Matthew Ragoza
Dr. Milos Hauskrecht
ISSP 2170 - Machine Learning
April 14, 2022

Assignment 10

Problem 1. Baseline models

a. I modified and ran the script hw10_DT.py to evaluate three different baseline models on the Pima data set. The baseline models consist of single decision tree classifiers with different settings. DT1 is a decision tree with a minimum of 5 samples per leaf node, but otherwise uses the default settings. DT2 is a "full" decision tree with no limit on the tree depth or minimum leaf node size. DT3 is a decision tree with a depth limit of one, i.e. it contains only a single split.

Evaluation metrics for these baseline models are shown in Table 1. DT2 completely memorized the training set, as seen in its perfect training accuracy and AUROC. However, it achieved a test accuracy of 0.732 and AUROC of 0.704. In contrast, DT1 had a test accuracy of 0.766 and test AUROC of 0.777. The fact that DT2 had perfect train performance, but worse test performance than a less flexible model (DT1), indicates that DT2 overfit to the training set.

On the other hand, DT3 showed signs of underfitting. It had the best overall test accuracy (0.775), which was even better than its train accuracy (0.736). Its test AUROC (0.681) was better than its train AUROC (0.652) as well. The low AUROC of DT3 compared to its test accuracy can be attributed to its extremely simple functional form, which results in a very sharp ROC curve with minimal "bulging" that is normally seen in such plots.

Model	Accuracy		AUROC	
	Train	Test	Train	Test
DT1	0.879	0.766	0.958	**0.777**
DT2	**1.000**	0.732	**1.000**	0.704
DT3	0.736	**0.775**	0.652	0.681

Table 1. Classification results for baseline decision tree models.

Problem 2. Bagging

a. I created a new script hw10_Bagging_DT.py which trains and evaluates ensembles of decision tree classifiers using the Bagging algorithm. I used the option 1 decision tree (DT1) from the previous problem and performed bagging with 5, 10, and 30 trees in the ensemble. The evaluation metrics for the bagging models in comparison with the single decision tree model are shown in Table 2 and plotted in blue in Figure 1.

The ensemble of five option 1 models (DT1-Bag5) performed slightly worse than the single model in terms of accuracy on the training set (0.868) and test set (0.758), but had similar train AUROC (0.955) and better test AUROC (0.823). Increasing the number of trees in the bagging models beyond 5 improved the training accuracy and AUROC, with the DT1-Bag30 model reaching a training accuracy of 0.924 and AUROC of 0.979. The bagging ensembles did not reach better test accuracy than the single model (0.766) but there was significant improvement in the test AUROC, with DT1-Bag30 reaching a test AUROC of 0.844.

b. I performed a similar analysis of bagging-based ensembles of decision trees, but using the option 2 model (DT2) from the previous problem. I ran the model with 5, 10, and 30 trees in the ensemble and recorded the evaluation metrics in Table 2. The models are also plotted in Figure 1 in orange. The bagging ensemble of 5 option 2 models (DT2-Bag5) saw a slight decrease in training performance in terms of both accuracy (0.970) and AUROC (0.992) compared to the single tree, which had achieved perfect training performance. However, test performance improved in both metrics (accuracy 0.740, AUROC 0.788).

Furthermore, increasing the number of trees in the option 2 bagging model to 10 or 30 continued to improve the test performance. The ensemble of 30 option

