Predicting a Customer's Satisfaction

1. Project Overview

Which customers are happy customers?

Customer satisfaction is a key measure of success and unhappy customers don't stick around. What's more, unhappy customers rarely voice their dissatisfaction before leaving.

Santander Bank is asking for help in predicting dissatisfied customers early in their relationship. Doing so would allow Santander to take proactive steps to improve a customer's happiness before it's too late.

This data set has hundreds of anonymized features to predict if a customer is satisfied or dissatisfied with their banking experience.

The steps taken to predict whether a customer was satisfied with their service or not was:

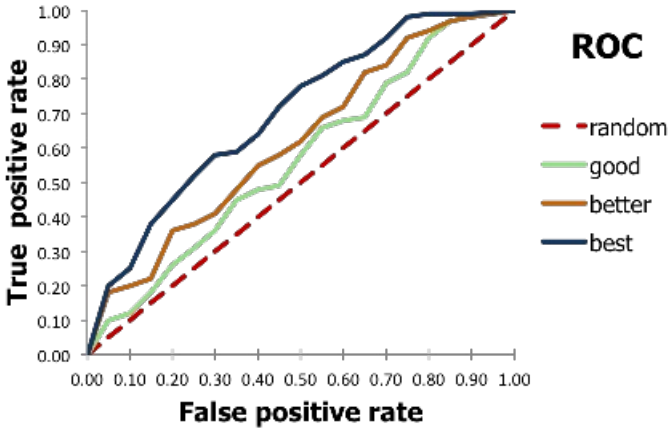
> 1. Define the metric that will be used to evaluate the performance of the model.
> 2. Select a set of models to train and from that set determine the best model to set as a benchmark.
> 3. Fine tune the best model and validate it through cross validation.

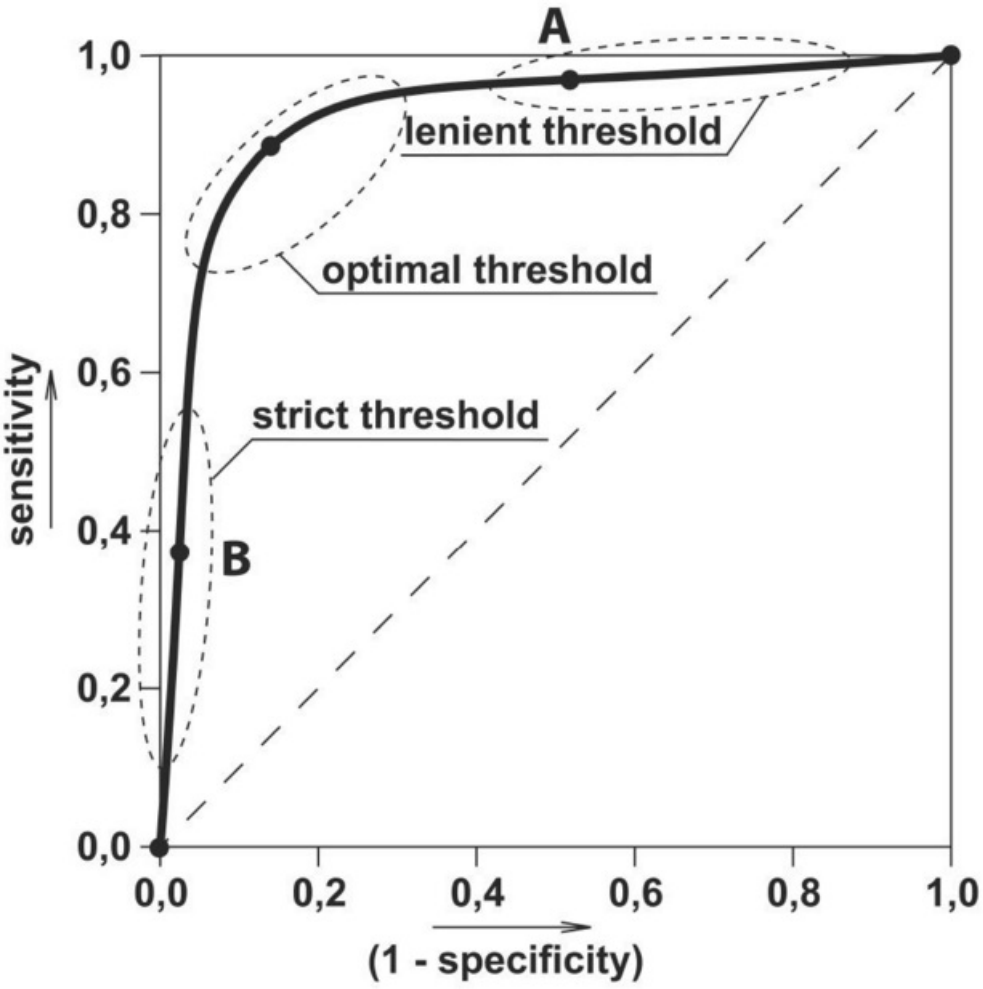
The data set can be found at https://www.kaggle.com/c/santander-customer-satisfaction/data (https://www.kaggle.com/c/santander-customer-satisfaction/data)

