Predicting Customer Satisfaction and Dealing with Imbalanced Data

Adrián Vallés Iñarrea

1st reader: Dr Dungang Liu

2nd reader: Dr Shaobo Li

MS Business Analytics Candidate 2018

Department of Operations, Business Analytics & Information Systems

Carl H Lindner School of Business: University of Cincinnati

Abstract

From frontline support teams to C-suites, customer satisfaction is a key measure of success. Unhappy customers don't stick around. On this paper, we will compare logistic regression, classification tree, random forest and extreme gradient boosting models to predict whether a customer is satisfied or dissatisfied with their banking experience. Doing so would allow banks to take proactive steps to improve a customer's happiness before it's too late. The dataset was published in Kaggle by Santander bank, a Spanish banking group with operations across Europe, South America, North America and Asia. It is composed of 76020 observations and 371 variables that have been semi-anonymized to protect the client's information. 96.05% of the customers are satisfied and only 3.95% are dissatisfied, making this classification problem to be highly imbalanced. Since most of the commonly used classification algorithms do not work well for imbalanced problems, we also compare in this paper two ways to deal with the imbalanced data classification issue. One is based on cost sensitive learning, and the other is based on a sampling technique. Both methods are shown to improve the prediction accuracy of the minority class, and have favorable performance comparing to the existing algorithms.

Table of Contents

1 Background and Goal

Customer satisfaction is key in creating a long-term relationship with your customers. Not only do loyal customers spend more long term, but it's actually cheaper to keep them happy than trying to acquire new customers. It's in fact 6 times more expensive to acquire a new customer than retaining an existing one. Satisfied customers will also recommend you to their network. On the other hand, unhappy customers don't stick around. Santander Bank is trying to identify 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. The dataset contains 371 semi-anonymized variables and 76020 observations. Among the total 76020 customers, 3003 (3.95%) are dissatisfied customers. The response variable of this study is TARGET (dissatisfied customer=1, Satisfied customer=0). Our main goal for this project is to build an accurate classifier to predict if a customer is satisfied or dissatisfied with their banking experience.

To ensure that the selected model predicts well for data not used to build the model, we use model validation on a testing sample. We will therefore build different models (e.g. Logistic regression, classification tree, random forest, XGboost) using the training sample, compare their performance on the testing sample, and select the best-performing model based on the area under the ROC curve, misclassification rate, false negative rate, precision and recall.

2 Data Description

The dataset contains 371 semi-anonymized variables and 76020 observations. Among the total 76020 customers, 3003 (3.95%) are dissatisfied customers. The response variable of this study is the binary variable TARGET (Dissatisfied customer=1, Satisfied customer=0). A plot summarizing the satisfaction of the customers is shown below in **Figure 1.**

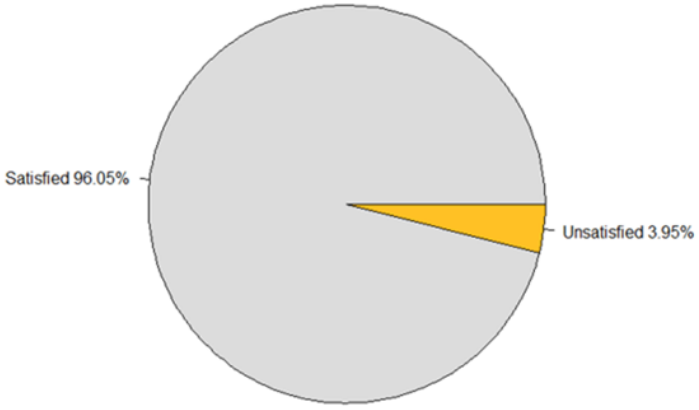

Figure 1 – Santander customer satisfaction distribution

Regarding the quality of the data, there are some important considerations that we need to take into account for a better understanding of the data:

- Sparse data (97% of the values are zero).
- Highly correlated variables.
- Although there are no duplicated observations, there are variables that have the same values for all the observations (duplicated columns).

- Some columns have variance of 0 meaning that they have the same exact value for every observation.
- Although there are not missing values, there are some abnormal values (i.e. 9999999999 for 6 different variables).

Data cleaning is therefore crucial in this dataset. Given the large number of variables, it would be computational too expensive to use all the variables for the modeling stage.

As stated earlier, while this dataset doesn't contain missing values or duplicated observations, we still need to take some actions to get rid of some insignificant variables. First, we are getting rid of the variable ID, which is a simple count of the customers and is therefore useless for the modeling stage. Next, some of the variables have only values of 0's on their observations. This means that these features have zero variance and therefore we can drop them since they are also useless for modeling. We drop the duplicated columns as well.

Finally, given that we have so many variables that are highly correlated, we will delete those with a correlation higher than 0.7 to reduce potential multicollinearity in the modeling stage. This is a measure taken due to the very large number of correlated variables. For this, we have used the findCorrelation function in the caret package in R that considers the pair-wise correlations. If two variables have a high correlation, the function looks at the mean absolute correlation of each variable and removes the variable with the largest mean absolute correlation. By doing all the previous steps, we have decreased the number of variables from the initial 371 to 104.

While we have considerably reduced the number of variables, 104 variables are still too many to start the modeling stage. An alternative to this problem is to do feature selection before we get into modeling. We use feature selection to help cut down on runtime and eliminate unnecessary variables prior to building a prediction model. We will apply random forest to the 104 variables to see what features are more important, and we will select the top 30. A formal definition of random forest is given in the modeling stage. **Figure 2** shows the variable reduction summary process.

Figure 2- Variable reduction process

Random forests can be used to rank the importance of variables in a regression or classification problem in a natural way. The following technique was described in Breiman's original paper and is implemented in the R package randomForest. The first step in measuring the variable importance in a data set. During the fitting process the out of-bag error for each data point is recorded and averaged over the forest (errors on an independent test set can be substituted if bagging is not used during training). To measure the importance of the j-th feature after training, the values of the j-th feature are permuted among the training data and the out-of-bag error is again computed on this perturbed data set. The importance score for the -th feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score is normalized by the standard deviation of these differences.

Table 1 below shows the 30 variables with the highest mean decrease in accuracy. This means that if we drop a variable, the accuracy of the random forest decreases by that amount. Therefore, the higher the value the most important the variable.

