Banking Transaction Prediction - Capstone Project

Rohit Goyal

01/05/2021

1. Introduction

1.1 Objective

The objective of this project is to develop a model that will predict whether a given customer of a bank will make a specific type of transaction in future. Such models are used by banks to identify relevant products for their customers and make targeted recommendations to them through their various advertising and outreach channels.

1.2 Dataset Used

The **Santander Customer Transaction Prediction** dataset which is available on **kaggle.com** (*https://www.kaggle.com/c/santander-customer-transaction-prediction/data*) will be used to build the prediction model using machine learning. The dataset contains 200,000 anonymized transaction records with 200 numeric feature variables and a target variable which is 0 or 1, denoting whether the trasanction is made or not. This data was made available by Santander bank as part of a competition on Kaggle.

1.3 Summary of Steps

- Prepare the the transaction data set for analysis
- Split the dataset into 2 parts - training set (to build the predicition model) and validation set (to check the performance of model)
- Explore the dataset to identify trends
- Evaluate different prediction models using training data and select final model
- Assess performance of the final model on the validation dataset

2. Detailed Analysis of Steps

2.1 Data Preparation

- The first step is to download the data set. For this project, the dataset was downloaded in advance from Kaggle website (which requires registration and login to access the data). The original data file was ~300 mb in size so it was split into 10 chunks and uploaded on github in rda format. As a first step to begin the analysis, the data files are downloaded from github.
- The individual data files are then combined to create the full dataset. Target variable is transformed from 0/1 to no/yes.

- This dataset is then split into 2 parts.

1. 80% as training set (to build the model).
2. Remaining 20% as validation set (to test our final model). The proportion of customers who make a transaction is expected to be much less than the those who dont. So 20% of data is kept aside instead of the typical 10% to ensure that the final model gets a chance to be evaluated on sufficient positive cases.

- Given that there are 200 variables (features), next it is checked whether any of the variables can be discarded if they are not informative. A check is done to see if any variables have **near zero variance**. No variables satisfy this criterion. Lowest 5 values are shown below.

Variables with near zero var
0

Lowest 5 Std Devs
0.0071864
0.1524959
0.1711306
0.1848531
0.1901298

- The variables are then standardized using matrix operations by subtracting mean and dividing by std deviation.

2.2 Data Exploration and Visualization

- The data is now ready for initial exploration. Here is a snippet of the prepared data. Only the first 3 and last 2 feature variables are shown.

target	var_0	var_1	var_2	var_198	var_199
no	-0.5767262	-1.2745783	0.4509082	-1.0272359	0.2152783
no	0.1250576	-0.1288915	-0.6684027	0.6952834	-0.5424063
no	-0.2771310	0.0363425	0.8168928	0.7044779	-0.5242791
no	0.2617721	-0.1700270	0.7159394	-1.0934228	-0.1362294
no	0.3711700	0.3824836	-0.5179949	-1.6557128	0.0123240
no	0.9460680	-1.5716732	1.1966989	-2.3940919	0.6533215

- The dimensions of training dataset (rows, columns) including target variable are as follows:

Dimensions: rows, columns
159999
201

- The proportion of yes/no for the target variable is checked and is as follows. This shows that class imbalance is present and may need to be factored during model development.