2. Metrics

Because trying to predict whether a customer is satisfied or unsatisfied with their service, this becomes a supervised, classification problem. The metric that is used to evaluate the model is the Area Under the Receiver Operating Characteristic (AUROC).

The receiver operating characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, or recall. The false-positive rate is also known as the fall-out and can be calculated as (1 - specificity). The ROC curve is thus the sensitivity as a function of fall-out. The advantage of the roc curve is that it explores all possible setting of the threshold.





3. Analysis

Data Exploration

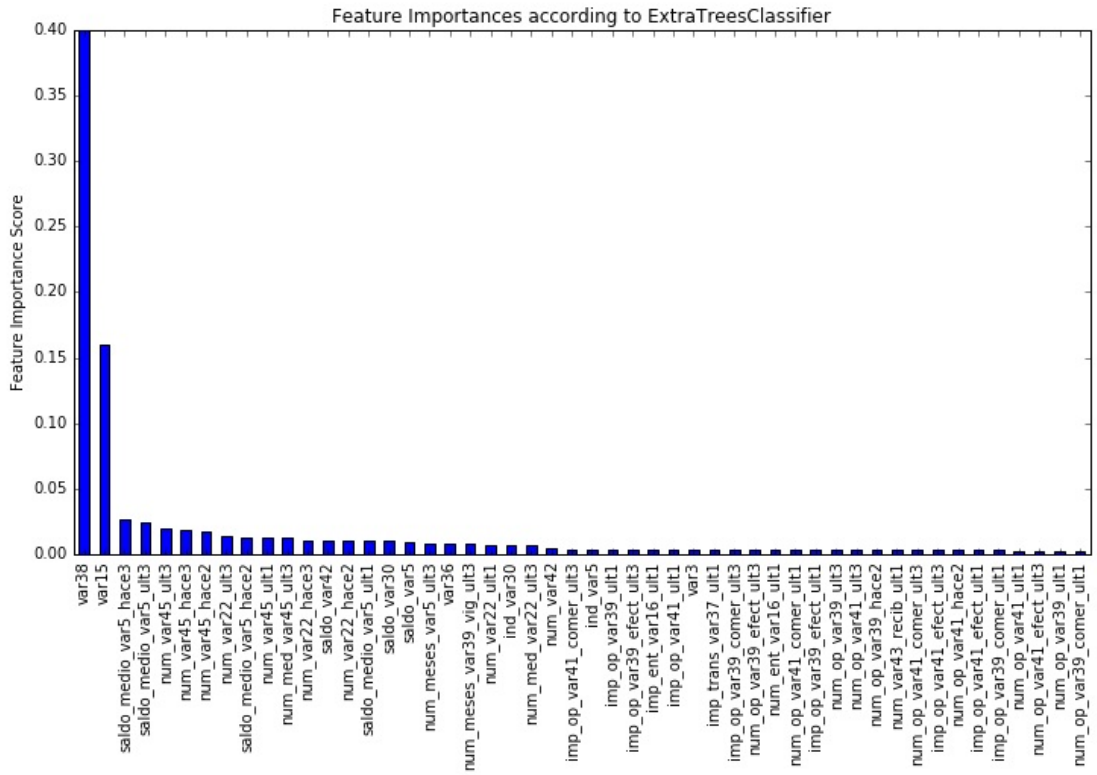
The data set appears to very clean with the exception of a lot of 0 values. The table below illustrates the data prior to being clean.