Variables	M.D.A	Variables	M.D.A	Variables	M.D.A
"num_var45_hace3"	38.09988	"saldo_var8"	37.26204	"saldo_medio_var5_hace2"	37.12389
"num_var22_hace3"	32.09259	"num_var45_ult1"	28.4995	"num_meses_var39_vig_ult3"	24.9312
"num_op_var41_hace2"	24.46637	"imp_op_var41_comer_ult1"	22.27106	"num_var43_recib_ult1"	22.18823
"var36"	21.85776	"imp_op_var41_efect_ult3"	21.61292	"ind_var43_recib_ult1"	20.62775
"saldo_var37"	20.32283	"num_var22_hace2"	18.96518	"var15"	18.89012
"num_var37_med_ult2"	17.82756	"num_meses_var5_ult3"	17.81556	"saldo_medio_var8_hace2"	17.79455
"num_var24_0"	17.4508	"saldo_medio_var12_hace2"	17.05778	"num_var42_0"	16.50506
"saldo_var5"	16.32811	"num_var5_0"	15.83212	"saldo_medio_var5_hace3"	15.50904
"num_op_var41_efect_u	15.32124	"saldo_medio_var13_corto_ha	14.72462	"imp_aport_var13_hace3"	14.68576
"var3"	13.25985	"saldo_var25"	12.46703	"saldo_medio_var13_corto_ha	11.44937

Table 1-Important variables given their mean decrease in accuracy

Once we have the cleaned data with the 30 variables that we are going to use for modeling, we can do some further exploratory data analysis on these variables.

3 Exploratory Data analysis

Given that the variables are semi-anonymized, it becomes difficult to obtain an interpretation of the variables. However, there is an exception: var15 has a minimum value of 5 and a maximum value of 105. Looking to **Figure 3** below, which represents the density plot of var15 by satisfaction level, we can essentially conclude that var15 represents the age of the customers.

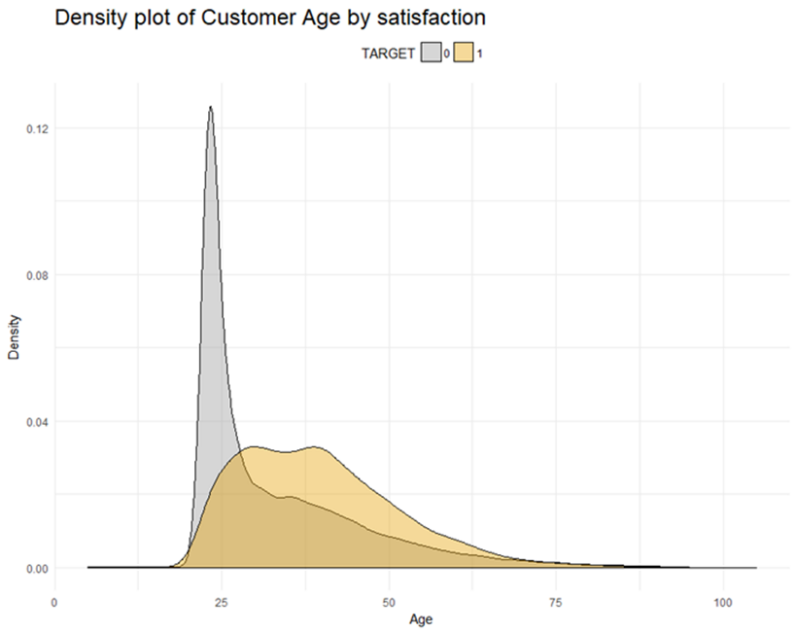

Figure 3 – Density plot of age by satisfaction level

Satisfied customers' density plot, in grey, quickly decrease when the age increases. On the other hand, dissatisfied customers' density plot doesn't decrease like the satisfied customers. This involves that the ratio of dissatisfied to satisfied customers is higher for older people. This therefore suggests that Santander should target older customers for their satisfaction campaigns. In fact, as we have seen in the data description part, Var15 (Age) is one of the most significant predictors for satisfaction. The older the client, the more likely is to be dissatisfied.

Since we have an idea of which are the most important predictors for customer satisfaction, we will analyze the distribution of the 8 most important variables given the ranking that we created using random forest in **table 1**. Since the variables are semi- anonymized, we can't understand the full meaning of the variables. However, we can still plot the density plots by satisfaction level as in **Figure 4**. The grey density plot represents the satisfied customers and the orange one the dissatisfied ones. Although it is challenging to extract any findings, it seems that the higher the value of most of the variables, the less likely a customer is to be unhappy. It seems that a lot of these variables are related to balance, which is the English translation to "saldo". It therefore makes sense that the higher the account balance, the happier customers are with the bank. This idea will be further explored in the logistic regression section when we study the table of the influence of each variable on the satisfaction level.

