff between loss of score and something like interpretation simplicity.

#4b)

Among total 8 features; 'Times Pregnant', 'Blood Glucose', 'Blood Pressure', 'Skin Fold Thickness', '2-Hour Insulin', 'BMI', 'Family History', 'Age', only four features ('Blood Glucose', 'BMI', 'Family History', 'Age') have survived and the other four features (‘Times Pregnant’, ‘Blood Pressure’, ‘Skin Fold Thickness’, ‘2-Hour Insulin’) have been abandoned after Wrapper-Based feature selection.

Q5.

Random Forest Acc: 0.76 (+/- 0.05)

Random Forest AUC: 0.82 (+/- 0.06)

CV Runtime: 1.7326512336730957

Among total 8 features; 'Times Pregnant', 'Blood Glucose', 'Blood Pressure', 'Skin Fold Thickness', '2-Hour Insulin', 'BMI', 'Family History', 'Age', only four features ('Blood Glucose', 'BMI', 'Family History', 'Age') have survived and the other four features ('Times Pregnant', 'Blood Pressure', 'Skin Fold Thickness', '2-Hour Insulin') have been abandoned after feature selection.

In short, the selected features are exactly the same with the ones in Q4 which is also associated with the fact that the Acc and AUC scores are also the same with the ones in Q4. Unlike the features which is selected based on Wrapper-Based feature selection, features in Q5 are manually selected through seeking to the variable-importance of each variable and choosing only features whose figure is larger than the average of the feature-importances. As a result, 'Blood Glucose', 'BMI', 'Family History' and 'Age', which were selected in Q4 as well, were regarded as important features in predicting target class. With the same Random Forest model, the same features, the same data and the same No. of folds for cross-validation, it is natural that the scores recorded in Q4 and Q5 are exactly the same to each other.

Q6.

Random Forest RMSE:: 0.65 (+/- 0.02)

Random Forest Expl Var: 0.33 (+/- 0.11)

CV Runtime: 2.2160484790802

*MSE is a sum of differences between values predicted by a model and the values observed and RMSE is a root form of MSE. Unlike the other metrics, for RMSE lower is better. Explained Variance measures the proportion to which a mathematical model accounts for the variation (dispersion) of a given data set. Both are for regression problem.

Similar with the answer for Q2, in all the trials of CV the same result comes out unlike the trials in the train/test split situation. In k- fold cross validation, the data is divided into k parts (folds). And k-1 of the parts are used for training, and 1 is used for testing. This procedure is repeated k times rotating the test set then those results are averaged. The folds are divided as they are now even whenever CV is conducted. Thus the 0.65 of RMSE and -0.33 of Expl Var are averaged ones of k (5) cases, which means they give an insight on how the model will generalize to an independent dataset.

Also (+/- '0.02') and (+/- 0.11) on the back of RMSE(0.65) and Expl Var(0.33) mean they guarantee each 0.63~0.67(RMSE) and 0.22~-0.44(Expl Var) of error range with 95% confidence level – for example, '0.02' is 2 (originally 1.96 that stands for 95% confidence level) * standard deviation. Thus, standard deviation of RMSEs and Expl Vars for each fold case of CV would be 0.01 and 0.055.

Q7.

(No. of Trees)	RF RMSE	RF Expl Var	CV Runtime
5	**0.70 (+/- 0.04)**	**0.20 (+/- 0.13)**	**0.15239161491394043**
10	**0.67 (+/- 0.04)**	**0.28 (+/- 0.09)**	**0.2549670124053955**
20	**0.66 (+/- 0.04)**	**0.30 (+/- 0.09)**	**0.5971987247467041**
50	**0.65 (+/- 0.03)**	**0.32 (+/- 0.11)**	**1.4201991558074951**
200	**0.64 (+/- 0.02)**	**0.34 (+/- 0.10)**	**4.737312078475952**
500	**0.64 (+/- 0.02)**	**0.34 (+/- 0.11)**	**11.095213890075684**
1000	**0.64 (+/- 0.02)**	**0.34 (+/- 0.10)**	**23.239818334579468**

#7a)