	Data Summary (Pre Cleaning)
# of Entries:	76020
# of Features:	371
# of Satisfied	73012
# of Unsatisfied	3008

Throughout my data auditing, I found that there were many duplicate features and constant features. I dealt with this by simply removing these columns from the data set. After removing the previously mention features, only the the most important features were kept. The table below illustrates the data after being cleaned.

	Data Summary (After data cleaning)
# of Entries:	76020
# of Features:	39
# of Satisfied	73012
# of Unsatisfied	3008

The graph below shows the top 50 most important features. By reducing the features, training time of the models will be increased.



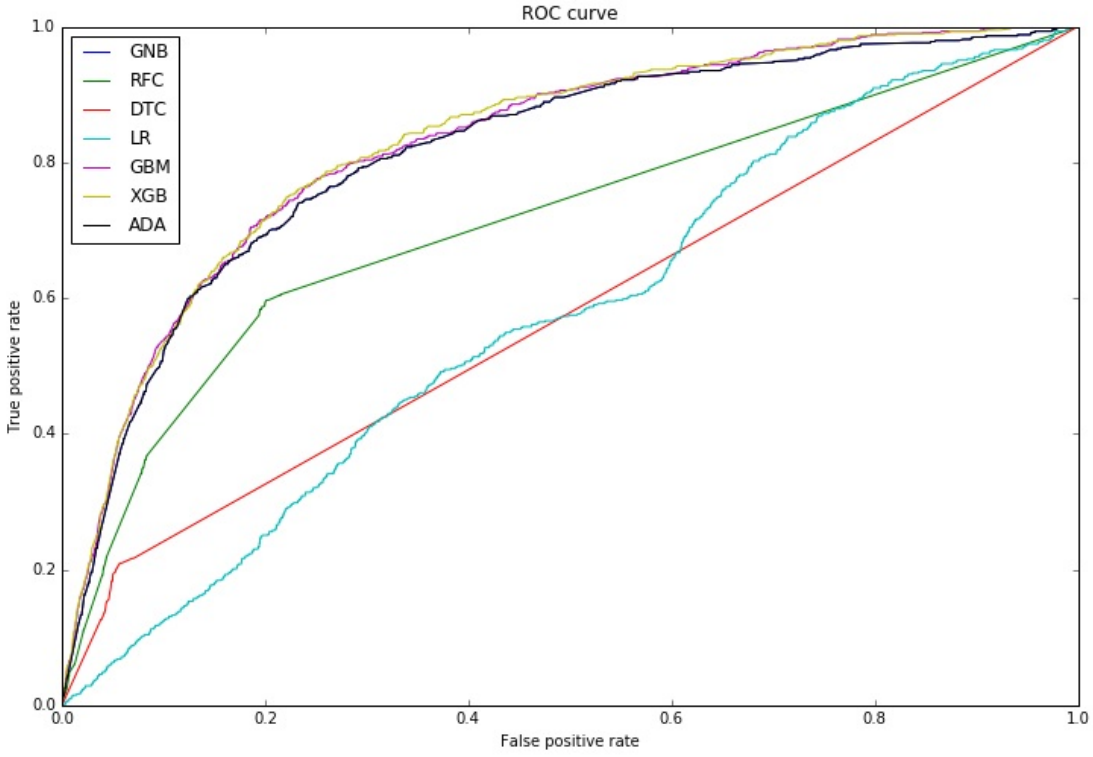
Algorithms and Techniques

In trying to find the best model, seven different untuned classifiers were used. The performance of each model can be seen on the table below.