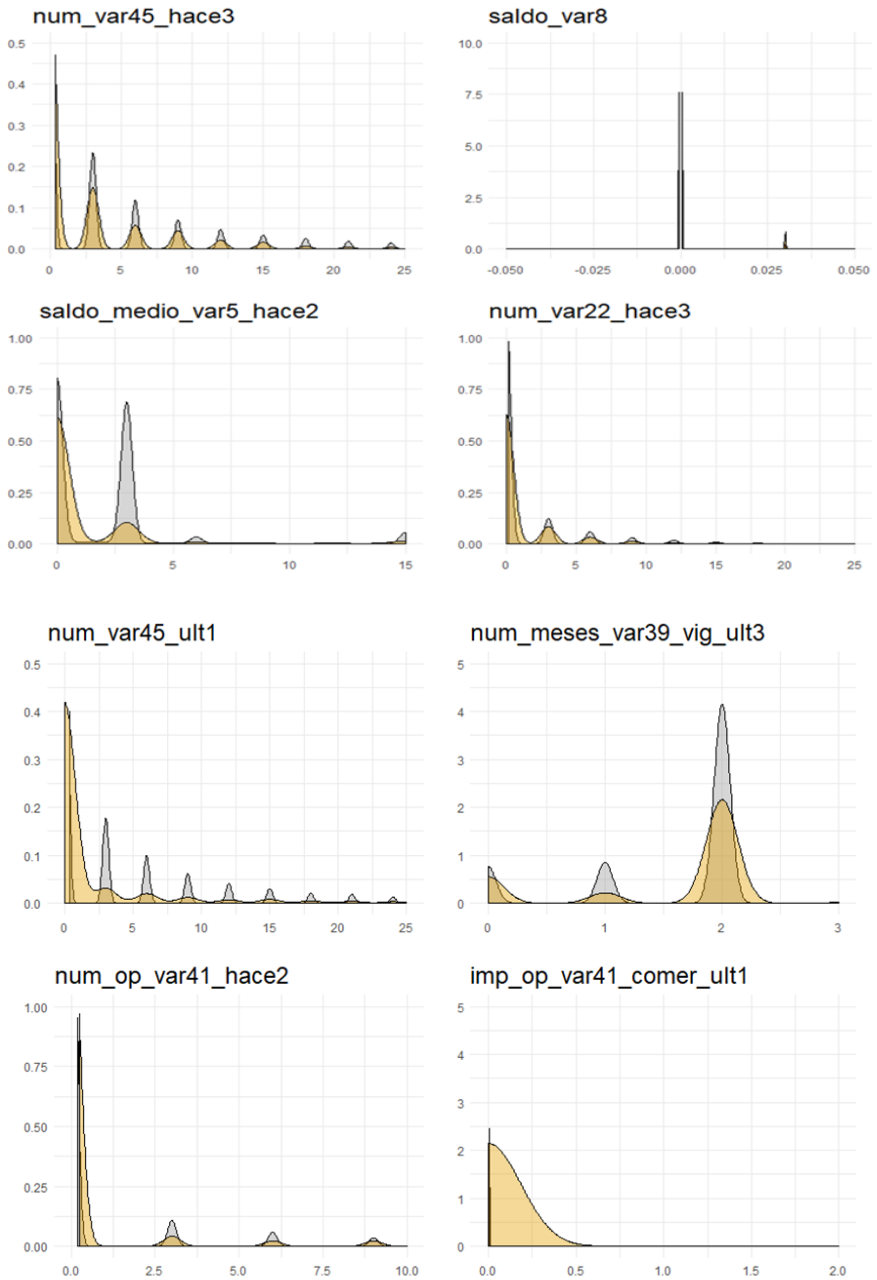

Figure 4 – Density plots by satisfaction level

4 Modeling

4.1 Performance Metrics

To evaluate the performance of the models on a testing sample, we first need to introduce the concept of a confusion matrix. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. It shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

If we consider a binary classifier we have four possible outcomes in the confusion matrix as shown in **Figure 5**. The four possible outcomes are the following:

- True negative: We predict 0 and the class is actually 0.
- False negative: We predict 0 and the class is actually 1.
- True positive: We predict 1 and the class is actually 1.
- False positive: We predict 1 and the class is actually 1.

		Predicted Class	
		Class 0	Class1
True class	Class 0	True Negative (TN)	False Postive (FP)
	Class 1	False Negative (FN)	True Positive (TP)

Figure 5 – Confusion matrix

Now that the confusion matrix has been defined, we can elaborate on the out-of-sample performance metrics that we are going to use to compare the models.

The ROC curve is a fundamental tool for diagnostic test evaluation. In a ROC curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points of a parameter. True Positive Rate: (TPR)= TP / (TP + FN) and False Positive Rate (FPR)= (FP/ FP + TN). Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two diagnostic groups (diseased/normal). We will use Area under the curve (AUC) on the testing sample as the main metric to compare model performance.

False Negative Rate**:** When it's actually 1, how often does it predict 0? FNR= FN / (FN + TP). False Negative Rate is very important in this case since we are very interested in the False negatives, given that is particularly harmful for a bank to predict that a customer is satisfied when they really are unhappy. This could result in losing a customer from the bank without further notice. This type of error is more severe than the False Positive (FP), since predicting that customers are unhappy when they really are satisfied would never lead in customers leaving the bank.

Other secondary metrics that have been considered to evaluate the performance are:

Misclassification Rate: Overall, how often is it wrong? MR = (FP + FP) / (FP + TP + FN + TN)

False positive Rate: When it's actually 0, how often does it predict 1? FPR = FP / (FP + TN)

Precision: When it predicts 1, how often is it correct? Precision = TP / FP + TP

Recall: What percent of the 1's did you catch? Recall = TP / (TP + FP)

4.2 Statistical Models

To identify dissatisfied clients, we use four different methods: logistics regression, classification trees, random forests and extreme gradient boosting (XGBoost from (Chen and Guestrin, 2016). We train these models using the training set and compare their performance using the testing sample. The dataset has been randomly split on training (80%) and testing (20%).

4.2.1 Logistic Regression

Logistic regression is a relatively 'simple' machine learning algorithm used to describe data and to explain the relationship between one dependent binary variable, TARGET (0 or 1), and some independent variables. It is used to estimate the probability of the binary response based on the predictor variables. While it doesn't usually get great performance metrics, it is pursued here as a baseline to compare to more sophisticated models since it's easy to interpret.

To get an initial idea of how logistic regression works, we are showing the summary of fitting a logistic regression model with all 30 candidate variables. This is called the full model and its summary is shown in **Table 5,** where the variables have been ordered by significance level**.**

