**May**

2017

**Springboard Second Capstone project**

Predicting Customer Satisfaction at Santander

Yi Zhao

08

**Fall**

# Background and Description of Problem

Santander is a retail and commercial bank founded in 1856 with headquarters in the U.K. They have 125 million customers worldwide. Customer satisfaction is a key measure of their success and is a strong indicator to future customer retention. Identifying unhappy customers and finding ways to improve their satisfaction will prevent them from leaving Santander. Though, unhappy customers rarely voice their dissatisfaction before leaving.

Thus, it is crucial for Santander to predict and identify dissatisfied customers early in their relationship in order to take proactive steps to improve a customer's happiness before it's too late.

# Data and Description of Features

There are in total 76020 observations and 370 predictive features.[[1]](#footnote-1) It is assumed that are the observations are independent with each other and each represent a customer.

The "TARGET" column is customer satisfaction that we want to predict. It equals 1 for dissatisfied customers and 0 for satisfied customers. 3008 out of 76020 customers are dissatisfied, or only 4%.

Some variables have hidden missing values, such as "-999999" as a minimum value and “99” as a maximum value of a variable with all other values being 0, 1, 2, 3, 4, 5. Those missing values are replaced with NA.

Variables with 20 or less unique values among all 76020 observations were assumed to be categorical variables. Even though it was unclear whether they were ordered variables or not, they were spread into columns of dummy variables, or binary variable with zero and one entries. For each original variable that was transformed, there was one last column omitted from all its binary columns.

Among features with greater than 20 unique values, only three have a relatively small number of "0", and the rest of the features have 20K+ or over 26% of observations being zero. And the remaining 100 of those numerical predictor variables (> 20 unique values) have 60K+ observations with zero values, which is 79% of the sample size.

After taking all feature transformations, there were 851 features. Omitting all observations with missing values reduced the dataset from 76020 to 45908. Since there were only two features with missing values, those two features were omitted instead of omitting observations. In the rest of the analyses, there were 76020 observations and 849 predictive features used.

Usually for classification problems, numeric variables are centered (subtract the mean) and scaled (divide by the standard deviation) in a regression analysis so that those with a larger range of values would not have an extremely small coefficient. The centering and scaling assume the variables to have a normal distribution. However, many of the features in this dataset did not have a normal distribution. Often, they resemble more of a bimodal distribution with some values a few thousand times as high as others. To resolve this issue, min max scaling was used instead: where is a feature and is its index. In the rest of this report, a scaling transformation to the features would refer to a min max scaling.

# Feature Selection and Analysis

There were two major feature selection methods used: selecting K best features from chi-squared feature selection and Principal Components Analysis (PCA). 5-fold cross-validation and grid search was used to optimize the number of features in chi-squared feature selection and the number of components in PCA. The model evaluation used was area under the curve of the ROC, which is further explained in the next session. The optimal K, or number of features, was 100, and the optimal number of component was 5 on the original dataset and 60 on the scaled dataset. This would mean that instead of having 849 features, there would be only 100 from chi-squared method, and 5 dimensions from PCA.

# Binary Classification and Model Evaluation

Logistic regression is a common method for binary classification that predicts the probability of an observation being 1. In this case, we are predicting the probability of a customer being dissatisfied. Usually, a threshold of 50% or 0.5 is used to classify observations with a predicted probability of 50% or above being 1 and else being 0. However, a higher or lower threshold may lead to better model performance. The Receiver Operating Characteristics (ROC) curve is a good way of showing the change in model performances as the probability threshold or discrimination threshold changes. The curve plots the true positive rate (TPR) versus the false positive rate (FPR) at various probability thresholds. TPR is computed as , or the fraction of all actual positives that were predicted to be positive. FPR is computed as , or the fraction of all the actual negative values that were wrongly predicted as positive. The true-positive rate is also known as sensitivity, recall or probability of detection, while the false-positive rate is also known as the fall-out or probability of false alarm. If a classification is perfect, the TPR would be 1 when FPR is 0, and the area under the ROC curve would be 1. The higher the area under the ROC curve (AUC), the better the classification result is.

