Master of Big Data Analytics

Statistical & Machine learning approaches for marketing

Individual Assignment - Task 2

IÉSEG SCHOOL OF MANAGEMENT

Andres Olivera

This project conducted a benchmark for five tree based models: Decision Tree, Bagging, Random Forest, AdaBoost and Gradient Boosting.

All models were trained on a Bank's dataset already divided into Train and Test holdout. The former is conformed of 7000 observations and 21 variables and the latter of 3000 observations and 20 variables.

The goal of this project is to predict as accurate as possible if a customer will subscribe or not.

To measure the performance of the models, different metrics were measure. A table below shows the results of models on each metric. The evaluation metric to which we will stick to select the best model is the Area Under the Curve (AUC).

Two different strategies were taken to see reaction on the performance of the models.

The first strategy is as follows:

Data process:

1. Given that the target variable was unbalanced, the train data set was oversample with the method SMOTE. Changing the unbalanced from 14% to 40% of the total observations.
2. Divide the Train Data Set into Train and Validation (were called X_train and X_test, respectively)

Feature engineering:

3. Create a new feature out of age consisting of 5 different new categories.
4. Create dummies for all the others categorical variables

Feature Selection:

5. The data was Scaled using the mean
6. PCA was used as the method to reduce dimensionality and 3 variables were selected.

The Second one:

Data process:

1. Given that the target variable was unbalanced, the train data set was oversample with the method SMOTE. Changing the unbalanced from 14% to 40% of the total observations.
2. Divide the Train Data Set into Train and Validation (were called X_train_2 and X_test_2, respectively)

Feature engineering:

3. Create a new feature out of age consisting of 5 different new categories.
4. All categorical variables were transformed into numerical using **Feature Encoding**

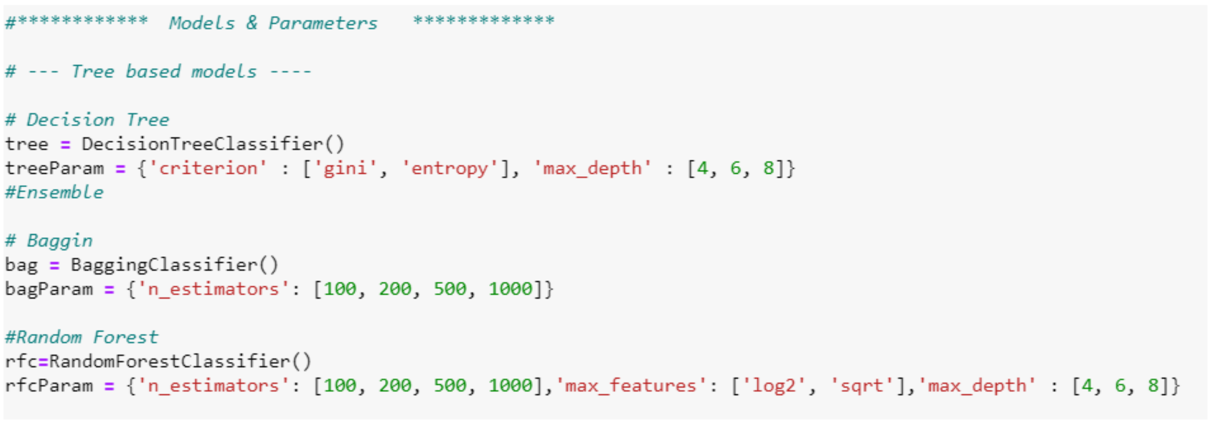
Feature Selection:

5. The data was Scaled using the mean
6. **RFE** was used as the method to reduce dimensionality and **10** variables were selected.

Hyperparameter Tunning:

To find the best hyperparamters of each model in order to get their best performance, I used 4-fold Cross Validation.

The hyperparamets used in each model are the following:



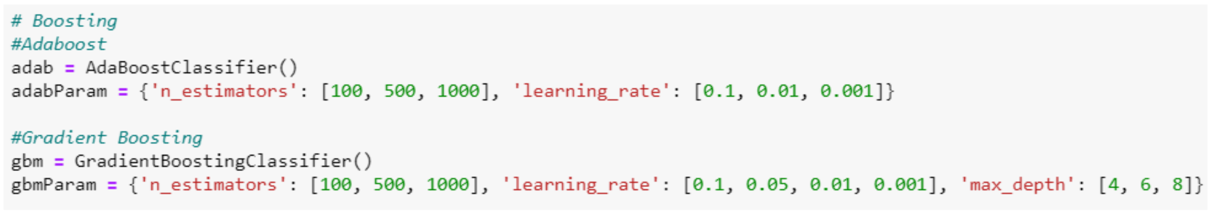
```
#************  Models & Parameters   ************

# --- Tree based models ----

# Decision Tree
tree = DecisionTreeClassifier()
treeParam = {'criterion' : ['gini', 'entropy'], 'max_depth' : [4, 6, 8]}
#Ensemble

# Baggin
bag = BaggingClassifier()
bagParam = {'n_estimators': [100, 200, 500, 1000]}

#Random Forest
rfc=RandomForestClassifier()
rfcParam = {'n_estimators': [100, 200, 500, 1000],'max_features': ['log2', 'sqrt'],'max_depth' : [4, 6, 8]}
```