Based on the result on the table RMSE (+ its standard deviation) and Expl Var tend to almost reduce and increase from 0.70 (+/- 0.04) to 0.64 (+/- 0.02) and 0.20 (+/- 0.13) to 0.34 (+/- 0.11) respectively as No. of trees increases, although they sometimes stay still (didn't get better); RMSE remained the same when No. of trees changed from 200 to 500 and 1000 with the error range being the same as well; Expl Var didn't change but only its standard deviation did a bit as No. of trees changed from 200 to 500 and 1000. Also, it is hard to say that RMSE and Expl Var are completely proportional and inverse - relationship to No. of trees since we cannot identify some noticeable changes in scores after No. of trees reach a certain level (N=200 seeing the table). But still we can say that there's a trend that performance gets better in terms of RMSE and Expl Var as No. of trees increases. This makes sense since our model is a Random Forest, which yield nice results in terms of performance by taking many idiots(weak trees) and averaging them.

Comparing the result in Q3 and in Q7, both are the same *only* in that the increase of the number of trees generally leads to overall improvement in scores. However, the moment scores (Acc & AUC and RMSE & Expl Var) almost converge (I'm not sure this terminology is appropriate here but I believe you know what I want to mean) was different, although it is hard to set a clear boundary for such moments considering just a few trials. And most importantly, scores in Q3 are for classification but those in Q7 are for regression – situations and performance scores each situation use are totally different, so making direct comparisons of scores is improper and only similar and approximate 'trend' between performance and No. of trees could be found.

#7b)

It is expected that the number of trees would be proportional to the CV runtime because runtimes are associated with the number of calculations within folds. When the number of folds is K, the data is divided into K parts (folds). And K-1 of the parts are used for training, and 1 is used for testing. This procedure is repeated K times rotating the test set, which roughly means there is K times of calculation within one tree. Considering I have N trees, runtimes will be almost proportional to KN. And all the K in the trees are 5 so they will be proportional to N.

And my expectation seems to be true - 5 : 10 : 20 : 50 : 200 : 500 : 1000 ≒ 0.15(CV Runtime when No. of tree = 5) : 0.25(N=10) : 0.59(N=20) : 1.42(N=50) : 4.73(N=200) : 11.09(N=500) : 23.23(N=1000).

When I decide what number of trees to choose based on the result on the table, the better scores and the shorter runtime, the better No. of trees. The performance of the model gets better as No. of trees increases in terms of RMSE with the score itself and the standard deviation getting smaller. The performance also gets better as No. of trees increases in terms of Expl Var with the score itself and the standard deviation getting larger and smaller, although there is exception when No. of trees = 500 (the standard deviation slightly increases by 0.005 (0.01/2)). Seeing the scores (but not runtimes), the performance seems to be the best when N = 200 & 500. And N = 200 is much better than N = 500 since it has much lesser runtime.

When comparing N = 200 with N = 50, however, the choice could be somewhat subjective – those two have almost no difference but N = 200 is slightly better in that 1) the minimum of RMSE within the error range is the same each other but the maximum of it is lower 2) Expl Var is higher and the standard deviation of it is lower, but the runtime becomes much lesser when N = 50. So trade-off problem between runtime and performance arises and the answer to this is dependent on what the decision maker think is the more important.

Q8.

#8a)

Random Forest RMSE:: 0.68 (+/- 0.03)

Random Forest Expl Var: 0.25 (+/- 0.11)

CV Runtime: 1.979701280593872

In Q6.

Random Forest RMSE:: 0.65 (+/- 0.02)

Random Forest Expl Var: 0.33 (+/- 0.11)

CV Runtime: 2.2160484790802

The mean of RMSE and error range (calculated in Q8. by 5-fold CV) are respectively 0.68 and 0.65~0.71. And the mean of Expl Var and error range (calculated in Q8. by 5-fold CV) are respectively 0.25 and 0.14~-0.36. Compared to the record in Q6, RMSE has increased by 0.03 and Expl Var has decreased by 0.08, which intuitively means the performance of our classification model become worse after Wrapper-Based feature selection. Considering error ranges of scores, overlapped sections between RMSE of Q6 and that of Q8 exist (Expl Var too). However, unlike the previous score comparisons such as Q2 vs. Q4 which was almost one-sided game due to more parsimonious model showing similar performance in Q4, this time the maximum and the mean of RMSE and the minimum of Expl Var are outside the expected error range in Q6.