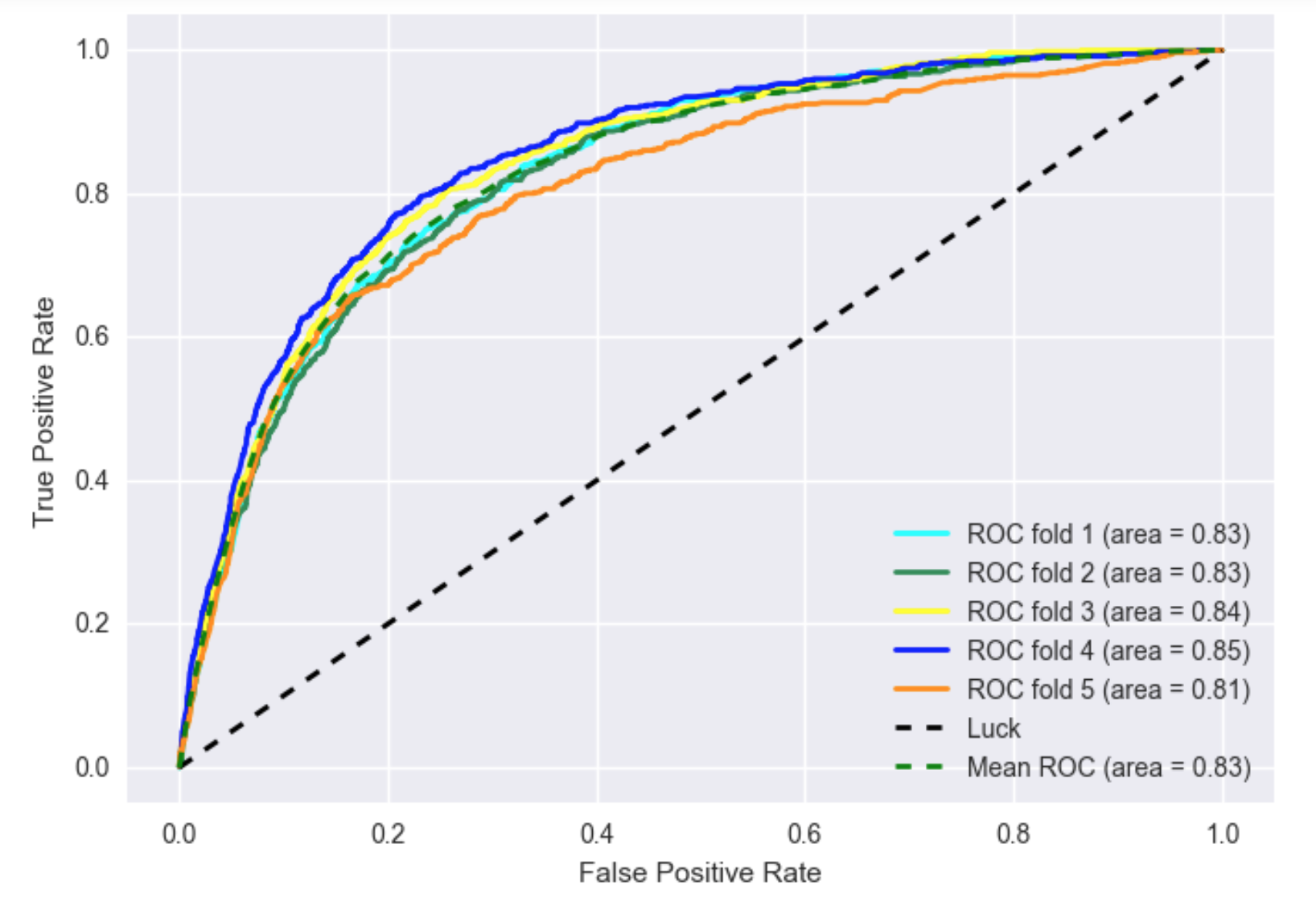
In order to benchmark the performance of our model, I generated a randomized set of features by rearranging the order of the observations with their corresponding predictor variables . The order of the predicted values remained unchanged, while all the predictive features were resampled without replacement, so there shall be no relationship between the predictive features and the predicted values. When I split this randomized dataset into a training set and a testing set and fit a logistic regression to the training set, the resulting AUC was 0.49, or less than 0.5. The average of AUC from running a 5-fold cross validation was still 0.50. This would be expected, as in a random scenario, the probability of guessing the head or tail correctly from flipping a coin would be 50%.