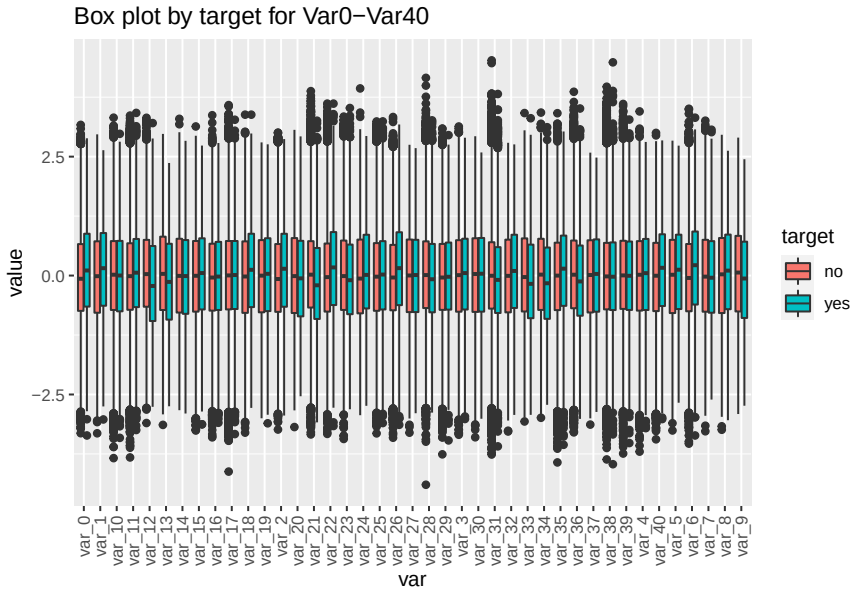
Class	Count
no	143921
yes	16078

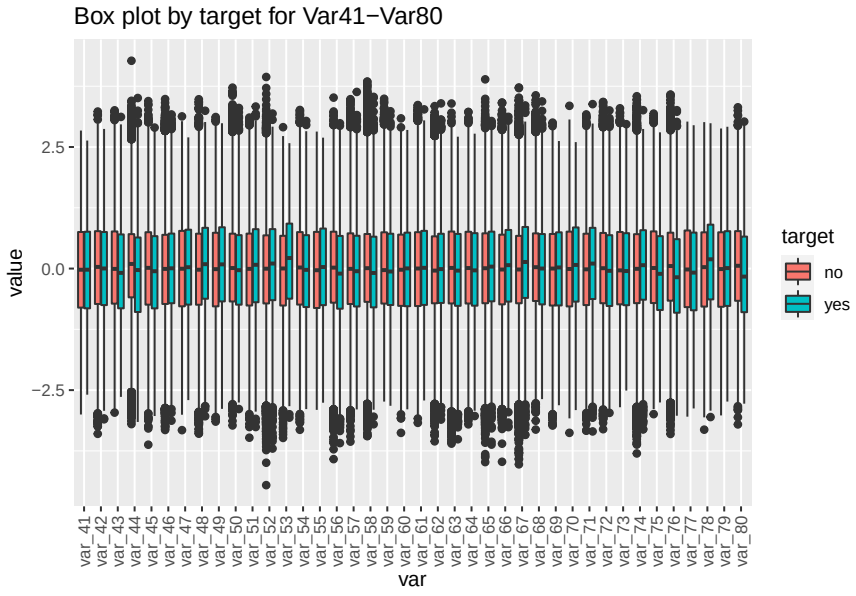
- **Principal component analysis (PCA)** is performed to see if we can transform our 200 feature variables and work with a smaller set by taking the most important variables without significant information loss. Looking at the cumulative variable importance from the PCA output, it is observed that the top most components are not providing any significant information. Therefore, it will not be possible to reduce our number of variables through PCA. For instance, even if we take upto PC100, we only get cumulative importance value of 0.51546.

