SANTANDER CUSTOMER PREDICTION REPORT

Table of Contents

Background

At Santander , mission is to help people and businesses prosper. We are always looking for ways to help our customers understand their financial health and identify which products and services might help them achieve their monetary goals. Our data science team is continually challenging our machine learning algorithms, working with the global data science community to make sure we can more accurately identify new ways to solve our most common challenge, binary classification problems such as: is a customer satisfied? Will a customer buy this product? Can a customer pay this loan?

Problem Statement

In this challenge, we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

Data Details

Provided with an anonymized dataset containing 200 numeric feature variables, the binary target column, and a string ID_code column.

Problem Analysis

This is a binary classification problem under supervised machine learning algorithm. The task is to predict the value of target column in the test set.

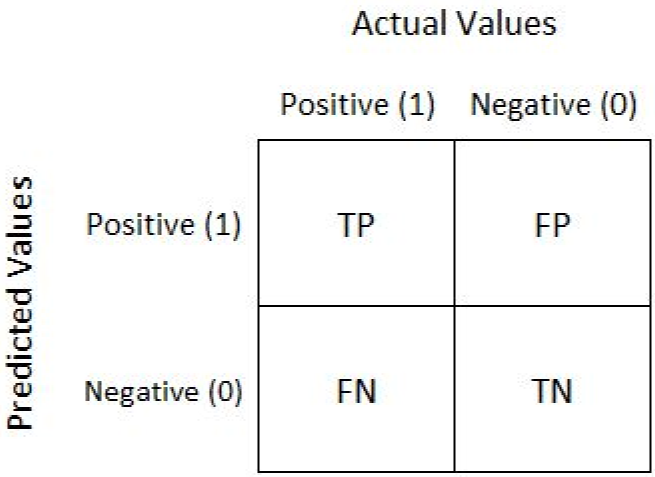
Evaluation Metrics

This is a classification problem and we need to understand confusion matrix for getting evaluation metrics. It is a performance measurement for machine learning classification problem where output can be two or more classes.

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. This is the key to the confusion matrix. The confusion matrix 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.

It is a table with 4 different combinations of predicted and actual values.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted to be negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted to be positive.

Based on confusion matrix we have following evaluation metrics

Accuracy
Out of all the classes, how much we predicted correctly. It should be high as possible.

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall
Out of all the positive classes, how much we predicted correctly. It should be high as possible.

$$\textbf{\textit{Recall}} = \frac{TP}{TP + FN}$$

Precision
Out of all the positive classes we have predicted correctly, how many are actually positive.

$$\textit{Precision} = \frac{TP}{TP + FP}$$

F-measure
It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more

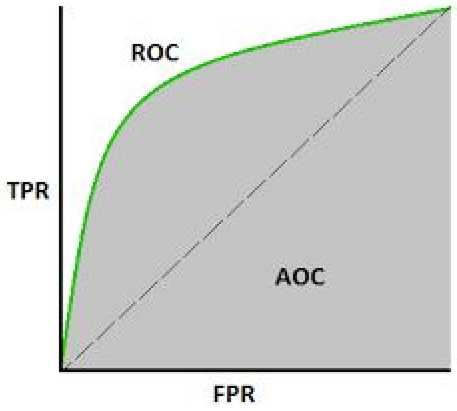
$$\textit{F - measure} = \frac{2*Recall*Precision}{Recall + Precision}$$

High recall, low precision
This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.

Low recall, high precision
This shows that we miss a lot of positive examples(high FN) but those we predicted as positive are indeed positive(low FP).

AUC-ROC curve
It is one of the most important evaluation metrics for checking any classification model's performance. It is also written as AUROC (Area Under the Receiver Operating Characteristics).

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s. The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.



$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{FPR} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP}$$

An excellent model has AUC near to 1 which means it has good measure of separability. A poor model has AUC near to 0 which means it has worst measure of separability. In fact it means it is reciprocating the result. It is predicting 0s as 1s and 1s as 0s. And when AUC is 0.5, it means model has no class separation capacity.