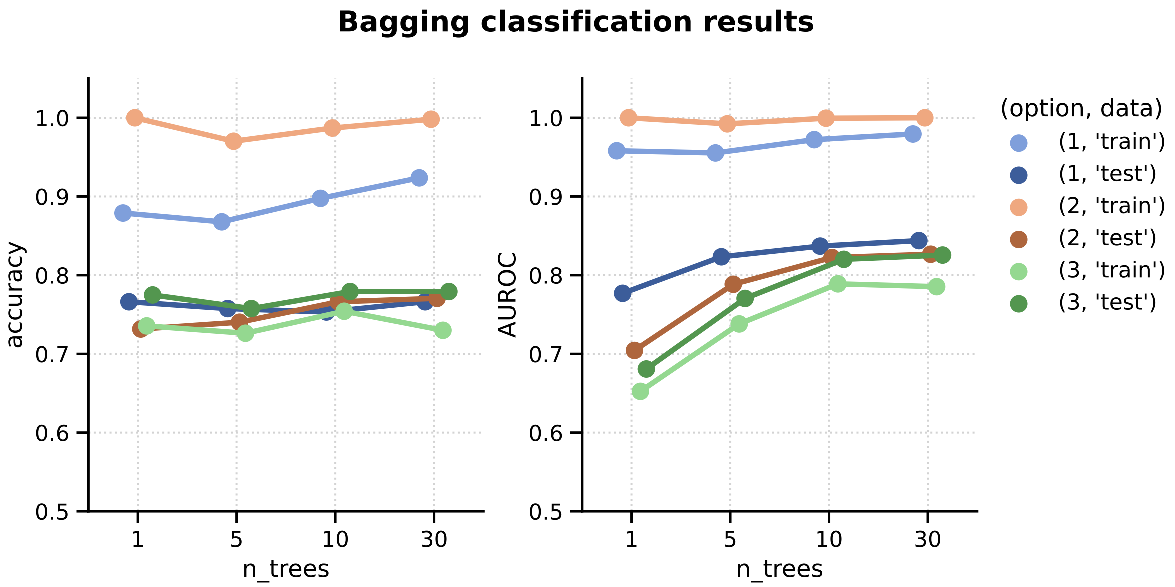

Figure 1. Classification results for bagging ensembles of decision trees.

2 models (DT2-Bag30) had nearly perfect training performance and far superior test performance compared to the single model, which overfit to the training set. DT2-Bag30 reached a test accuracy of 0.771 and AUROC of 0.827. This is because the effect of bootstrap aggregation is to reduce model variance, which is especially useful when the model is overfit. This was the case with the option 2 decision tree, which had singleton leaf nodes.

c. I repeated the analysis once more using the third decision tree (DT3) from the previous exercise, which was a low-variance model based on only a single split. I trained bagging models based on 5, 10, or 30 different option 3 models and reported the classification metrics in Table 2. These models are shown in green in Figure 1. Bagging did not have much effect on the training or test accuracy of these models. The test accuracy peaked at 0.779 using 10 trees (DT3-Bag10) with no further accuracy benefit from 30 trees. However, there was additional gain in test AUROC from using 30 trees (DT3-Bag30, AUROC 0.826). The underlying tree model already had low variance due to its extreme simplicity, so it was surprising that applying bagging (which reduces variance further) would result in such significant gains in test AUROC for this model.

d. The evaluation metrics for all 9 bagging models explored in this section are shown in Table 2, in comparison with the baseline models. In addition, the relation between the performance metrics and the parameters that were modified (decision tree option, number of trees, train vs. test data) are displayed in Figure 1. The important takeaway from these results is that increasing the number of trees in the ensemble tended to improve the test performance for all three tree options. This is due to the ability of bagging to reduce the variance of an overly flexible model, which often makes it more generalizable. The best performing model by test accuracy was DT3-Bag10 (0.779) and by test AUROC was DT1-Bag30 (0.844).

Model	Accuracy		AUROC	
	Train	Test	Train	Test
DT1	0.879	**0.766**	0.958	0.777
DT1-Bag5	0.868	0.758	0.955	0.823
DT1-Bag10	0.898	0.753	0.972	0.837
DT1-Bag30	**0.924**	**0.766**	**0.979**	**0.844**
DT2	**1.000**	0.732	**1.000**	0.704
DT2-Bag5	0.970	0.740	0.992	0.788
DT2-Bag10	0.987	0.766	0.999	0.822
DT2-Bag30	0.998	**0.771**	**1.000**	**0.827**
DT3	0.736	0.775	0.652	0.681
DT3-Bag5	0.726	0.758	0.738	0.770
DT3-Bag10	**0.754**	**0.779**	**0.789**	0.820
DT3-Bag30	0.730	**0.779**	0.785	**0.826**

Table 2. Classification results for bagging ensembles of decision trees.

Problem 3. AdaBoost

a. I created another script called `hw10_Adaboost_DT.py` which trains and evaluates an ensemble of decision tree classifiers on the Pima data set using the AdaBoost algorithm. I ran the script to evaluate AdaBoost ensembles based on sets of 5, 10, and 30 decision trees of the option 1-type (DT1) from Problem 3. The evaluation metrics are listed in Table 3 and the models are plotted in blue in Figure 2.