Thus I think this situation is closer to the trade-off problem between some loss of score and getting parsimonious model rather than one-sided game.

#8b)

Among total 11 features; 'fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol', only three features ('volatile acidity', 'sulphates', 'alcohol') have survived and the other eight features (‘fixed acidity’, ‘citric acid’, ‘residual sugar’, ‘chlorides’, ‘free sulfur dioxide’, ‘total sulfur dioxide’, ‘density’, ‘pH’) have been abandoned after Wrapper-Based feature selection.

Q9.

Random Forest RMSE:: 0.65 (+/- 0.02)

Random Forest Expl Var: 0.33 (+/- 0.11)

CV Runtime: 2.273958921432495

In Q6.

Random Forest RMSE:: 0.65 (+/- 0.02)

Random Forest Expl Var: 0.33 (+/- 0.11)

CV Runtime: 2.2160484790802

---

*Normalization is used to alter a feature so it is more similar to a normal (Gaussian) distribution and when features have different scales.

In all the trials of CV the same result comes out unlike the trials in the train/test split situation. In k- fold cross validation, the data is divided into k parts (folds). And k-1 of the parts are used for training, and 1 is used for testing. This procedure is repeated k times rotating the test set then those results are averaged. The folds are divided as they are now even whenever CV is conducted. Thus the 0.65 of RMSE and -0.33 of Expl Var are averaged ones of k (5) cases, which means they give an insight on how the model will generalize to an independent dataset.

Also (+/- '0.02') and (+/- 0.11) on the back of RMSE(0.65) and Expl Var(0.33) mean they guarantee each 0.63~0.67(RMSE) and 0.22~-0.44(Expl Var) of error range with 95% confidence level – for example, '0.02' is 2 (originally 1.96 that stands for 95% confidence level) * standard deviation. Thus, standard deviation of RMSEs and Expl Vars for each fold case of CV would be 0.01 and 0.055. Comparing the result in Q9 to the result in Q6, normalization might help features to have the same scale with others and become more similar to a normal (Gaussian) distribution but didn't directly change the scores (maybe it's a special case that normalization doesn't work?).

Q10.

	Bagging Regressor RMSE	**Bagging Regressor Expl Variance**
Trial1	**0.6294261208464481**	**0.4511437872835734**
Trial2	**0.6527004344589156**	**0.3981770487957813**
Trial3	**0.6200518411506297**	**0.4175545285791631**
Trial4	**0.6747883265990052**	**0.34297064802990973**
Trial5	**0.6109857375001249**	**0.417165221483499**

#10a)

In Q6.

Random Forest RMSE:: 0.65 (+/- 0.02)

Random Forest Expl Var: 0.33 (+/- 0.11)

CV Runtime: 2.2160484790802

RMSE of the Bagging Regressor stayed within around 0.6~0.7 (from 0.6109857375001249 in Trial5 to 0.6747883265990052 in Trial4). And explained variance is distributed around about 0.4. (from 0.34297064802990973 in Trial4 to 0.4511437872835734 in Trial1). Both have something in common in that they deal with the model and actual data. The lower RMSE and the higher Explained Variance, the better prediction of the regression model. However, we cannot identify the trend that RMSE grows as Expl Var decreases and that one decrease as one grows as well. Even with the same RMSE, Expl Var can de different and with higher RMSE Expl Var can be bigger as well. Seeing the Trial1 and 3, RMSE increases in Trial1 but Expl Var increases in Trial1 as well. So Expl Var seems to have nothing to do with the value of RMSE.

Defining how stable the scores are in each trial is not valid here without some further works such as calculating the average value and the standard deviation of RMSE and Expl Var in five trials.

And I cannot tell the difference (=compare them directly) between Q6 and Q10 because they take different evaluation methods: 5-fold CV and train/test split respectively. The two methods are fundamentally different in that 5-fold CV results from averaging five scores but test/train split randomly and newly draw train set at each trial which leads to constant change of scores.