In addition to logistic regression, Naïve Bayes is a similar binary classification method that also uses one single linear boundary. Decision tree based ensemble models are different in that each tree partitions the feature space into half-spaces with each split and resulting in multiple linear decision boundaries. Those models are also used to compare with the results from logistic regression. The testing set mean AUC from 5-fold cross validation of different models and different feature space is summarized in the table below. Note that the maximum depth parameter in the XGBoost and GradientBoost models were optimized using grid search for values of 1, 2, 3, 4, 5 and10.

Table 1 Area under the ROC curve for the testing set of various model and feature selection combinations

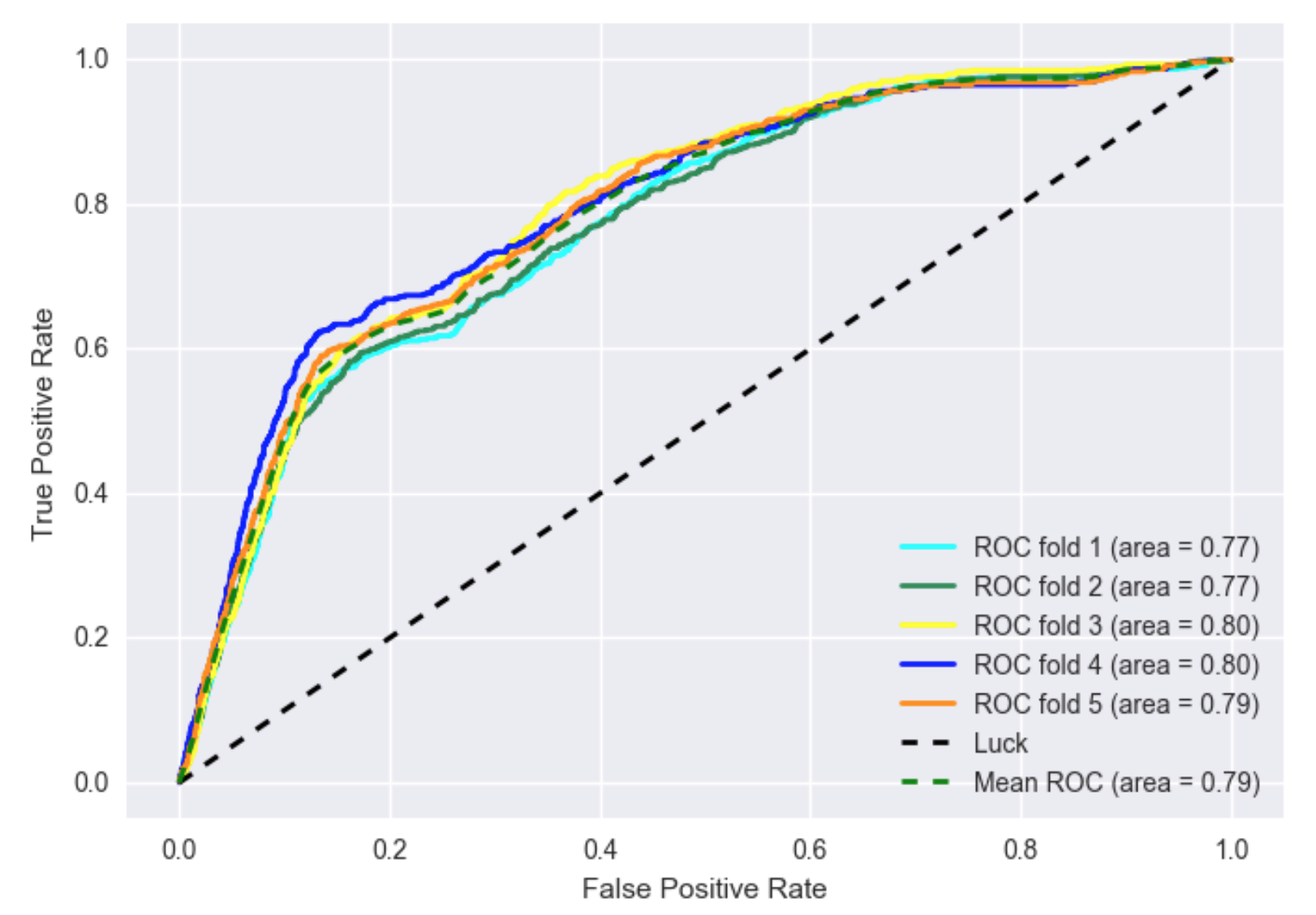
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Original features | Scaled features | Chi-squared chosen top 100 scaled features | PCA chosen top 5 original features | PCA chosen 60 scaled features |
| Logistic Regression | 0.599 | 0.786 | 0.774 | 0.595 | 0.776 |
| Random Forest | 0.686 | 0.670 | 0.705 | -- | 0.647 |
| XGBoost (Maximum depth = 3) | 0.832 | 0.832 | 0.807 | -- | 0.804 |
| GradientBoost (Maximum depth = 3) | 0.707 | 0.773 | 0.805 | -- | 0.803 |
| Bernoulli Naïve Bayes | 0.713 | 0.704 | 0.714 | -- | 0.757 |

As shown in Table 1, the min max scale transformation significantly improved results of Logistic Regression, but not the other classifiers. Feature selection with Chi-squared or PCA did not improve result except of Bernoulli Naïve Bayes and XGBoost. Since there is built in bootstrap resampling in ensemble trees, it is not surprising that feature selection was not helpful. Overall, the XGBoost had the best performances.

Figure ROC curves computed from 5-fold cross validation of XGBoost model

XGBoost is an improved version of gradient boost classifier. So the best performing decision tree based classifier was XGBoost, and Logistic Regression was a best single discrimination boundary classifier. The ROC curves of those two models are shown in Figure 1 and Figure 2. The black dashed line represents the performance of a benchmark random guess model. According to the ROC curves, logistic regression has a lower true positive rate (TPR) for the range of false positive rate (FPR) 0.18 to 0.6.

Figure ROC curves of Logistic Regression with MinMaxScaling



In order to further compare the two models, both models were run on the same individual training and testing set split. Similar to ROC curve as the TPR and FPR differs for different probability threshold of classifying as positive, the precision and recall are also metrics of classifier performance that change with probability threshold. Recall is the same as TPR, . And the precision is . In addition to recall and precision, F1-score synthesizes the information of both as . These metrics are plotted against the probability threshold, and precision is plotted against recall for XGBoost model and the Logistic Regression model (Figure 3, Figure 4).

The maximum F1-score is slightly higher in XGBoost model, and precision is higher for a recall score of lower than 0.3 (Fig. 3-4). When F1-score is the highest in both models, the recall score is about the same for XGBoost (0.48) and Logistic Regression (0.51), while the precision score higher for XGBoost (0.19) compared to Logistic Regression (0.16). The testing set confusion matrix of both models at their optimal F1-score shows that the number of true positives (TP) are similar in both models, but the number of false positives (FP) is much smaller in XGBoost (Table 2, Table 3).