Coefficients:	Estimate	Std. Error	z value	Pr(>|z|)		%
(Intercept)	-4.04E+00	1.71E-01	-23.666	2.00E-17	***	0.017597
var15	3.81E-02	1.40E-03	27.146	2.00E-16	***	1.038845
num_meses_var5_ult3	-4.87E-01	2.50E-02	-19.495	2.00E-16	***	0.614344
num_var22_hace2	3.02E-02	5.74E-03	5.265	1.40E-07	***	1.030681
num_var24_0	-4.26E-01	9.05E-02	-4.705	2.54E-06	***	0.653182
num_op_var41_efect_ult1	3.00E-02	6.41E-03	4.682	2.84E-06	***	1.030465
var3	-2.85E-02	6.77E-03	-4.214	2.51E-05	***	0.971883
saldo_medio_var8_hace2	-4.81E-04	1.17E-04	-4.125	3.71E-05	***	0.99952
num_var22_hace3	2.98E-02	7.68E-03	3.882	0.000104	***	1.030269
saldo_medio_var5_hace2	-3.01E-05	8.32E-06	-3.614	0.000302	***	0.99997
var36	2.44E-03	7.74E-04	3.147	0.001649	**	1.00244
imp_op_var41_comer_ult1	2.09E-04	6.75E-05	3.091	0.001993	**	1.000209
num_var45_ult1	4.54E-03	1.49E-03	3.048	0.002302	**	1.00455
num_var43_recib_ult1	-4.16E-02	1.39E-02	-2.995	0.002744	**	0.959301
imp_aport_var13_hace3	-1.85E-05	6.50E-06	-2.84	0.004517	**	0.999982
saldo_medio_var12_hace2	-8.27E-06	3.00E-06	-2.754	0.005891	**	0.999992
saldo_medio_var13_corto_hace2	-7.64E-06	2.92E-06	-2.615	0.00892	**	0.999992
saldo_var5	-2.48E-05	1.02E-05	-2.438	0.01475	*	0.999975
imp_op_var41_efect_ult3	3.55E-05	1.48E-05	2.409	0.016009	*	1.000036
saldo_var37	5.24E-05	2.48E-05	2.11	0.034885	*	1.000052
num_var42_0	7.14E-02	4.52E-02	1.579	0.114339		1.073989
saldo_medio_var13_corto_hace3	-1.14E-04	7.54E-05	-1.512	0.130523		0.999886
num_var45_hace3	3.59E-03	2.39E-03	1.503	0.132939		1.003596
saldo_var25	-6.17E-05	4.33E-05	-1.427	0.153635		0.999938
saldo_var8	-3.72E-05	3.09E-05	-1.204	0.228414		0.999963
saldo_medio_var5_hace3	-1.38E-05	1.47E-05	-0.94	0.347439		0.999986
num_meses_var39_vig_ult3	2.65E-02	2.83E-02	0.936	0.349362		1.026844
num_var5_0	-2.92E-02	3.50E-02	-0.834	0.404485		0.971261
ind_var43_recib_ult1	-5.87E-02	1.11E-01	-0.528	0.597277		0.943037
num_var37_med_ult2	-7.85E-03	1.53E-02	-0.514	0.606971		0.992184
num_op_var41_hace2	1.50E-03	3.24E-03	0.464	0.642663		1.001504

Table 5 – Summary of the full model for logistic regression.

Table 5 shows the effect of each predictor to the satisfaction of the customers. For instance, the coefficient for var15 says that, holding all the other variables constant, we will see a 3.9% increase in the odds of being dissatisfied for each one-unit increase in var15, given that exp(0.0381) = 1.0388 which rounds to an increase of approximately 3.9%. 0.0381 represents the coefficient of var15 in the full model. Given that var15 is the age of the customers, a one-year increase in the age of a client increases its chances of being dissatisfied by 3.9%.

Continuing with the findings of the Exploratory data analysis section, All the "saldo" variables have negative coefficients. As discussed earlier, "Saldo" in Spanish stands for balance. This involves that the higher the balance of an account, the lower the chance a customer is dissatisfied. This matches with what we saw in the Exploratory data analysis of Figure 4.

Although we can get some insights from this model, it is critical to analyze its performance before validating these insights. First, not all 30 variables in the model are significant considering a significance level of 5% and therefore there is a risk of overfitting. When selecting the threshold (p-value) that minimizes the overall misclassification rate, we get the performance metrics shown in **table 3**. This is a big problem when dealing with imbalanced data, since by predicting that all customers are satisfied, we still get an extremely high accuracy. As it can be seen in **table 3**, the performance of the full logistic regression model is extremely good in terms of misclassification rate, given that only 4.08% of the observations are misclassified. However, this p-cut gets a False Negative Rate equal to 1, meaning that the model is not predicting any dissatisfied customers. This is a big problem in the machine learning field, and therefore **my second motivation of this paper is to explore and compare ways of dealing with imbalanced data.**

		ROC	MR	FNR	FPR	PRECISION	RECALL	FMEASURE
Logistic Regression	Full model	0.7835	0.0408	1	0.00005	0.9592	0.9999	0.9791

Table 3 – Performance metrics for full logistic regression model.

4.2.2 Imbalanced Data

Many practical classification problems are imbalanced. For instance, as we have seen in this dataset, 96.05% of customers are satisfied, while only 3.95% of them are dissatisfied. This implies that it is particularly important for us to try to predict the dissatisfied ones or the "rare class", since they are the ones that we need to address to convince them to stay in the bank. However, the most commonly used classification algorithms do not work well for such problems because they aim to minimize the overall error rate, rather than paying special attention to the positive class (dissatisfied customers in this case). For instance, by predicting that all clients are satisfied, the accuracy obtained is around 96%. However, because all clients have been predicted as satisfied, the model fails to identify unhappy customers. Therefore, we need to apply some of the following techniques to account for this unbalance.

There are two common approaches to tackle the problem of extremely imbalanced data. One is based on cost sensitive learning: assigning a high cost to misclassification of the minority class and trying to minimize the overall cost. The other approach is to use a sampling technique: Either down-sampling the majority class or over-sampling the minority class. Therefore**, our second goal for this project is to study and compare these approaches to see which one tackles better the problem of imbalanced data.**

Cost sensitive learning: This method evaluates the cost associated with misclassifying observations. it highlights the imbalanced learning problem by using cost matrices which describes the cost for misclassification in a particular scenario. In this problem we logically care more about the False Negatives than False positives. In other words, the cost associated predicting a dissatisfied customer as satisfied is higher than predicting a satisfied customer as dissatisfied. Therefore, we will assign a cost ratio of 20:1 for FN vs FP and try to minimize the total cost instead of the total misclassification rate. We chose this cost ratio since the ratio of satisfied customers to dissatisfied ones is approximately 20 to 1.

The next approaches create balanced data distributions. This involves that the initial dataset from which the machine learning algorithm is taking the data is already balanced (i.e. same number of satisfied customer than dissatisfied).

Undersampling: This method works with majority class. It reduces the number of observations from majority class to make the data set balanced. So, it randomly chooses observations from majority class which are eliminated until the data set gets balanced. This method is usually best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles.