   All of the AdaBoost ensembles of DT1 trees reached perfect training performance (1.000) both in terms of accuracy and AUROC. The test accuracy of the 5-tree model (DT1-Ada5, 0.727) was worse than the single tree model, and the test accuracy degraded even further when increasing to 10 or 30 trees in the ensemble. However, there were slight improvements in test AUROC maxing out at 0.785 for the 10-tree model (DT1-Ada10).

b. I repeated the analysis by training AdaBoost ensembles of 5, 10, and 30 trees, this time based on the option 2 model (DT2) from Problem 1. These models are shown in Figure 2 in orange. As seen in Table 2, even the single-tree option 2 model attained perfect training performance, so it is unsurprising that all of the AdaBoost ensembles did as well. However, the test performance of the option 2 ensembles degraded compared to the baseline model. The model with 5 trees (DT2-Ada5) had test accuracy of 0.697 and AUROC of 0.664, and there was no change from this when increasing to 10 or 30 trees.

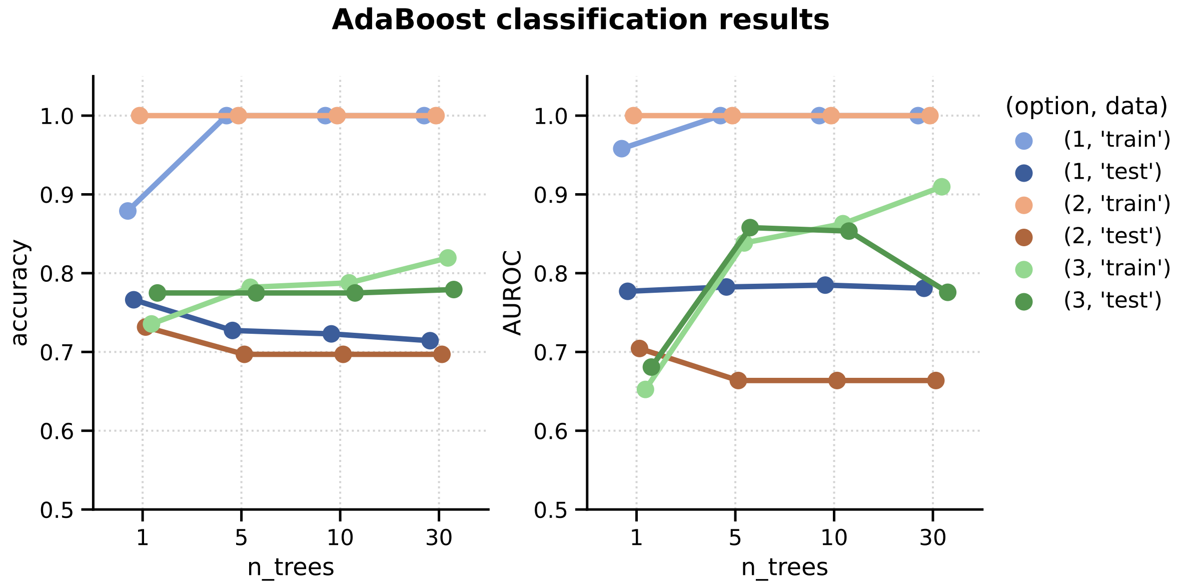

Figure 2. Classification results for AdaBoost with decision trees.

The AdaBoost algorithm tries to correct the mistakes made by an ensemble of weak learners by iteratively adding additional weak learners that are trained on data reweighted according to the ensemble classification error. Since the single-tree option 2 model already reached perfect discrimination on the training set, the only effect the AdaBoost algorithm had was to make the model even more overfit to the training set, reducing the test performance.

c. I performed an additional analysis by training and evaluating AdaBoost ensembles of 5, 10, and 30 trees of option 3 type (DT3), which consist of a single data split. The metrics for these methods are reported in Table 2 and plotted in green in Figure 2. In this setting, the training accuracy and AUROC both increased with the number of trees, reaching a maximum of 0.819 and 0.910, respectively, for 30 trees (DT3-Ada30). The test accuracy improved drastically when using AdaBoost with 5 trees (DT3-Ada5, 0.858) but then started to decline beyond that. Increasing the number of trees had minimal effect on the test accuracy, but the AdaBoost model with 30 trees had the peak test accuracy of 0.779.

d. The metrics for all AdaBoost models trained for this problem are listed in Table 3 and plotted in Figure 2. The AdaBoost algorithm had the effect of increasing the training performance for all tree models, but was prone to overfitting when based on models with high variance. The best improvement in test performance was from applying AdaBoost to option 3, which was the simplest model we evaluated. It seems that AdaBoost increases the variance of a model, in comparison with Bagging which decreases variance. This can help in turning a weak learner into a strong leaner, but can make an already strong learner overfit the training data. In summary, the best model in this set by test accuracy was DT3-Ada30 (0.779) and by test AUROC was DT3-Ada5 (0.858).