Exploratory Data Analysis

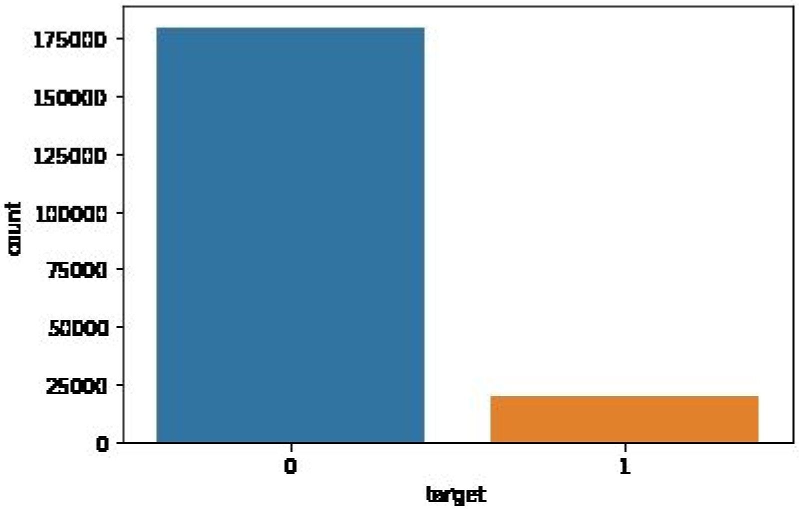
Check for variable data types in train and test data.
Id_code is object type, target is int type and 200 anonymous variables of float type.

Check for missing values
No missing values.

Shape of data
200000 observations with 202 columns in train data and 200000 observations with 201 columns in test data.