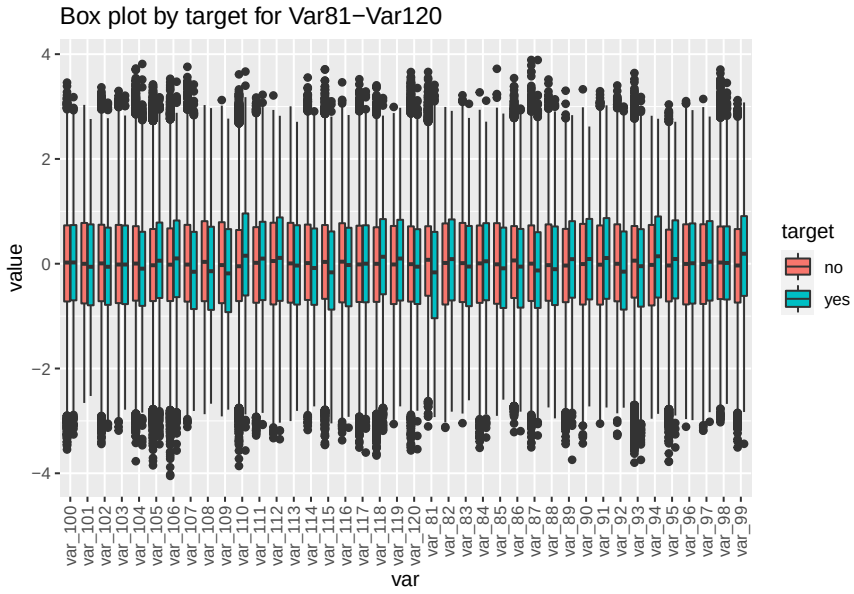
```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
## 0.00611 0.01146 0.01679 0.02212 0.02744 0.03275 0.03806 0.04336 0.04866 0.05396
##    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20
## 0.05924 0.06452 0.06980 0.07508 0.08034 0.08560 0.09086 0.09611 0.10136 0.10660
##    PC21    PC22    PC23    PC24    PC25    PC26    PC27    PC28    PC29    PC30
## 0.11184 0.11707 0.12230 0.12753 0.13275 0.13797 0.14319 0.14840 0.15361 0.15881
##    PC31    PC32    PC33    PC34    PC35    PC36    PC37    PC38    PC39    PC40
## 0.16402 0.16922 0.17441 0.17960 0.18479 0.18998 0.19517 0.20034 0.20552 0.21069
##    PC41    PC42    PC43    PC44    PC45    PC46    PC47    PC48    PC49    PC50
## 0.21586 0.22102 0.22619 0.23135 0.23650 0.24165 0.24680 0.25195 0.25709 0.26222
##    PC51    PC52    PC53    PC54    PC55    PC56    PC57    PC58    PC59    PC60
## 0.26736 0.27249 0.27762 0.28275 0.28788 0.29300 0.29812 0.30323 0.30835 0.31346
##    PC61    PC62    PC63    PC64    PC65    PC66    PC67    PC68    PC69    PC70
## 0.31857 0.32368 0.32878 0.33387 0.33897 0.34406 0.34914 0.35423 0.35931 0.36439
##    PC71    PC72    PC73    PC74    PC75    PC76    PC77    PC78    PC79    PC80
## 0.36947 0.37454 0.37961 0.38467 0.38974 0.39480 0.39986 0.40491 0.40997 0.41502
##    PC81    PC82    PC83    PC84    PC85    PC86    PC87    PC88    PC89    PC90
## 0.42007 0.42512 0.43016 0.43520 0.44024 0.44527 0.45030 0.45534 0.46036 0.46539
##    PC91    PC92    PC93    PC94    PC95    PC96    PC97    PC98    PC99   PC100
## 0.47041 0.47543 0.48044 0.48545 0.49046 0.49547 0.50047 0.50547 0.51047 0.51546
##   PC101   PC102   PC103   PC104   PC105   PC106   PC107   PC108   PC109   PC110
## 0.52046 0.52544 0.53043 0.53541 0.54039 0.54537 0.55034 0.55531 0.56028 0.56524
##   PC111   PC112   PC113   PC114   PC115   PC116   PC117   PC118   PC119   PC120
## 0.57021 0.57516 0.58012 0.58507 0.59002 0.59497 0.59992 0.60486 0.60980 0.61474
##   PC121   PC122   PC123   PC124   PC125   PC126   PC127   PC128   PC129   PC130
## 0.61967 0.62461 0.62954 0.63446 0.63939 0.64431 0.64923 0.65415 0.65907 0.66398
##   PC131   PC132   PC133   PC134   PC135   PC136   PC137   PC138   PC139   PC140
## 0.66889 0.67379 0.67870 0.68360 0.68850 0.69339 0.69828 0.70317 0.70805 0.71293
##   PC141   PC142   PC143   PC144   PC145   PC146   PC147   PC148   PC149   PC150
## 0.71781 0.72269 0.72756 0.73243 0.73729 0.74216 0.74702 0.75188 0.75673 0.76158
##   PC151   PC152   PC153   PC154   PC155   PC156   PC157   PC158   PC159   PC160
## 0.76643 0.77128 0.77612 0.78096 0.78580 0.79064 0.79547 0.80030 0.80513 0.80995
##   PC161   PC162   PC163   PC164   PC165   PC166   PC167   PC168   PC169   PC170
## 0.81477 0.81959 0.82440 0.82921 0.83402 0.83882 0.84362 0.84842 0.85322 0.85801
##   PC171   PC172   PC173   PC174   PC175   PC176   PC177   PC178   PC179   PC180
## 0.86280 0.86758 0.87236 0.87714 0.88192 0.88669 0.89145 0.89622 0.90098 0.90574
##   PC181   PC182   PC183   PC184   PC185   PC186   PC187   PC188   PC189   PC190
## 0.91049 0.91524 0.91998 0.92473 0.92946 0.93420 0.93893 0.94366 0.94839 0.95311
```