Model	Accuracy		AUROC	
	Train	Test	Train	Test
DT1	0.879	**0.766**	0.958	0.777
DT1-Ada5	**1.000**	0.727	**1.000**	0.782
DT1-Ada10	**1.000**	0.723	**1.000**	**0.785**
DT1-Ada30	**1.000**	0.714	**1.000**	0.781
DT2	1.000	0.732	1.000	**0.704**
DT2-Ada5	1.000	**0.697**	1.000	0.664
DT2-Ada10	1.000	**0.697**	1.000	0.664
DT2-Ada30	1.000	**0.697**	1.000	0.664
DT3	0.736	0.775	0.652	0.681
DT3-Ada5	0.782	0.775	0.838	**0.858**
DT3-Ada10	0.788	0.775	0.863	0.853
DT3-Ada30	**0.819**	**0.779**	**0.910**	0.776

Table 3. Classification results for AdaBoost with decision trees.

Problem 4. Gradient boosting

a. Next, I wrote a script `hw10_Gradientboosting_DT.py` which trains and evaluates an ensemble of decision trees on the Pima data set using gradient boosting. I evaluated gradient boosting ensembles of 5, 10, and 30 decision trees using the option 1 model (DT1) from the first problem. The evaluation statistics for these models are shown in Table 4 and the results are visualized in blue in Figure 3.

The training performance of the gradient boosted DT1 ensembles tended to increase with the number of trees. With 5 trees (DT1-Grad5), the train accuracy and AUROC were 0.907 and 0.990, and both metrics reached 1.000 by using 30 trees (DT1-Grad30). Interestingly, the test accuracy dipped at first for the ensemble of 5 trees (DT1-Grad5, 0.745) compared to the baseline model, and didn't quite recover as the number of trees increased further (DT1-Grad30, 0.758). However, the test AUROC improved with the number of gradient boosting trees, with peak performance achieved by using 30 trees (DT1-Grad30, 0.804).

b. I reran the analysis of gradient boosting ensembles by instead using option 2 trees (DT2), again evaluating ensembles of 5, 10, and 30 trees. The metrics are shown in Table 4 and plotted in orange in Figure 3. Similar to AdaBoost, gradient boosting based on DT2 trees resulted in perfect training performance across both metrics and all different ensemble sizes that we considered. The test accuracy of the ensembles was worse than the baseline model, with the best-performing gradient boosting ensemble having 10 trees (DT2-Grad10, 0.710). The test AUROC of this model was somewhat better than the baseline (0.719) but there was no further improvement by increasing the number of trees to 30.