#10b)

*Bagging Regression encompasses several works from the literature. When random subsets of the dataset are drawn as random subsets of the samples, then this algorithm is known as Pasting. If samples are drawn with replacement, then the method is known as Bagging. *When random subsets of the dataset are drawn as random subsets of the features, then the method is known as Random Subspaces*. / Inferring from this part of API link document, we need to handle these two parameters;

-**bootstrap_features :** Whether features are drawn with replacement.

Default of this parameter is 'False' so we need to change it into 'True'.

-**bootstrap :** Whether samples are drawn with replacement.

Default of this parameter is 'True' so we should change it into 'False' so that I could *create a Random Subspaces model***.**

#10c)

The key to bagging is that trees are repeatedly fit to bootstrapped subsets of the observations. One can show that on average, each bagged tree makes use of around two-thirds of the observations.3 The remaining one-third of the observations not used to fit a given bagged tree are referred to as the out-of-bag (OOB) observations. We out-of-bag can predict the response for the ith observation using each of the trees in which that observation was OOB. This will yield around B((No. of separate training sets))/3 predictions for the ith observation. In order to obtain a single prediction for the ith observation, we can average these predicted responses - *we are dealing with Regressor*. This leads to a single OOB prediction for the ith observation. An OOB prediction can be obtained in this way for each of the n observations, from which the overall OOB MSE can be computed. The resulting OOB error is a valid estimate of the test error for the bagged model, since the response for each observation is predicted using only the trees that were not fit using that observation. It is widely known that with B sufficiently large, OOB error is virtually equivalent to LOOCV error which is unbiased but with higher variance with respect to k-fold CV. The OOB approach for estimating the test error is particularly convenient when performing bagging on large data sets for which cross-validation would be computationally onerous. However, since OOBE(Errors) ensembles are composed of less trees and are trained on overlapping training sets, the variance of the estimate can be larger.

So we can use one of them as our likes but neither estimator is by default better than the other, even though OOBE has a clear computational advantage since I can "fit and test" our forest at the same time.

Q11.

In Diabetes Dataset

In terms of accuracy and AUC scores, Random Forest(RF) outperformed Decision Tree(DT). If I compare the results from 5-fold cross validation (since we didn't conduct 3, 8 and 10 fold CV against RF), Acc and AUC of DT were 0.71(+/-0.08) and 0.69(+/-0.07) but those of RF are 0.77 (+/- 0.08) and 0.83 (+/- 0.07) (with 100 trees from Q2.).

In Wine Dataset

Likewise, RF outperformed DT comparing the results from 5-fold CV ;

DT) RMSE: 0.90 (+/- 0.10) / Expl Var: -0.31 (+/- 0.17)

RF) RMSE:: 0.65 (+/- 0.02) / Expl Var: 0.33 (+/- 0.11) / Scores in RF greatly improved, deviating far from the error range of DT's.

WHY?

Random forest leverages the power of multiple decision trees. It does *not* rely on the feature importance given by a single decision tree. Therefore, the random forest can generalize over the data in a better way. This randomized feature selection makes random forest much more accurate than a decision tree.

In Easy Words :

"Through exploring many diverse cases (each case corresponds to DT) and averaging them, Random Forest becomes powerful and robust to variation of data compared to a single Decision Tree which is very sensitive to the data it's been trained on. It resembles collective intelligence – one idiot cannot but multiple (numerous) idiots can make a clever decision."

Q12.

There are two typical of evaluating models - train/test split and cross validation. Especially cross validation(CV) is regarded as 'gold standard' in evaluating models since it has many strengths in general compared to traditional train/test splitting: 1) CV generalizes better since it uses all the data in evaluating. 2) It prevents underfitting when the given data is small. 3) It brings higher accuracy since it uses all the data in training 4) ,,, However, this takes much more time than train/test split due to its No. of iterations. Thus, although CV is recommended in most case and gold standard, this does not mean that it is almighty.

For example, when we have sufficient amount of data so that we can simply and reasonably evaluate the model just using train/test split we don't have to waste our time in cross validating. Likewise, there would be many other contextual situations where CV suits well or does not. I think that's why CV is so-called 'gold standard', not 'golden law' or something similar.

In many cases CV would be preferable, but not always.