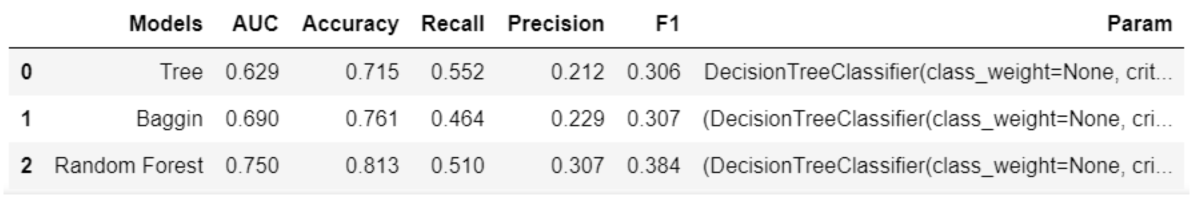
```
# Boosting
#Adaboost
adab = AdaBoostClassifier()
adabParam = {'n_estimators': [100, 500, 1000], 'learning_rate': [0.1, 0.01, 0.001]}

#Gradient Boosting
gbm = GradientBoostingClassifier()
gbmParam = {'n_estimators': [100, 500, 1000], 'learning_rate': [0.1, 0.05, 0.01, 0.001], 'max_depth': [4, 6, 8]}
```

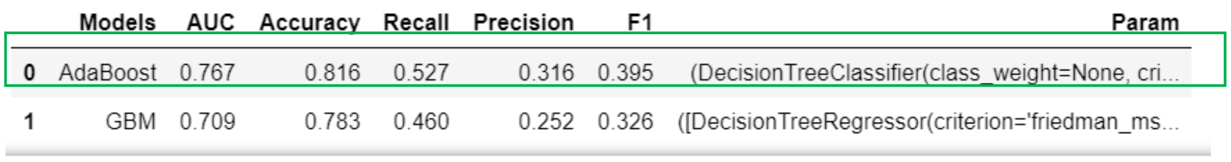
Results:

First Strategy:

The performance metrics of all models using the method one are below. As can be seen, Adaboost outperform the results of the models on every metric except for F1. It had a 76.7% of AUC and an 81.6% of accuracy while Random Forest had a 75% AUC and 81.3% of accuracy.



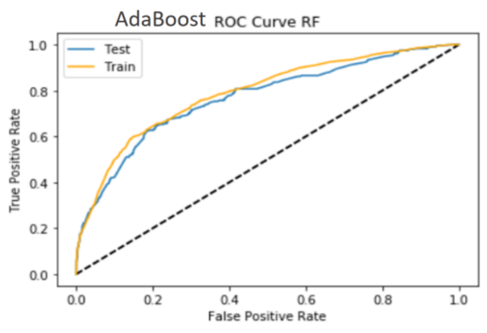
	Models	AUC	Accuracy	Recall	Precision	F1	Param
0	Tree	0.629	0.715	0.552	0.212	0.306	DecisionTreeClassifier(class_weight=None, crit...
1	Baggin	0.690	0.761	0.464	0.229	0.307	(DecisionTreeClassifier(class_weight=None, cri...
2	Random Forest	0.750	0.813	0.510	0.307	0.384	(DecisionTreeClassifier(class_weight=None, cri...