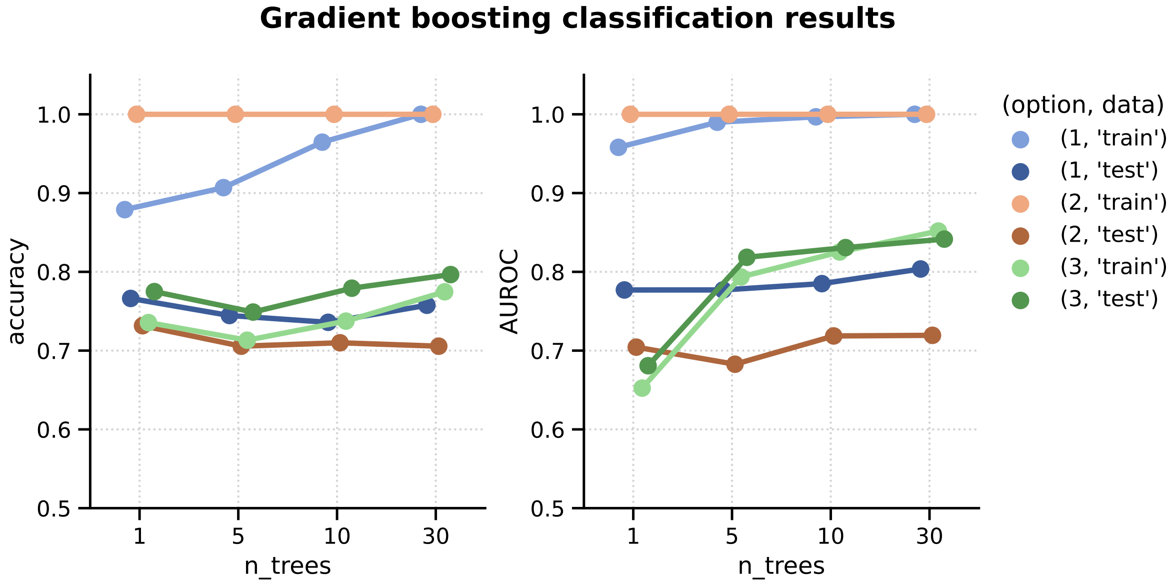

Figure 3. Classification results for gradient boosting of decision trees.

c. I again performed the evaluation of gradient boosting with 5, 10 or 30 trees, but based on option 3-type trees (DT3). These results are shown in Table 4 and colored green in Figure 3. These simpler models benefitted the most from increasing the ensemble size. There was an initial decline in train and test accuracy for the 5-tree ensemble (DT3-Grad5, 0.713, 0.749), but adding more trees improved the test accuracy greatly. In addition, both the train and test AUROC improved with increasing ensemble size compared to the baseline. The best model used 30 trees (DT3-Grad30) and reached a test accuracy of 0.797 and AUROC of 0.842.

d. The results for gradient boosting ensemble models are summarized in Table 4 and visualized in Figure 3. Gradient boosting is a similar algorithm as AdaBoost, but makes use of additive models that correct the mistakes of the previous ensemble based on the gradient of an error function. This analysis seems to indicate that gradient boosting is more robust to overfitting than AdaBoost, as we observed an improvement in test accuracy using this method across each of the different base tree options. However, the greatest improvement in test performance over the single-tree baseline was for DT3, which is a similar result as was seen with AdaBoost. In conclusion, the best gradient boosting model by both test accuracy and AUC was DT3-Grad30 (accuracy 0.797, AUROC 0.842). Gradient boosting appears to be an effective method for robustly improving model performance, especially when based on weak learners.

Model	Accuracy		AUROC	
	Train	Test	Train	Test
DT1	0.879	**0.766**	0.958	0.777
DT1-Grad5	0.907	0.745	0.990	0.777
DT1-Grad10	0.965	0.736	0.997	0.785
DT1-Grad30	**1.000**	0.758	**1.000**	**0.804**
DT2	1.000	**0.732**	1.000	0.704
DT2-Grad5	1.000	0.706	1.000	0.683
DT2-Grad10	1.000	0.710	1.000	**0.719**
DT2-Grad30	1.000	0.706	1.000	**0.719**
DT3	0.736	0.775	0.652	0.681
DT3-Grad5	0.713	0.749	0.794	0.818
DT3-Grad10	0.737	0.779	0.825	0.831
DT3-Grad30	**0.775**	**0.797**	**0.852**	**0.842**

Table 4. Classification results for gradient boosting of decision trees.

Problem 5. Random forest

a. For the final experiment, I created a script `hw10_RandomForest_DT.py` that trains a random forest ensemble classifier on the Pima data set. I ran this script using 5, 10, and 30 trees in the ensemble using option 1 (DT1) as the basic model. The evaluation metrics are reported in Table 5 and plotted in blue in Figure 4.

   The random forest with 5 DT1 trees had worse train accuracy (0.853) and AUROC (0.936) than the basic single-tree model, but it performed better on the test set in both metrics (accuracy 0.771, AUROC 0.817). Its test performance continued to improve as the number of trees increased. The test accuracy plateaued at 0.784 when using 10 trees (DT1-RF10) while the test AUROC reached 0.839 with 30 trees (DT1-RF30).

b. I redid the analysis of random forest with 5, 10, or 30 trees, this time using option 2 decision trees (DT2) from Problem 1. The results for these models are in Table 5 and shown in orange in Figure 4. Both the train and test accuracy declined compared to the baseline model when using a random forest of 5 trees (DT2-RF5, train accuracy 0.976, test accuracy 0.701). The accuracy improved as the number of trees grew, and the best accuracy on the test set was reached by using 30 trees (DT2-RF30, 0.762). The training AUROC stayed close to 1.000 across all random forests based on DT2, but the test AUROC improved with the ensemble size. The best model was a random forest of 30 trees (AUROC 0.825).

c. For the final analysis, I evaluated random forests of size 5, 10, and 30 based on option 3 models (DT3). Results for these models are recorded in Table 5 and shown in green in Figure 3. None of the random forest ensembles based on DT3 had better accuracy than the baseline model on either the training or test set.

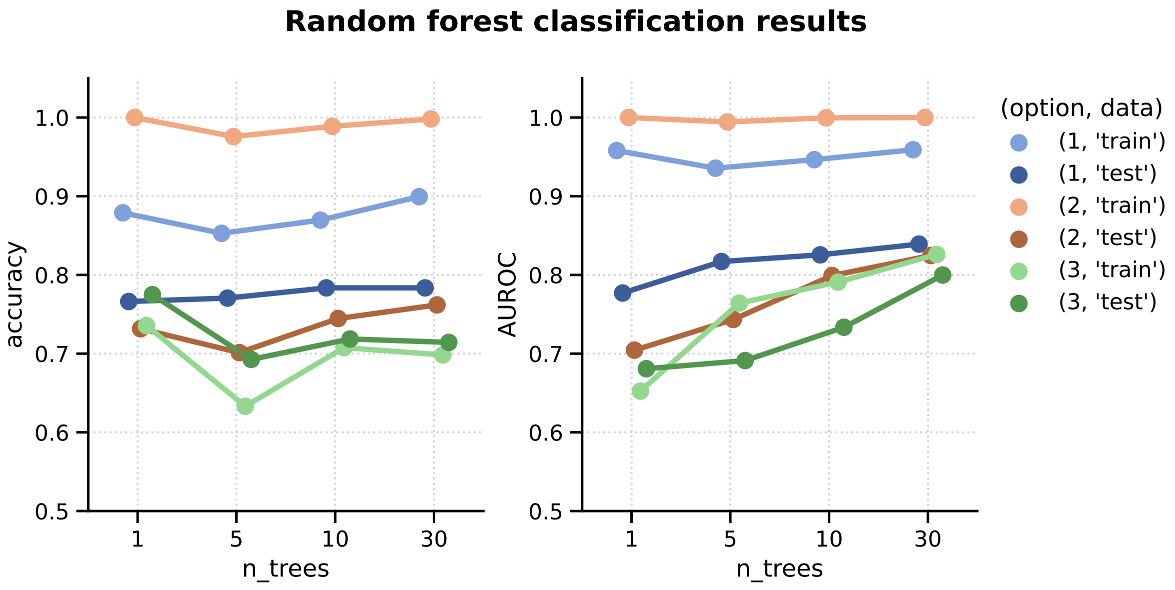

Figure 4. Classification results for random forest.

However, the train and test AUROC improved as the number of trees in the model grew. The highest test accuracy among the DT3 random forest models was 0.719 (DT3-RF10) while the highest test AUROC was 0.800 (DT3-RF30).

d. The summary metrics for the random forest classifiers are shown in Table 5 and plotted against the experimental parameters in Figure 4. Random forest is a somewhat different approach to ensemble modelling than bagging and boosting. The meoth trains the submodels based on random feature subsets instead of data subsets. We observed in this analysis that the best base model for random forest was DT1, which was neither the highest or lowest variance model that we tested. This model achieved a test accuracy of 0.784 and test AUROC of 0.839. However, test performance improved with the number of trees in the random forest classifier across all three base models. We can conclude that random forests are a significant improvement over basic decision trees with low tendency for overfitting.

Model	Accuracy		AUROC	
	Train	Test	Train	Test
DT1	0.879	0.766	0.958	0.777
DT1-RF5	0.853	0.771	0.936	0.817
DT1-RF10	0.870	**0.784**	0.946	0.826
DT1-RF30	**0.899**	**0.784**	**0.959**	**0.839**
DT2	**1.000**	0.732	**1.000**	0.704
DT2-RF5	0.976	0.701	0.994	0.743
DT2-RF10	0.989	0.745	**1.000**	0.799
DT2-RF30	0.998	**0.762**	**1.000**	**0.825**
DT3	**0.736**	**0.775**	0.652	0.681
DT3-RF5	0.633	0.693	0.764	0.691
DT3-RF10	0.708	0.719	0.791	0.733
DT3-RF30	0.698	0.714	**0.826**	**0.800**

Table 5. Classification results for random forest.