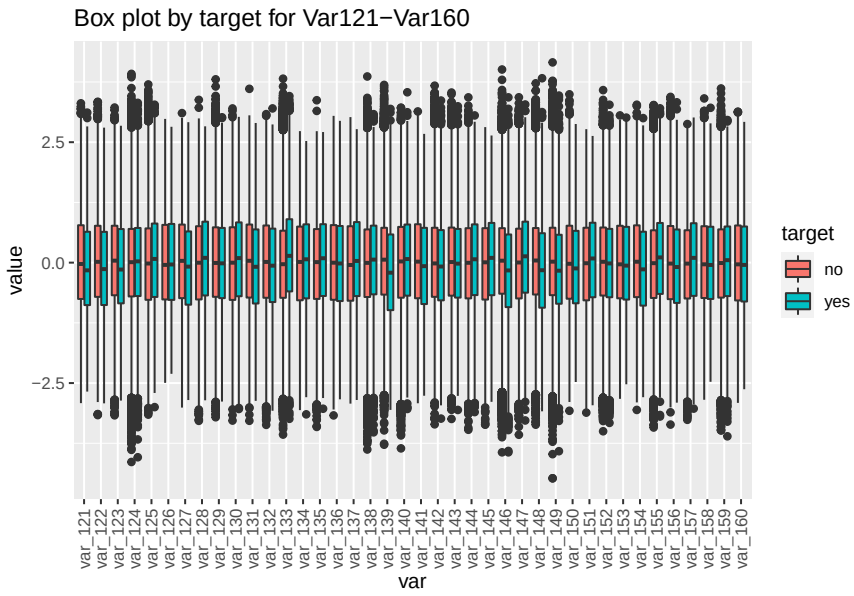
```
##    PC191    PC192    PC193    PC194    PC195    PC196    PC197    PC198    PC199    PC200
## 0.95783 0.96254 0.96724 0.97194 0.97664 0.98133 0.98601 0.99069 0.99535 1.00000
```

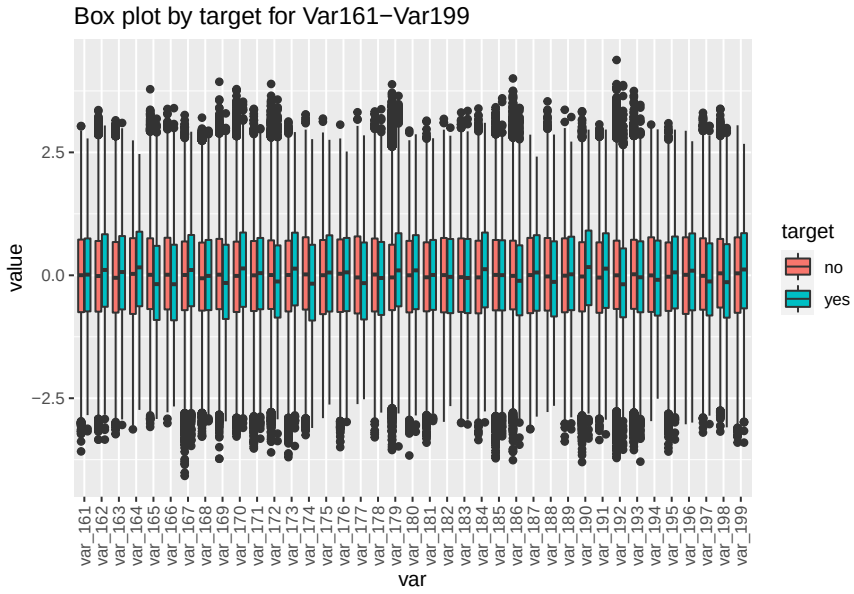
- **Box plots** are created for all our 200 feature variables to analyse the variability by target variable. It is noted that the feature ranges overlap a lot for both target values. There are no features which stand out as significant differentiators.