Check Balance of target column

0	179902
1	20098

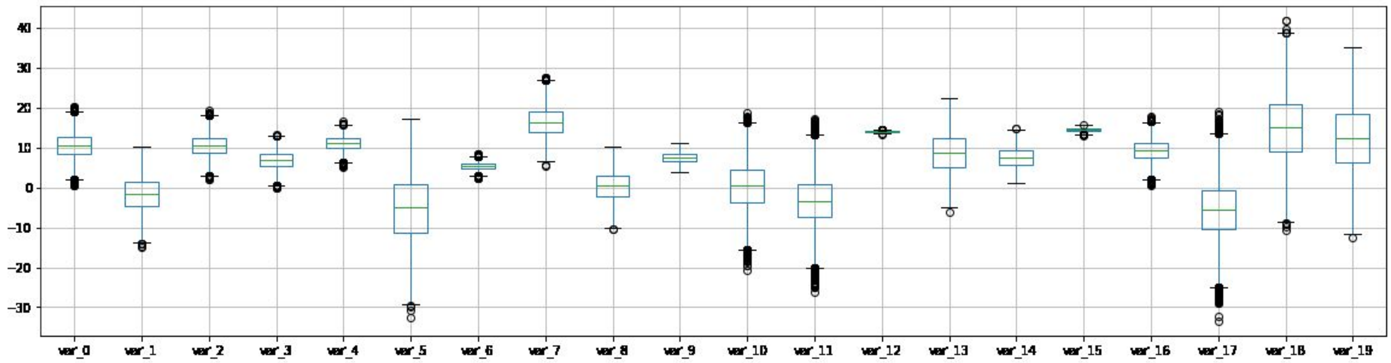


The dataset has 89.95 % of target 0 and 10.05 % of target 1

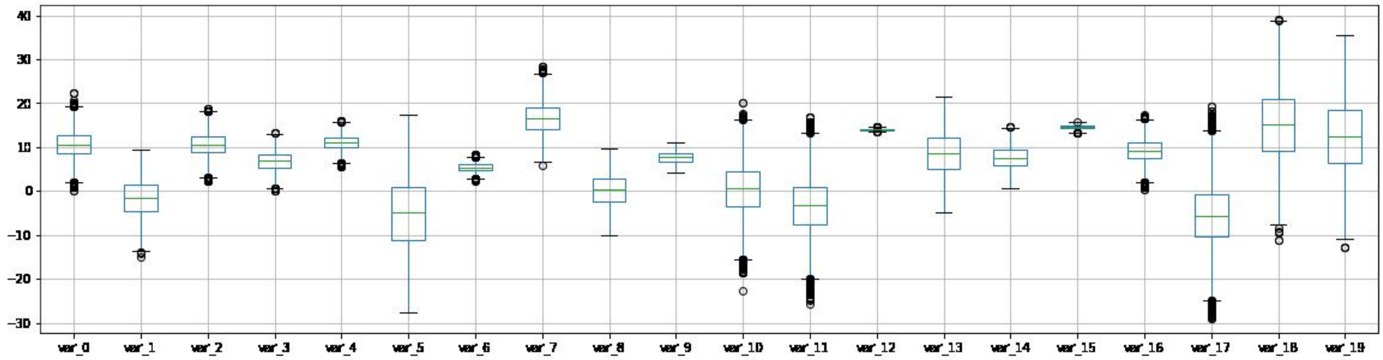
It has been observed that the given dataset is an imbalanced dataset.

Outlier Analysis

Boxplots have been used to plot the outliers in both the training and testing datasets. A boxplot for first 19 variables of train_data irrespective of target value is pasted below.



Likewise, outliers for first 19 variables of test_data.csv, irrespective of target values is pasted below.



The outliers in both the datasets have been removed by dropping the rows containing the outliers. The shape of the dataset after removing the dataset has been reduced to (173443, 201) for test_data and (175070, 202) for train_data.

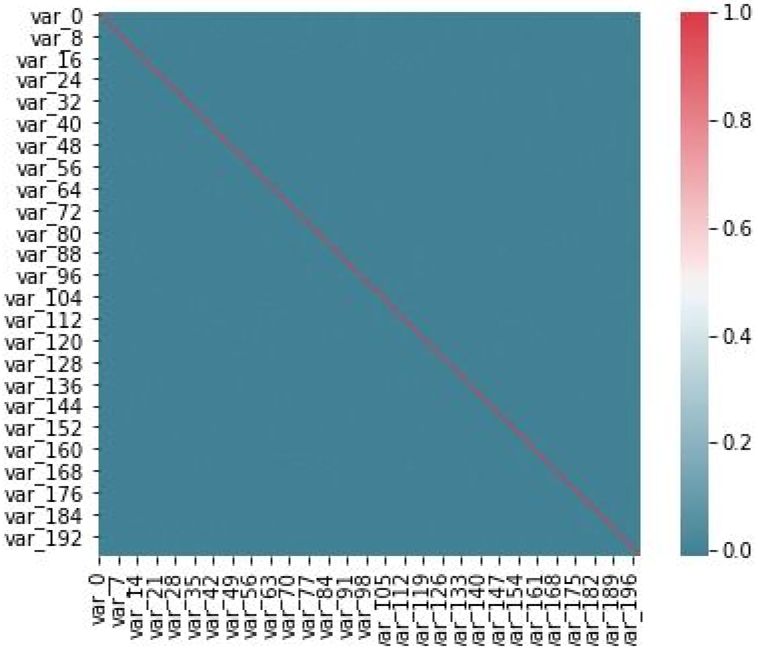
Distribution of data

Almost all features follow normalised distribution

Correlation among the variables

Minimum correlation among variables is -0.010291520700875701
Maximum correlation between variables is 0.009833584703433004

We have 200 features that are mostly uncorrelated between them



Duplicates in the datasets

No duplicates in both datasets.

Dealing Imbalanced dataset

Before modelling for this dataset let us understand how to deal with imbalanced dataset for classification problem. Traditional Machine Learning algorithms tend to produce unsatisfactory classifiers when faced with imbalanced datasets. For any imbalanced data set, if the event to be predicted belongs to the minority class and the event rate is less than 10%, it is usually referred to as a rare event. The conventional model evaluation methods do not accurately measure model performance when faced with imbalanced datasets. Standard classifier algorithms like Decision Tree and Logistic Regression have a bias towards classes which have large number of instances. They tend to only predict the majority class data. The features of the minority class are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.