Oversampling: This method works with minority class. It replicates the observations from minority class to balance the data. It is also known as *upsampling.* Random oversampling balances the data by randomly oversampling the minority class. An advantage of using this method is that it leads to no information loss. The disadvantage of using this method is that, since oversampling simply adds replicated observations in original data set, it ends up adding multiple observations of several types, thus leading to overfitting.

Given that the resampling approaches (undersampling and oversampling) randomly select observations from one of the classes, the performance obtained each time we run an iteration varies. Therefore, we will run 10 iterations for each approach and take average of the performance metrics. This will lead to a more accurate comparison of the different approaches to deal with imbalanced data. For each of the predictive models (logistic regression, classification tree, random forest and XGBoost) we will try every approach to tackle imbalanced data. By doing this, we will get a good idea of which approach leads to the highest performance.

4.2.3 Classification tree

Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. The tree can be explained by two entities, nodes and leaves. The leaves are the decisions or the final outcomes, and the decision nodes are where the data is split. An example of a classification tree can be seen in **Figure 6,** which is the tree that gets the best predictive performance using a cost sensitive learning approach.

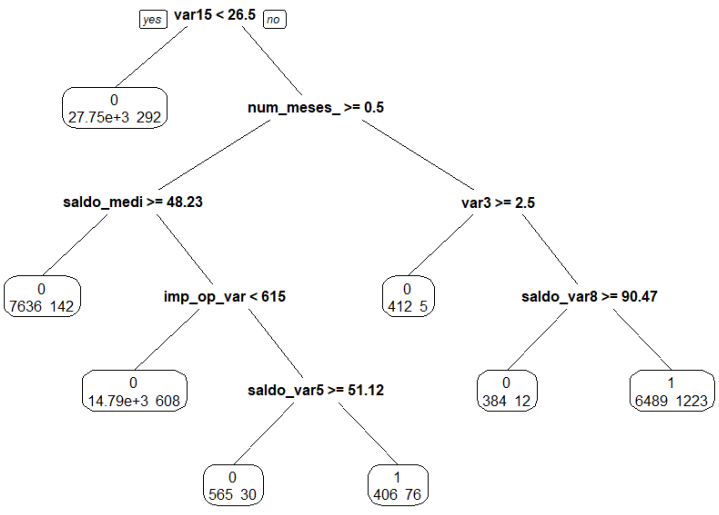

Figure 6 – Classification Tree

The concept is simple. When a prediction is made, go from the top of the tree to the bottom. Go left at the top if var15 is lower than 26.5, right otherwise. Repeat this process for all nodes until a leaf is reached and the prediction made corresponds to the class label of that leaf.

4.2.4 Random Forest

Random forest (Breiman, 2001) is an ensemble of unpruned classification or regression trees, induced from bootstrap samples of the training data, using random feature selection in the tree induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble. Random forest generally exhibits a substantial performance improvement over the single tree classifier such as CART and C4.5. It yields generalization error rate that compares favorably to Adaboost, yet is more robust to noise. However, similar to most classifiers, RF can also suffer from the curse of learning from an extremely imbalanced training data set. As it is constructed to minimize the overall error rate, it will tend to focus more on the prediction accuracy of the majority class, which often results in poor accuracy for the minority class. To alleviate the problem, we propose the two solutions stated earlier: balanced random forest (BRF) using undersampling and oversampling, and a weighted random forest (WRF) using the cost sensitive learning approach.

4.2.5 Extreme Gradient Boosting (XGBoost)

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. Gradient Boosting is a sequential technique which works on the principle of ensemble. It combines a set of weak learners and delivers improved prediction accuracy. At any instant t, the model outcomes are weighed based on the outcomes of previous instant t-1. The outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher. The name XGBoost, though, actually refers to the engineering goal to push the limit of computations resources for boosted tree algorithms. Which is the reason why many people use XGBoost. For model, it might be more suitable to be called as regularized gradient boosting. Unfortunately, XGBoost is still sensitive to imbalanced data

The XGBoost Advantage: Some of the improvements from gradient boosting are:

1. Regularization: Standard GBM implementation has no regularization like XGBoost, therefore it also helps to reduce overfitting. In fact, XGBoost is also known as ‘regularized boosting ‘technique.
2. Parallel Processing: XGBoost implements parallel processing and is blazingly faster as compared to GBM.
3. High Flexibility: XGBoost allow users to define custom optimization objectives and evaluation criteria.

5 Results

5.1 Overall results

Table 4 below summarizes the results of the 4 statistical models as well as each of the approaches used to deal with imbalanced data.

Using the area under the ROC curve as the main performance metric, we can conclude that XGBoost leads to the highest predictive performance, followed by random forest. Finally, although logistic regression and classification tree have less predictive power than XGBoost and random forest, they are simpler models to interpret.

		ROC	MR	FNR	FPR	PRECISION	RECALL
Logistic Regression	Imbalanced	0.7775	0.0408	1	0.00005	0.9592	0.9999
	Weighted 20:1	0.775	0.196	0.3725	0.1887	0.9808	0.811
	OverSampling	0.7818	0.3752	0.2145	0.3821	0.9854	0.6178
	UnderSampling	0.7751	0.381	0.2177	0.3879	0.9851	0.612
		ROC	MR	FNR	FPR	PRECISION	RECALL
Classification tree	Imbalanced	0.5	0.04077	1	0	0.9592	1
	Weighted (20:1)	0.763	0.1437	0.4673	0.1297	0.9774	0.8702
	OverSampling	0.7819	0.2726	0.2532	0.2735	0.9853	0.7264
	UnderSampling	0.783	0.2686	0.2629	0.2688	0.9849	0.7311
		AUC	MR	FN	FPR	PRECISION	RECALL
RF	Imbalanced	0.55	0.0408	1	0	0.9592	1
	Weighted (20:1)	0.808	0.1423	0.4745	0.1286	0.9778	0.8713
	Balanced (over)	0.8167	0.1539	0.4548	0.1411	0.9779	0.8588
	Balanced (under)	0.8229	0.236	0.2306	0.2372	0.9873	0.7627
		ROC	MR	FNR	FPR	PRECISION	RECALL
XGBoost	Imbalanced	0.8382	0.0422	0.9887	0.0019	0.9595	0.998
	Weighted (20:1)	0.8388	0.1719	0.3274	0.1653	0.9835	0.8346
	OverSampling	0.829	0.2226	0.2629	0.2209	0.9858	0.779
	UnderSampling	0.834	0.293	0.1838	0.2977	0.9889	0.7022

Table 4 – Performance results

Regarding the different approaches to deal with imbalanced data on this dataset, we can conclude that both Weighted and Balanced (undersampling and oversampling) have performance superior to the imbalanced approach, which fails to predict dissatisfied customers. Between weighted, undersampling and oversampling approaches, however, there is no clear winner. Depending on the statistical model used, certain approach performs slightly better than others, but the truth is that this difference is so minimal that all three approaches are equally valid. In general terms, undersampling is recommended for huge datasets, oversampling for small ones and the weighted approach for when the cost ratio between False Negatives and False Positives is accurately known. The next sections will dig deeper on each of the parameter tuning process used for each statistical model.