Classifier	Train Time (sec)	AUROC Score (Train)	Test Time (sec)	AUROC Score (Test)
Naive Bayes	0.45	0.745	0.08	0.625
Random Forest	9.68	0.679	1.32	0.677
Decision Tree	8.25	0.575	1.23	0.564
Logistic Regression	45.58	0.580	8.22	0.582
Gradient Boosting	111.50	0.835	14.89	0.821
Extreme Gradient Boosting	13.55	0.837	5.67	0.826
Ada Boosting	36.23	0.829	6.17	0.810

As mentioned previously, the AUROC score was used to measure performance of each algorithm. Therefore, a 10-fold cross-validation method was used. How this works is one fold of the data set is chosen as the test set, and the rest is chosen training set. The process is repeated 10 times and the average score is reported. The running time includes the total time of training and testing time in cross-validation process.

Here we can see that classifiers random forest the top three scores were Extreme Gradient Boosting (XGB), gradient boosting (GBM), and adaboosting. With a slightly higher score than gradient boosting, the model that seemed to perform the best was XGB. Not only was the score slightly higher, but XGB was blazingly faster than GBM.



Here we can see on the ROC Curve that gradient boosting and Extreme Gradient Boosting are nearly identical. We can also see by their position that these are the best two models to use.

Benchmark

Since XGB had the highest score and a faster time than GBM, an untuned XGB will become the benchmark for this model with a training score of 0.837.

4. Methodology

Now knowing what model that will be used, fine tuning the parameters would be the next step.

XGB parameters can be divided into three catergories:

1. General Parameters: Parameters that define the overall functionality of XGBoost.
2. Booster Parameters: Guides the individual booster (tree/regression) at each step.
3. Learning Task Parameters: Parameters used to define the optimization objective the metric to be calculated at each step.

To achieve the best possible model, a variety of different combinations were used to increase the AUROC score.

The parameters that were used to tune the XGB algorithm was:

min_child_weight [default=1]: Defines the minimum sum of weights of all observations required in a child.

max_depth [default=6]: The maximum depth of a tree

gamma [default=0]: A node is split only when the resulting split gives a positive reduction in the loss function.

subsample [default=1]: Denotes the fraction of observations to be randomly samples for each tree.

colsample_bytree [default=1]: Denotes the fraction of columns to be randomly samples for each tree.

alpha [default=0]: Can be used in case of very high dimensionality so that the algorithm runs faster when implemented

The below table illustrates the parameters used in the final model.

Final XGB Model Score	**Prev Score**	**New Score**
Training	0.837	0.841
Test	0.826	0.837

6. Conclusion

Extreme gradient boosting was chosen as the best algorithm to use for this data set. For the final model the training set improved to a 0.841 from 0.837 through tuning the parameters of the XGB model. I got a test score fo 0.837, 0.011 higher than the untuned test score. Although that's not significantly higher it is an improvement. With the data set being anoymous, I find it very hard to do be able to visualized the data and perform an analysis on it.

7. Reflection

I really enjoyed working on this project. I found it to very interesting in trying to predict if a customer will be satisfied or not with a company's service. With the data set being anonymized, it makes it very hard to explore what each feature actually means and to truly explain the true importance of a feature to another. I also found out that the more complexed ensemble method had better performance scores than simple classifiers. Although the tuning of the model's parameters increased the the final score, methods like feature engineering, creating ensemble of models, stacking, etc may improve the model significantly. Although this would be a good thing, it may take a long time to train.

Reference

https://www.kaggle.com/c/santander-customer-satisfaction (https://www.kaggle.com/c/santander-customer-satisfaction)

http://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/ (http://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/)

https://en.wikipedia.org/wiki/Receiver_operating_characteristic (https://en.wikipedia.org/wiki/Receiver_operating_characteristic)

In [ ]: