John Ferrara

Assignment 2 – DATA 622

Experimentation and Model Selection Assignment

For assignment 2 students were to experiment with different modeling techniques and tuning in order to yield a good performing model. Specifically, there were to be a minimum of two different experiments carried out for each of the three modeling techniques: decision trees, random forest, and AdaBoosting. For this assignment, I carried out two experiments using the Decision Tree methodology, three using the Random Forest technique, and four using the AdaBoosting technique. The main metric used in comparing the performance of each model generated was the PR-AUC, Precision-Recall Area Under the Curve. This metric focuses on the "yes" values, which is good in this case. This is because the "yes" and "no" values are severely imbalanced with 1,051 yes values (~11.8%) and 7,833 no values (~88.1%). Additionally, false negatives were important as these are missed clients.

Lastly, the data was taken from the output of Assignment 1, as it was cleaned to remove variables like "duration" that were not predictive, and other processing that made the data ready for modeling. This data was encoded, split into one training and test set, which was used for each experiment. Only one training and test set was derived for this assignment, as I was curious how each model would perform on their own with the same data.

Decision Tree Experiments

As mentioned, there were two different experiments were carried out using this modeling technique. The first made use of the default input values of the function, and the second the tree depth was limited to 10, the minimum number of samples per leaf was 15, and the class weights was set to balanced in order to help with "yes" predictions. These changes did improve the model when compared to the first experiment.

Algorithm	Experiment ID	Key Parameters	PR-AUC (primary)	Recall (Yes)	Precision (Yes)	F1 (Yes)	FN	ROC-AUC
Decision Tree	dec_tree1	max_depth=DEFAULT; min_samples_leaf=DEFAULT; class_weight=DEFAULT	0.174	0.330	0.288	0.308	704	0.610
Decision Tree	dec_tree2	max_depth=10; min_samples_leaf=15; class_weight=balanced	0.404	0.594	0.288	0.387	427	0.774

These changes in the second experiment, for weighting the class and limiting tree depth, etc. improved the model. As in the first experiment the imbalance of "yes"/"no" values was not considered and the tree depth was over 40. The explicit limit on depth and leaf size prevented overfitting.

Random Forest Experiments

Similar to the decision tree section, I started off with the baseline performance for the first experiment. This made use of the default inputs. However, for the second and third I attempted to refine these variables to better the model. The second introduced class weighting to handle the imbalance of "yes"/"no" values, tree depth limitation at 20, and a minimum of 5 samples before making a decision to split. The third experiment further refined those settings and yielded the best overall performance. The tree depth limit was increased to 25, the minimum sample number was increased to 20, and the number of estimators was increased to 200, from the default of 100 in the first two experiments.

Algorithm	Experiment ID	Key Parameters	PR-AUC (primary)	Recall (Yes)	Precision (Yes)	F1 (Yes)	FN	ROC-AUC
Random Forest	rand_forest1	max_depth=DEFAULT; min_samples_split=DEFAULT; n_estimators=DEFAULT; class_weight=DEFAULT	0.4239	0.2407	0.6010	0.3438	798	0.7861
Random Forest	rand_forest2	max_depth=20; min_samples_split=5; n_estimators=DEFAULT; class_weight=balanced	0.4350	0.4139	0.4971	0.4517	616	0.7958
Random Forest	rand_forest3	max_depth=25; min_samples_split=20; n_estimators=200; class_weight=balanced	0.4422	0.5243	0.4509	0.4848	500	0.8017

Overall, the baseline Random Forest already outperformed the baseline decision tree results. The refinements made on tuning the model in the second and third runs, did further improve the model. The third experiment here was actually the best performing model overall.

AdaBoost Experiments

As mentioned, there were four different experiments carried out using this modeling technique. The first made use of the default input values. The second increased weak-learner depth, adjusted the number of estimators and learning rate to capture simple interactions. The third tried a more aggressive setup that raised true positives, but also raised false positives, hurting overall PR-AUC. For the fourth I attempted to add more estimators with a lower learning rate in order to try and improve the model.

Algorithm	Experiment ID	Key Parameters	PR-AUC (primary)	Recall (Yes)	Precision (Yes)	F1 (Yes)	FN	ROC-AUC
AdaBoost	adaboost1	estimator=DEFAULT, n_estimators=DEFAULT, learning_rate = DEFAULT	0.4153	0.1808	0.6441	0.2823	861	0.7670
AdaBoost	adaboost2	max_depth via estimator= 5, n_estimators=DEFAULT, learning_rate = 1.5	0.3874	0.2769	0.5379	0.3656	760	0.7721
AdaBoost	adaboost3	max_depth via estimator= 10, n_estimators=DEFAULT, learning_rate = 2	0.2746	0.3701	0.1960	0.2563	662	0.6406
AdaBoost	adaboost4	max_depth via estimator= 10, n_estimators=300, learning_rate = 1.25	0.3879	0.2303	0.5589	0.3261	809	0.7636

Overall, the default settings in experiment one generated the strongest model when looking at PR-AUC scores. The adjustments made in experiments two through four did not surpass this first run. Additionally, the random forest 3 model remains the strongest model overall.

Experiment Choices

I opted for these experiments, as I wanted to explore what small changes to input variables did for the performance of each model type. Weighting the classes for to accommodate the imbalance of values, and playing with the leaf and tree limits was important for me to get a sense of each model's abilities. For each of these modelling techniques different responses were found. The variance for the decision tree modeling was high initially, but leveled out with some regulations on limits. Capping the maximum tree depth, minimum leaf size, number of estimators, and weighting to values to weight the "yes" values more yielded decent results. The random forest did better "out of the box" so to speak, and improved its general predictive power with increasing sampling, moderating the tree depth, and increasing the floor needed for a split. This isn't a surprise as this bagging methodology is known to perform well when generalizing predictions. AdaBoost, which is the most iterative, the weaker learner settings tended to perform better than a higher learner rate, however the increasing the inputs like n_estimators while keeping the learning rate lower, seemed to help not overfit the model.

Model Selection

After looking at the performance of all of the models, the batch of experiments using the Random Forest methodology were the best performing overall. However, looking at each category solo, we can see the best performing variation from each methodology in the table below.

Algorithm	Experiment ID	Key Parameters	PR-AUC (primary)	Recall (Yes)	Precision (Yes)	F1 (Yes)	FN	ROC-AUC
Decision Tree	dec_tree2	max_depth=10; min_samples_leaf=15; class_weight=balanced	**0.404**	**0.594**	0.288	0.387	**427**	0.774
Random Forest	rand_forest3	max_depth=25; min_samples_split=20; n_estimators=200; class_weight=balanced	**0.442**	**0.524**	0.451	0.485	**500**	0.802
AdaBoost	adaboost1	defaults (stumps, n_estimators=50, lr=1.0, SAMME.R)	**0.415**	**0.181**	0.585	0.276	**861**	0.770

Using the PR-AUC as the main metric, the best Decision Tree (dec_tree2) model reached a PR-AUC of 0.404 with Recall 0.594 (FN 427). The best AdaBoost run (adaboost1) had a PR-AUC 0.415, but had low Recall 0.181 (FN 861). The strongest overall model was rand_forest3 with a PR-AUC of 0.4422. The recall was 0.5243 (FN 500) too. The precision score for this model was 0.4509. Based on these results and the goal of trying to reduce missed “yes” clients, rand_forest3 should be our final model.