However, while working in an imbalanced domain accuracy is not an appropriate measure to evaluate model performance. Hence we need evaluation metrics such as Recall, Precision, F1_score (harmonic mean of Precision and recall), AUC-ROC score along with Accuracy.

How to deal with these imbalanced datasets?

- Using Resampling techniques such as Random under Sampling, Random Over Sampling, Informed over Sampling (synthetic Minority Over Sampling Technique) are useful.

- Changing machine learning algorithm.

If we use sampling then we will be increasing observations (already we have 200000 observations) and hence speed will be lower. Also useful techniques such as SMOTE are susceptible to outliers and not fit for high dimensional data (we have 410 features). Other sampling techniques can cause overfitting also. Hence I am not using sampling techniques.

Modelling

Logistic regression

This is the classification problem. We can use logistic regression for this problem. Logistic Regression is used when the dependent variable(target) is categorical. It has a bias towards classes which have large number of instances. It tends to only predict the majority class data. The features of the minority class are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.

Random Forest

Random Forest is a bagging based ensemble learning model. Random forests is slow in generating predictions because it has multiple decision trees. Whenever it makes a prediction, all the trees in the forest have to make a prediction for the same given input and then perform voting on it. This whole process is time-consuming.

Support Vector Machine

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well. Support Vectors are simply the coordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line). It is effective in cases where the number of dimensions is greater than the number of samples. But it doesn’t perform well, when we have large dataset because the required training time is higher.

Naive Bayes

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. It performs well in case of categorical input variables compared to numerical variable(s). For numerical variables, normal distribution is assumed (bellNaive Bayes Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. It performs well in case of categorical input variables compared to numerical variable(s). For numerical variables, normal distribution is assumed (bell curve, which is a strong assumption). Also numerical variables are very less correlated. On the other hand, naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.

LightGBM

LightGBM is Gradient Boosting ensemble model which is faster in speed and accuracy as compared to bagging and adaptive boosting. It is capable of performing equally good with large datasets with a significant reduction in training time as compared to XGBOOST. But parameter tuning in LightGBM should be done carefully.

Observations

Model	Accuracy	Recall	Precision	F-score	ROC-AUC
Logistic Regression (R)	0.9165	0.9244	0.9880	0.9551	0.6302
Logistic Regression (Python)	0.9161	0.2651	0.6757	0.3808	0.6257
Random Forest(ntrees=5)	0.8926	0.0822	0.3069	0.1297	0.5311
Random Forest(ntrees=10)	0.9031	0.0185	0.5652	0.0358	0.5084
Random Forest(ntrees=30)	0.9029	0.0032	0.7826	0.0063	0.5015
Naive Bayes	0.9228	0.9331	0.9848	0.9582	0.6745
SVM	0.9073	0.5365	0.9009	0.1089	0.5266
LightGBM	0.89	0.6494	0.4591	0.5379	0.7834

NOTES:

1. Due to some VS Build issues in the system I was not able to perform Light GBM in the R environment but could run the same algorithm in Python environment.

2. Also, the SVM model wasn't converging even after training the model for 8 hours. Hence, SVM model couldn't be implemented in R.

3. The submission using R environment is done on the basis of Naive Bayes Modelling.

4. The submission using Python environment is done on the basis of LightGBM

Conclusions

This was a classification problem on a typically unbalanced dataset with no missing values. Predictor variables are anonymous and numeric and target variable is categorical. Visualising descriptive features and finally I got to know that these variables are not correlated among themselves.

I chose LightGBM as my final model having highest AUC score since it was performing well on the given imbalanced data. Logistic Regression after applying SMOTE analysis did give a good performance measure metrics but its accuracy came out to be lower than the accuracy of Light GBM model. The other models might be giving a good accuracy, recall and precision but it would be a trap considering the dataset was imbalanced. My second preference would be Naive Bayes because it took less time to train and and has given a reasonable performance metrics.

It is important to have accuracy, recall and AUC score to be on the higher side, if not the banks might lose valuable business if they miss out on potential customers.