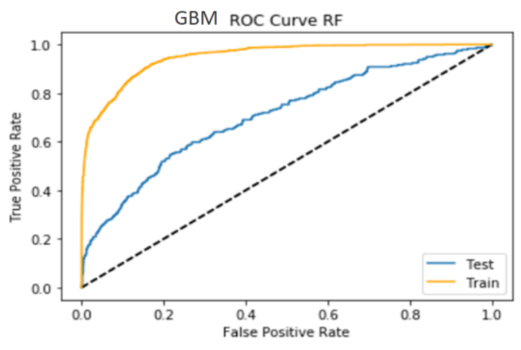


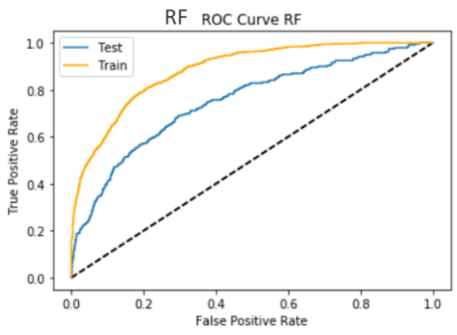
	Models	AUC	Accuracy	Recall	Precision	F1	Param
0	AdaBoost	0.767	0.816	0.527	0.316	0.395	(DecisionTreeClassifier(class_weight=None, cri...
1	GBM	0.709	0.783	0.460	0.252	0.326	([DecisionTreeRegressor(criterion='friedman_ms...

On the graphs below, we can the best three models performances.

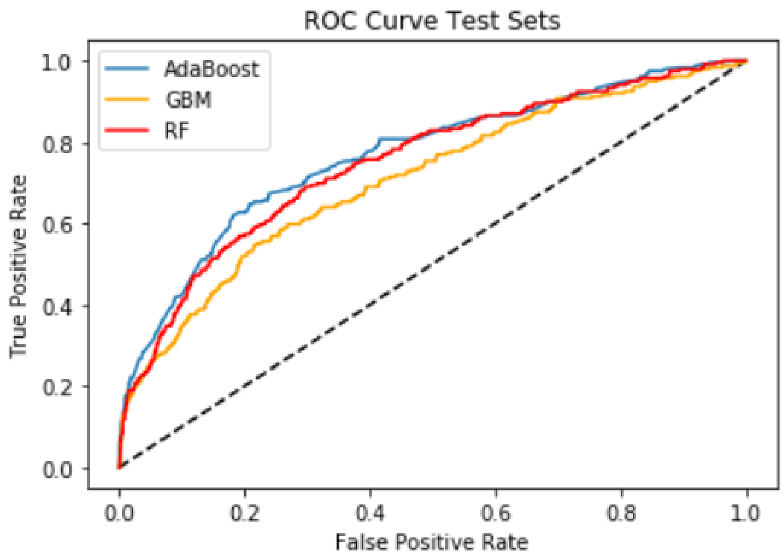
On the first graph, corresponding to AdaBoost, is on the one hand the model with the highest evaluations metrics. On the other, is practically no overfitting at all. While on the contrary, both Random Forest and Gradiend Boosting overfit dramatically.







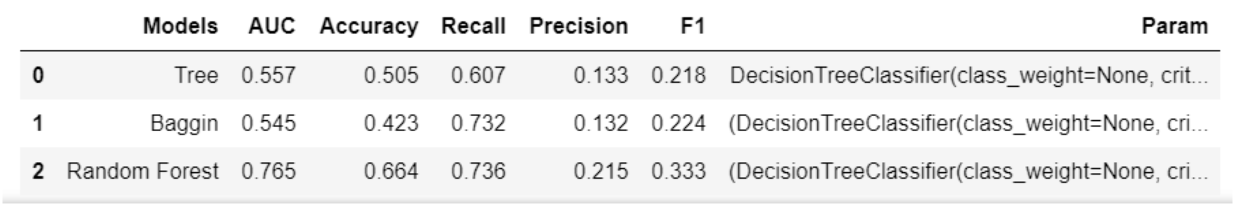
On this graph, we can see the different performance on the test set. Adaboost, not for a lot, is better than Random Forest. However, as we saw on the graphs before, AdaBoost, besides that has a slightly better AUC, should be picked because is the most stable model. This model was able to get such an accurate score without overfitting its training data set. In other words, Adaboost has a good combination of Bias-Variance, and it's clear than in this case the variance is the low.



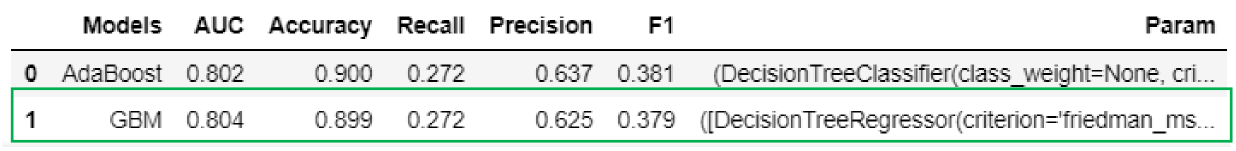
Second Strategy:

On the second procedure, the score of both Tree and Bagging lowered down drastically. However, AdaBoost and GBM manage to get an AUC and Accuracy over 80%. The only downside is that the recall, which means that only it is miss classifying the clients that will not subscribe. On the other hand, prediction goes up because usually there is a tradeoff between the two metrics, when one goes up the other goes down. That means that the models is predicting very well the clients that will no subscribe.