5.2 Logistic Regression Results

As previously shown in section 4.3.1, the full logistic regression model using the traditional approach leads to predicting that all customers are satisfied due to the imbalanced data.

For variable selection in logistic regression, we used LASSO. However, LASSO doesn't work well for imbalanced data and therefore we decided to use stepwise variable selection method for the weighted approach. In each step of stepwise variable selection method, a variable is considered for addition to or subtraction from the set of candidate variables based on some prespecified criterion, minimizing the AIC in this case. The final model contained 23 variables and used grid search method to find the cutoff value that leads to the lowest misclassification cost given the weights of 20 for False negatives and 1 for False Positives. The cost associated with each cutoff value can be seen in **Figure 7**. The optimal cutoff value for the weighted approach was found to be 0.06, meaning that all customers with a predicted probability higher than 0.06 are classified as dissatisfied customers.

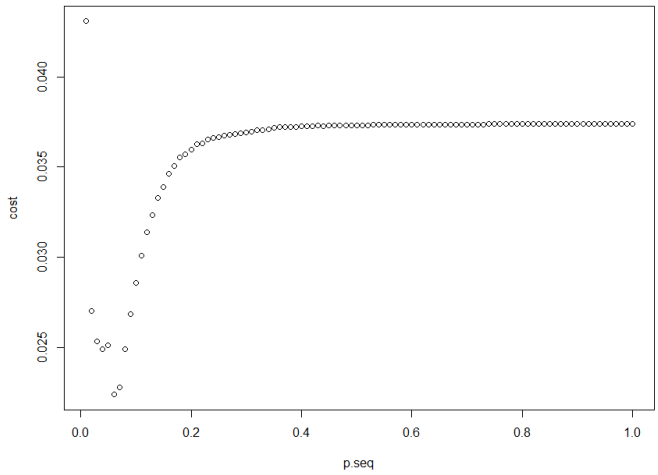

Figure 7 – P-cut selection for the weighted approach

For the balanced approaches, we decided to use LASSO variable selection. LASSO - Least Absolute Shrinkage and Selection Operator - was first formulated by Robert Tibshirani in 1996. It is a powerful method that perform two main tasks: regularization and feature selection. The LASSO method puts a constraint on the sum of the absolute values of the model parameters, the sum has to be less than a fixed value (upper bound). In order to do so the method applies a shrinking (regularization) process where it penalizes the coefficients of the regression variables shrinking some of them to zero. During features selection process the variables that still have a non-zero coefficient after the shrinking process are selected to be part of the model. The goal of this process is to minimize the prediction error. **Figure 8** and **9** show the variables coefficients and the misclassification error associated with each lambda value respectively when the undersampling approach is carried out.

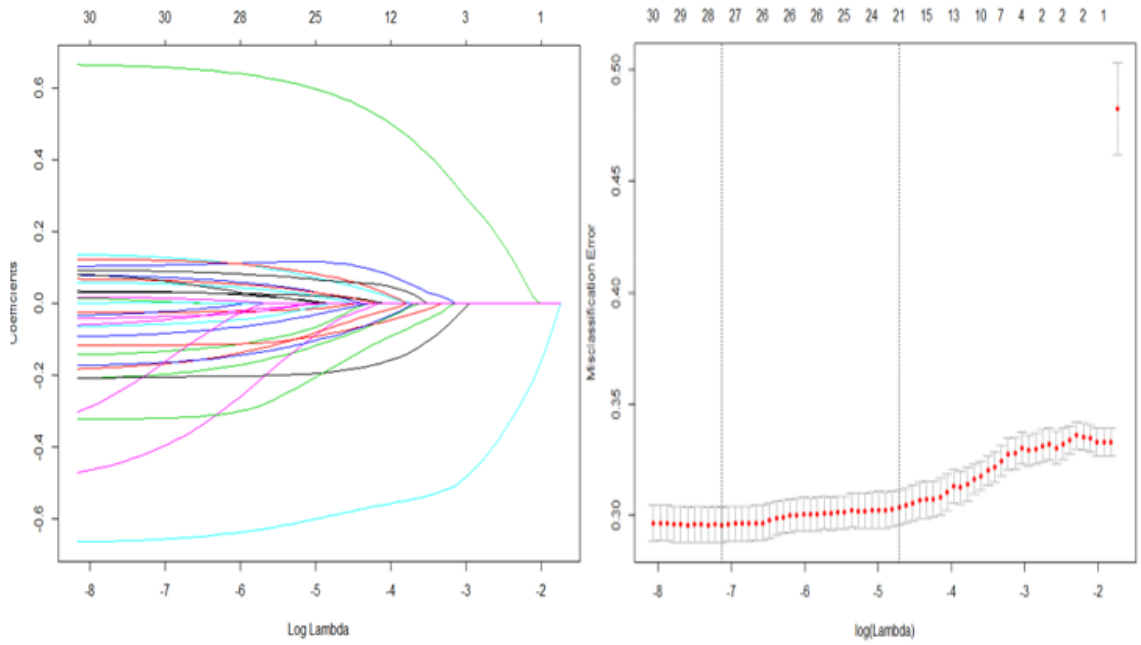

Figure 8 and **9-**Lambda parameter tunning

On this case, we decided to select the lambda value of the right bar of of Figure 9 given that its misclassification error is within 1 standard error of the smallest error. We picked this lambda instead of the one that gives the lowest error in order to get a simpler model (21 variables instead of 28) and gain some interpretability.