Table Confusion Matrix of XGBoost classifier at probability threshold 0.13

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted  True | Predicted  False | Total  Actual |
| Actual  True | 357 (TPR = 48%) | 390 | 747 |
| Actual  False | 1489 (FPR = 8%) | 16769 | 18258 |
| Total  Predicted | 1846 | 17159 | 19005 |

Table Confusion Matrix of Logistic Regression classifier at probability threshold 0.08

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted  True | Predicted  False | Total  Actual |
| Actual  True | 379 (TPR = 51%) | 368 | 747 |
| Actual  False | 2003 (FPR = 11%) | 16255 | 18258 |
| Total  Predicted | 2382 | 16623 | 19005 |

Figure Precision, recall and F1-score of XGBoost classifier at a range of probability thresholds

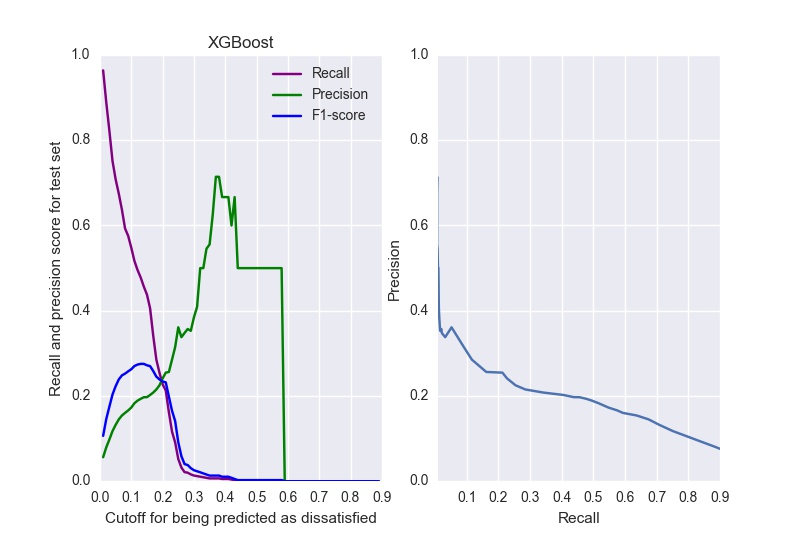
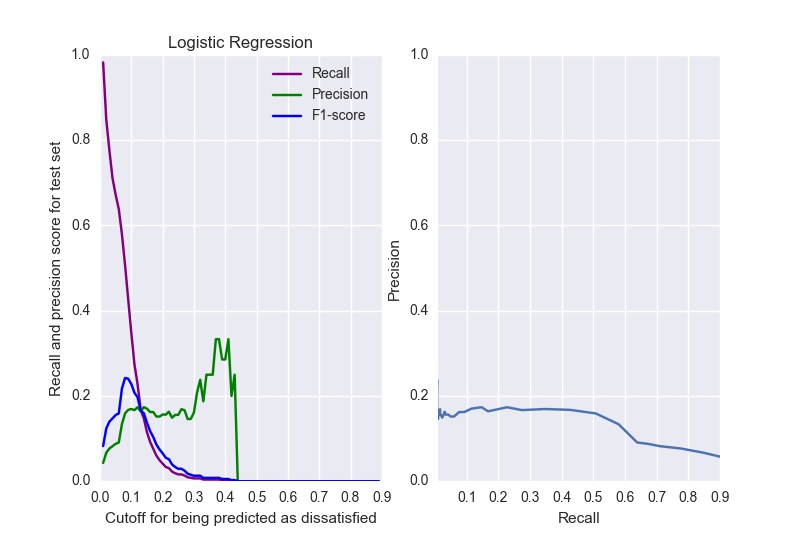