In general, those are good results. However, the model is pumping up the scores thanks of the class in which we have less interest in, the clients that won't subscribe. The model should be getting high scores, or at least the ideal should be that it classifies very well and even better the clients that will subscribe than the ones that will nor.

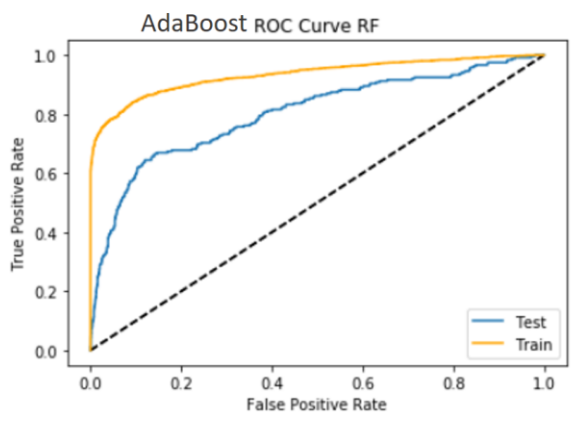


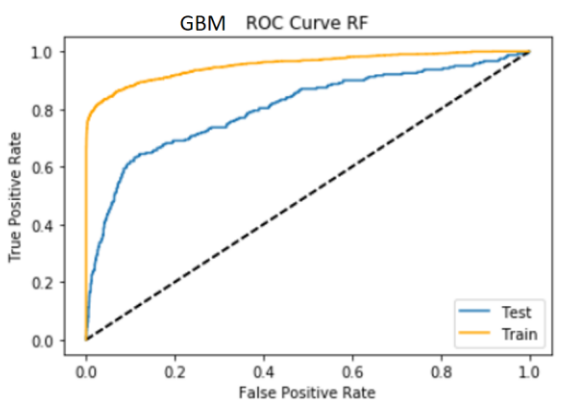
	Models	AUC	Accuracy	Recall	Precision	F1	Param
0	Tree	0.557	0.505	0.607	0.133	0.218	DecisionTreeClassifier(class_weight=None, crit...
1	Baggin	0.545	0.423	0.732	0.132	0.224	(DecisionTreeClassifier(class_weight=None, cri...
2	Random Forest	0.765	0.664	0.736	0.215	0.333	(DecisionTreeClassifier(class_weight=None, cri...



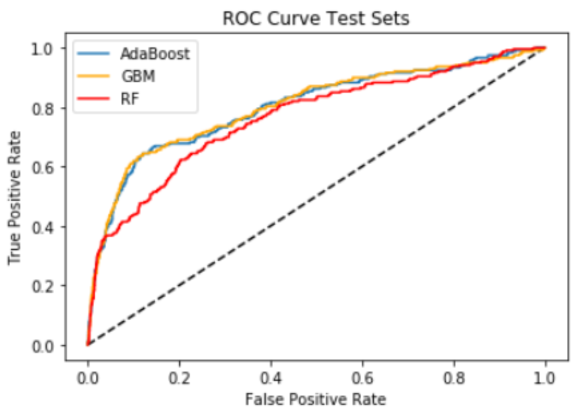
	Models	AUC	Accuracy	Recall	Precision	F1	Param
0	AdaBoost	0.802	0.900	0.272	0.637	0.381	(DecisionTreeClassifier(class_weight=None, cri...
1	GBM	0.804	0.899	0.272	0.625	0.379	([DecisionTreeRegressor(criterion='friedman_ms...

On the graphs below, we can see that both models are overfitting. A little bit less Adaboost but it still has a lot of variance.





Overall, AdaBoost and GBM are the models that perfomed the best in both procedures. Below, we can see that



Conclusion:

The benchmark showed that with the second strategy the models performed better. Among the best, AdaBoost is best perfermor followed by Gradiend Boosting.

The hyperparameter selected in the Cross Validation can be seen below. It’s interesting to note that while Adaboost needed 1000 trees to get its best fit, Gradient Boosting only needed 100. On the other hand, for the two models, the learning_rate was 0.1.

```
AdaBoostClassifier( Learning_rate=0.1,
                    n_estimators=1000
                     )


GradientBoostingClassifier( Learning_rate=0.1,
                            max_depth=4,
                            max_features='sqrt',
                            n_estimators=100,
                             )
```