The ROC curves for the approaches to deal with imbalanced data using logistic regression can be found in **Figure 10.** As it is shown, very similar are under the curve are provided. It is important to realize that the undersampling/oversampling Area under the curve values don't necessarily match with the ones in table 4, since those are the averages of 10 repetitions.

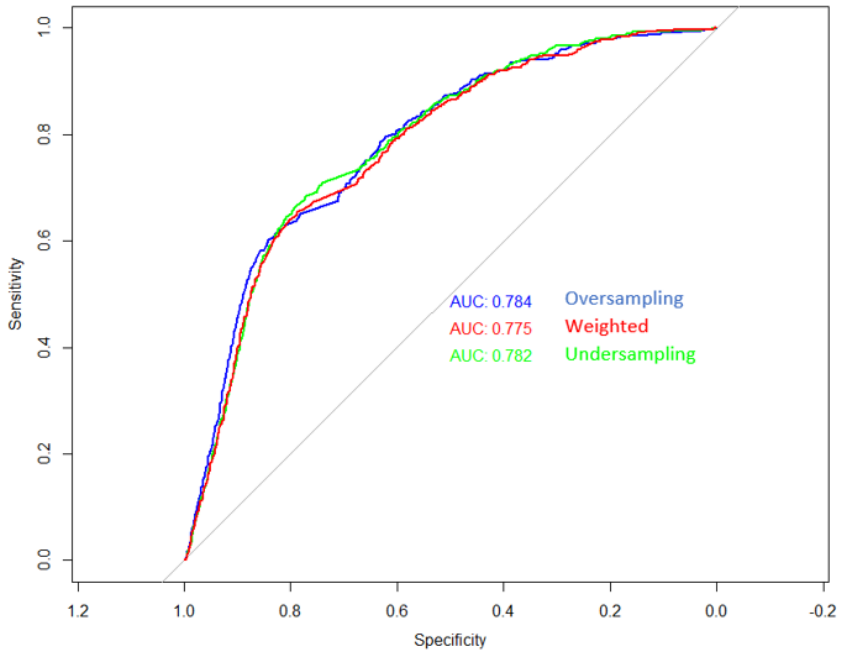

Figure 10- ROC curves for logistic regression approaches

5.3 Classification Tree Results

Regarding classification trees, Undersampling, oversampling and weighted approaches get similar results in terms of Area Under the Curve, and clearly outperform the imbalanced approach. The weighted approach gets a higher False Negative Rate, but also gets a lower False Positive Rate than the undersampling and oversampling approaches. It is therefore up to the users to understand which metric is more important to them before selecting the model. The performance of the trees has been improved by pruning. This involves removing the branches low-importance features. By doing this, we reduce the complexity of the tree, and thus increase its predictive power by reducing overfitting. **Figure 11** below shows ROC curves for the three different approaches.

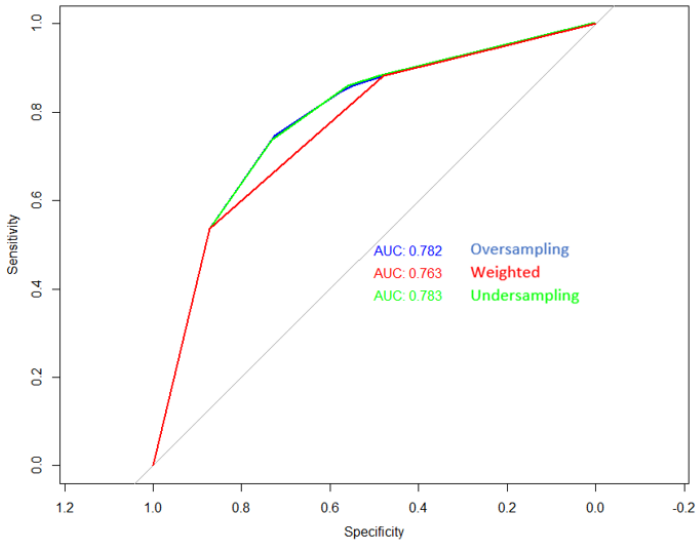

Figure 11- ROC curves for Classification tree approaches

5.4 Random Forest Results

Once again, the approaches to tackle imbalanced data outperforms the imbalanced approach, which is unable to predict dissatisfied customers. Moreover, there is not a considerable difference between the results of these approaches. For this case, prediction has been made by aggregating predictions of 100 trees and selecting the majority vote. By doing this, we have been able to improve performance over a single tree. **Figure 12** shows the ROC curves for the different approaches.

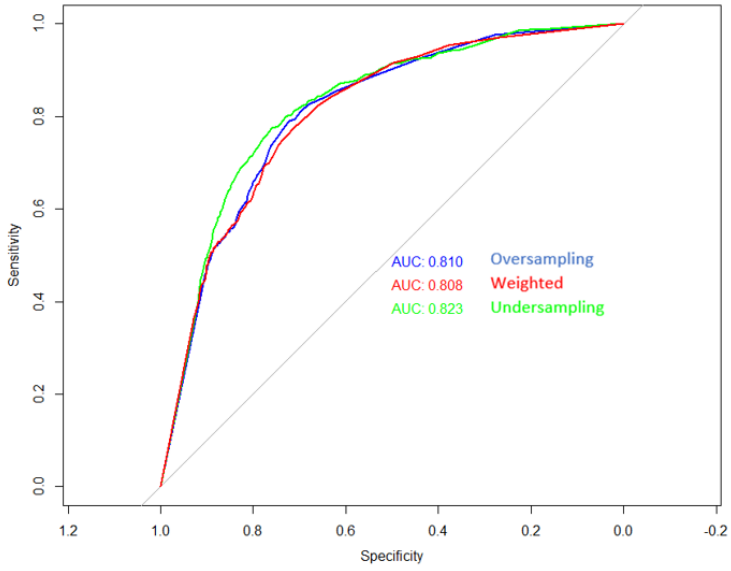

Figure 12- ROC curves for Random Forest approaches

As we did for variable selection, we can use random forest to compute the importance that each variable has on the satisfaction level of the customers. This is shown in **Figure 13**, which ranks the 20 most important variables to predict satisfaction. The mean decrease in Gini means that if we drop a variable, the accuracy of the random forest would decrease by that amount. Therefore, the higher the value the more important the variable is. Var15 or "Age" is the most influential variable, which matches with the results obtained in the logistic model in **table 5**.