Figure Precision, recall and F1-score of logistic regression

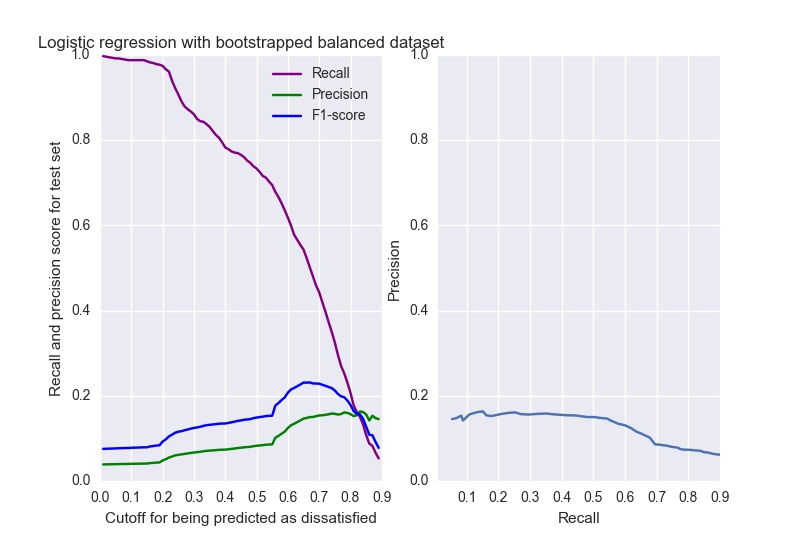


The XGBoost classifier used a threshold of 0.13, so observations with a probability of higher than 13% being dissatisfied customers were classified as dissatisfied customers. This probability threshold for logistic regression was 8%. These thresholds were small because there was only 4% of the raw dataset being dissatisfied customers. When I used bootstrap resampling to resample the same number of observations with a target of 0 and 1, the resulting optimal probability threshold for maximizing F1-score became 66% (Figure 5). Nevertheless, the recall rate and F1-score were not different from without using the bootstrap resampling (Figure 5). Precision (0.15) and recall (0.52) were similar to without the bootstrap as well (Table 4).

Table Confusion matrix of logistic regression with balanced bootstrapped dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted  True | Predicted  False | Total  Actual |
| Actual  True | 389 (TPR = 52%) | 358 | 747 |
| Actual  False | 2233 (FPR = 12%) | 16025 | 18258 |
| Total  Predicted | 2622 | 16383 | 19005 |

Figure Precision, recall and F1-score of bootstrapped and thus balanced dataset



The sensitivity of our results to the way we defined categorical variables was tested by varying the minimum number of unique values for considering a variable as numerical. Using the same logistic regression model and the min max scaling transformation, the resulting AUC of the ROC curve was similar for not distinguishing categorical versus numerical variables, using a threshold of 10 unique values, 20 and 30 unique values. The difference in AUC was within 0.003.

# Conclusion

Given an anonymized dataset, transformation of those predictor variables is crucial to building high performance predictive models. In this analysis, variables with less than 20 unique values out of 76K observations were considered to be categorical and transformed into binary variables. Overall, a simple logistic regression predicts dissatisfied customers of Santander’s at a reasonable performance level, outperforming Naïve Bayes and Random Forest models. The extreme gradient ensemble classifier, XGBoost, has a better performance overall in terms of AUC and F1-score, but the recall rate is no better than logistic regression. For Santander, recall is more important than precision or other measures, because the number of False Positives is less alarming than the number of False Negatives. Falsely identifying satisfied customers as dissatisfied and trying to improve service to those that were not going to leave would not hurt. However, falsely ignoring the dissatisfied customers, or identifying them as satisfied customers, would be much more hurtful. The higher the probability threshold, the better recall rate becomes, but precision also drops in the meantime. The question is how much it costs Santander to improve satisfaction of customers or to prevent a customer from potentially leaving. If the marginal cost of each customer is low, there would be a higher return to Santander to insure a higher recall rate at the cost of more falsely identified dissatisfied customers to please. However, if the marginal cost is high, the optimal probability threshold for identifying dissatisfied customers would be lower, or a lower recall rate but higher precision. In both scenarios, there need to be an assessment of both the potential monetary gain from preventing each dissatisfied customer from leaving and the expected cost of improving satisfaction of each extra falsely identified dissatisfied customer.

1. https://www.kaggle.com/c/santander-customer-satisfaction/data [↑](#footnote-ref-1)