Homework 3
Shuyu Zhou

GitHub Link:
https://github.com/szhou12/MachineLearning4PublicPolicy/tree/master/pa3

Test window = 6 months long:
(Results are provided in *Results_Analysis_test6.ipynb*)

- Training time, testing time:

Decision trees model (parameters = {'max_depth': 1, 'criterion': 'gini', 'min_samples_split': 10, 'max_features': 'sqrt'}) is the most efficient one in terms of the training time whereas random forests model (parameters = {'max_depth': 1, 'min_samples_split': 2, 'n_estimators': 1, 'max_features': 'log2'}) requires the least testing time.

- Accuracy, F1 score, AUC-ROC:

In terms of accuracy score, random forests model (parameters = {'max_depth': 50, 'min_samples_split': 10, 'n_estimators': 100, 'max_features': 'log2'}) does better than other classifiers. This same random forests model also ranks the highest in F1 score, with the score being about 96% as well as the highest in AUC-ROC, with the corresponding score being 91.2%.

- Precision at 50%:

Decision trees model (parameters = {'max_depth': 1, 'criterion': 'entropy', 'min_samples_split': 2, 'max_features': 'sqrt'}), KNN (parameters = {'algorithm': 'kd_tree', 'weights': 'uniform', 'n_neighbors': 1}), and Random Forests model (parameters = {'max_depth': 1, 'min_samples_split': 2, 'n_estimators': 1, 'max_features': 'sqrt'}) all have the 100% precision score at 50%. However, among all three models, decision trees model has the least training time. But I also notice that decision trees model and random forests model both have max depth being 1, which doesn't give much explanation power. Similarly, KNN takes 1 neighbor to compute the distance, which again can't provide much generalization. Therefore, the boosting model next to above three models probably gives better prediction for this particular dataset at threshold = 50%.

- Precision at 30%:

Decision trees model (parameters = {'max_depth': 100, 'criterion': 'gini', 'min_samples_split': 10, 'max_features': 'log2'}), KNN model (parameters= {'algorithm': 'auto', 'weights': 'uniform', 'n_neighbors': 1}), and Random Forests model ({'max_depth': 1, 'min_samples_split': 2, 'n_estimators': 1, 'max_features': 'sqrt'}) altogether rank the highest in precision score at 30%.

- Recall at 50%:

The three models that have the highest precision score at 50% also give the highest score at recall score at 50%, with the recall score being about 70.1%.

- Recall at 30%:

Similar to the above case, the three models that have the highest precision score at 30% also give the highest score at recall score at 30%, with the recall score being about 28.1%.

Test window = 12 months long:
(Results are provided in *Results_Analysis_test12.ipynb*)

Results for testing window being 12-month long are almost consistent with those for testing window being 12-month, with the only difference happened in the case where decision trees model (parameter = { 'criterion': 'gini', 'max_depth': 1,

'min_samples_split': 5, 'max_features': 'sqrt'}) dominates random forests model in testing time.

The Change of Results Over Time:

The performance of models in terms of training time, testing time, accuracy, F1 score and AUC-ROC scores remain consistent as the test window goes from 6-month long to 12-month long (Shown in Table 1). Table 2 shows that the performance of models for precision score at 50%, 30%, 20% and 10% when the test window is 6-month long is consistent with the case when the test window is 12-month long. The change appears when discussing the precision score at 5%, 2%, and 1%. Boosting is the best ensemble model as the test window is 6-month long. However, Bagging dominates as the test window expands to 12-month long. This trend is also shown in calculating recall scores.

Recommendation:

Overall, decision trees model or random forests model gives the best performance under different evaluation metrics. But we should notice that this good performance often comes with selecting max depth to be 1, which barely gives much explanation in predicting the target variable by given many features. Indeed, as I'm calculating precision or recall score at 5% or less, boosting or bagging appears to be a better model. On the other hand, the high frequency that decision trees model or random forests model with max depth being 1 gets to the top of performance can indicate that probably there is a specific feature that is enough to provide sufficient information gain to predict the outcome of target variable. Which conclusion is more appropriate for this particular dataset may need further study.

As for recommendation, the results verify that starting analysis by decision trees or random forests is a good idea for they are being less time consuming and more accurate in predicting. In this particular dataset, boosting, bagging, KNN, and logistic regression can also provide reasonably good performance. As we are settled down with what the most useful features are using decision trees model or random forests model, we may consider using boosting, bagging, KNN, or logistic regression to better fit the data.

Also notice that SVM is not included in my results. This is because running SVM is extremely slow. I couldn't finish the operation even if I kept running SVM for almost a day. So it remains unknown whether SVM could provide better performance than decision tree or random forest does. But it's certain that it's the least time efficient model.

Testing window	6 months	12 months
Training time		
1	DT	DT
2	RF	RF
3	BAG	BAG
Testing time		
1	RF	DT
2	DT	RF
3	LR	LR
Accuracy		

1	RF	RF
2	DT	DT
3	BST	LR
F1 score		
1	RF	RF
2	DT	DT
3	BST	LR
AUC-ROC		
1	RF	RF
2	DT	DT
3	BST	BST

Table 1. Top 3 models in training time, testing time, accuracy score, F1 score, AUC.

Testing window	6 months	12 months	**Testing window**	6 months	12 months
P at 50%			R at 50%		
1	DT	DT	1	DT	DT
2	KNN	KNN	2	KNN	KNN
3	RF	RF	3	RF	RF
P at 30%			R at 30%		
1	DT	DT	1	DT	DT
2	KNN	KNN	2	KNN	KNN
3	RF	RF	3	RF	RF
P at 20%			R at 20%		
1	DT	DT	1	DT	DT
2	KNN	KNN	2	KNN	KNN
3	RF	RF	3	RF	RF
P at 10%			R at 10%		
1	BST	BST	1	BST	BST
2	DT	DT	2	DT	DT
3	KNN	KNN	3	KNN	KNN
P at 5%			R at 5%		
1	BST	BAG	1	BST	BAG
2	DT	BST	2	DT	BST
3	KNN	DT	3	KNN	DT
P at 2%			R at 2%		
1	BST	BAG	1	BST	BAG
2	DT	BST	2	DT	BST
3	KNN	DT	3	KNN	DT
P at 1%			R at 1%		
1	BST	BAG	1	BST	BAG
2	DT	BST	2	DT	BST
3	KNN	DT	3	KNN	DT

Table 2. Top 3 models in precision, recall scores at kth threshold