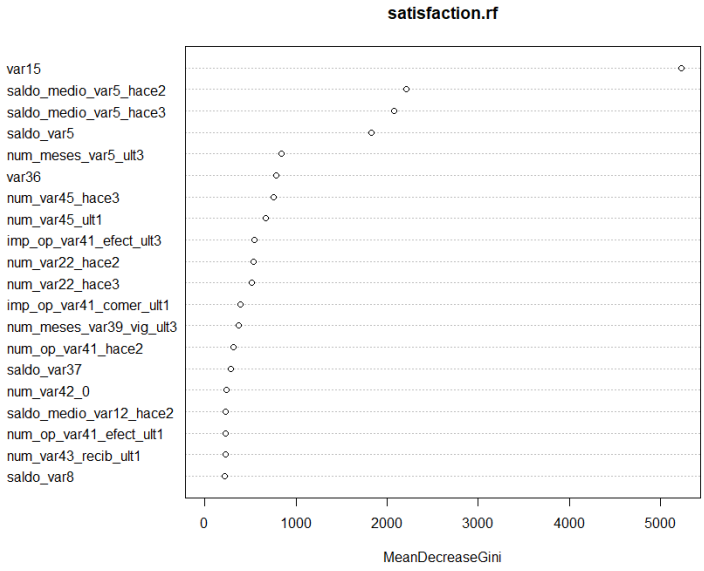

Figure 13- Feature importance using random forest

5.5 XGBoost results

Finally, XGBoost has been found to be the best model in terms of predictive power. Although there is not a considerable difference in the performance of the oversampling, undersampling and weighted approaches (**Figure 14**), they still outperform the imbalanced approach.

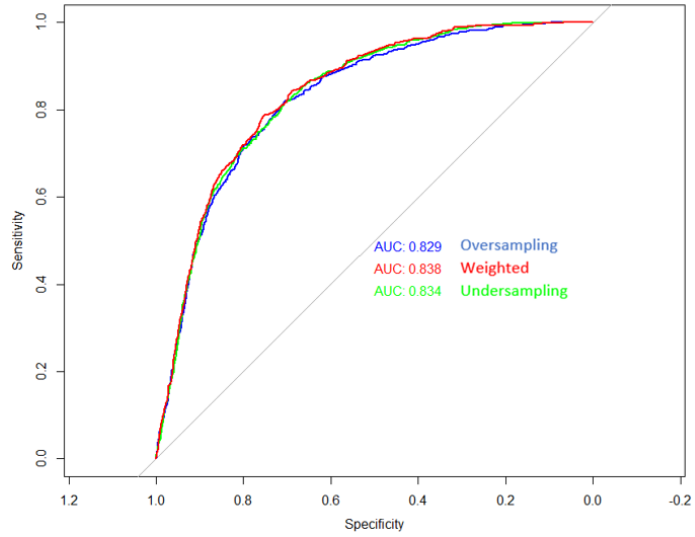

Figure 14- ROC curves for XGBoost

Improving performance by parameter tuning in XGBoost is somewhat complex since the model has multiple parameters. By alternatively selecting different combinations of parameters we can improve the performance of the XGBoost model. For instance, the XBoost model that resulted in the highest Area Under the Curve had the parameters shown in **table 5**.

Parameters	eta	max_depth	subsample	colsample_bytree	nrounds
XGBoost	0.02	8	0.9	0.085	350

Table 5 – Parameters selected for weighted XGBoost

6 Conclusion and Recommendations

This paper investigates several machine learning algorithms to predict whether customers from Santander Bank are satisfied or dissatisfied. Through the different statistical models implemented using R, XGBoost provides the highest predictive accuracy, followed by random forest. Although Logistic regression and decision trees models result in lower predictive power than XGBoost and RF, they are easier to interpret and can be useful for scenarios where we care more about interpretability than prediction. Therefore, the paper provides Santander bank with different models to predict dissatisfied customers.

Given that the variables are semi-anonymized, it is difficult to provide specific recommendations that would allow the bank to increase customer's satisfaction. For instance, the bank can target older customers on their satisfaction campaigns because they are more likely to be dissatisfied as it was shown in the density plot of Age by satisfaction level. Since Santander bank are the only ones that know the meaning of every variable, they should focus on the variables that have the biggest impact on customer satisfaction, as shown in Figure 13, and try to find ways of modifying the client's values on these variables so the number of dissatisfied customers decreases

Finally, we also presented two ways of learning imbalance: Weighted and balanced approaches. Weighted approaches put more weights on the minority class, thus penalizing more heavily on misclassifying the minority class. Balanced approaches are divided in undersampling and oversampling. They both artificially alter the class distribution so that classes are represented equally. From the experiment on the customer satisfaction data set, we can conclude that both Weighted and Balanced approaches have performance superior to the original imbalanced algorithms. Between Weighted and Balanced approaches, however, there is no clear winner.

References

Chao Chen, Andy Liaw, Leo Breiman (2004). Using Random Forest to Learn Imbalanced Data. Department of Statistics, UC Berkeley.

Tianqi Chen, Tong He, Michaël Benesty (2014). Xgboost presentation. *https://cran.r-project.org/web/packages/xgboost/vignettes/xgboostPresentation.html*

ROC curve analysis. *https://www.medcalc.org/manual/roc-curves.php*

Valeria Fonti (2017). Feature Selection using LASSO. Vrije Universiteit Amsterdam

Hoang Duong (2016). Decision Trees, Random Forest, Gradient Boosting. *http://hduongtrong.github.io/2016/04/04/Trees-Based-Models/*

Analyhtics Vidhya content team (2016). Practical Guide to deal with Imbalanced Classification Problems in R. https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

Santander Bank (2016). https://www.kaggle.com/c/santander-customer-satisfaction

Jason Brownlee (2016). What is a confusion matrix in machine learning? https://machinelearningmastery.com/confusion-matrix-machine-learning/

University of Cincinnati logo Picture obtained from: http://www.fbschedules.com