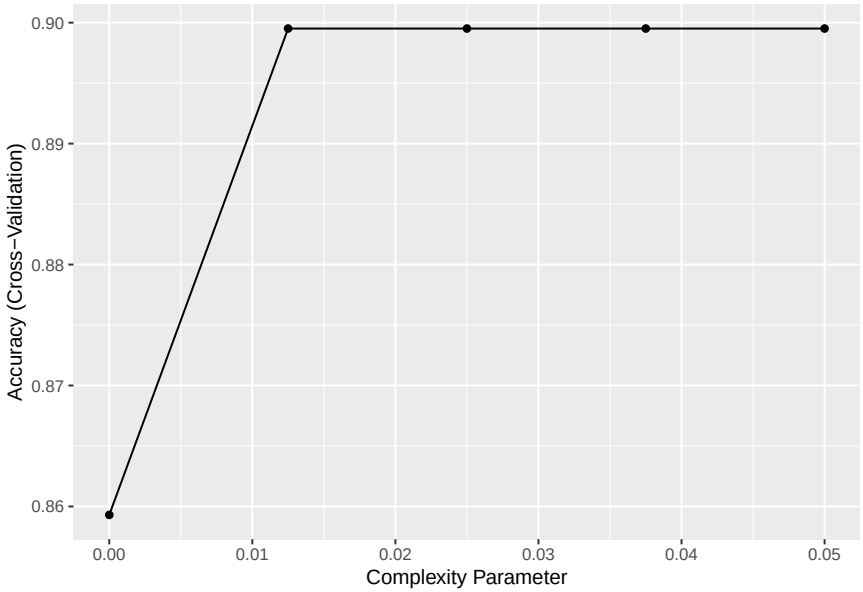

Box plot by target for Var81−Var120
value
4
2
0
−2
−4
var_100
var_101
var_102
var_103
var_104
var_105
var_106
var_107
var_108
var_109
var_110
var_111
var_112
var_113
var_114
var_115
var_116
var_117
var_118
var_119
var_120
var_81
var_82
var_83
var_84
var_85
var_86
var_87
var_88
var_89
var_90
var_91
var_92
var_93
var_94
var_95
var_96
var_97
var_98
var_99
var
target
no
yes


Box plot by target for Var121−Var160
value
2.5
0.0
−2.5
var_121
var_122
var_123
var_124
var_125
var_126
var_127
var_128
var_129
var_130
var_131
var_132
var_133
var_134
var_135
var_136
var_137
var_138
var_139
var_140
var_141
var_142
var_143
var_144
var_145
var_146
var_147
var_148
var_149
var_150
var_151
var_152
var_153
var_154
var_155
var_156
var_157
var_158
var_159
var_160
var
target
no
yes



2.3 Model Development

- The training set is first further split into training and test data sets for model development.
- For model comparison, 3 metrics will be captured - Overall Accuracy, Sensitivity and Precision. Given the low pervalance of our positive target class (i.e prediction that customer will do the transaction), Sensitivity is of primary importance.
- Cross validation will be used for training all models.
- Different types of classification algorithms will be tried to see which give the best results.
- **Model 1 - Rpart** - The first model chosen is Rpart as we have a classification problem with large number of features. Rpart algorithm is used with tuning parameter of cp. It achieves an overall accuracy of near 90% however, fails to pickup any of the target = yes cases. This is the reason why Sensitivity is zero and Precision is NA.



Model	Accuracy	Sensitivity	Precision
Rpart	0.8995031	0	NA

- **Model 2 - glm** - For this model, the logistic regression is used. A Sensitivity of 0.2702114 and Precision of 0.6805012 is observed. This is an improvement over Rpart. However, still the Sensitivity is quite low.

Model	Accuracy	Sensitivity	Precision
Rpart	0.8995031	0.0000000	NA
glm	0.9139089	0.2702114	0.6805012

- **Model 3 - lda** - Next we try a generative model - Linear Discriminant Analysis (lda). We do not attempt Quadratic discriminant analysis (qda) which would be computationally very intensive due to the large number of features. A Sensitivity of 0.2792289 and Precision of 0.6656783 is observed. The Sensitivity is slightly better than glm but it is at the expense of loss in Precision.

Model	Accuracy	Sensitivity	Precision
Rpart	0.8995031	0.0000000	NA
glm	0.9139089	0.2702114	0.6805012
lda	0.9134715	0.2792289	0.6656783

- Based on the model results thus far, it is observed that the overall accuracy is seen near 90% but the models perform poorly in identifying target=yes cases (as seen with low Sensitivity values). This is likely because of the high prevalance of the target=no cases in our data (as noted during data exploration stage). To cater to this problem, models glm and lda are retrained by **down sampling** the prevalent class (target = no). Rapart is discarded at this stage given that it had zero Sensitivity.
- **Model 4/5 - With Down Sampling** - The results for glm and lda with down sampling are presented below along with the earlier model results. There is a significant improvment in Sensitivity (above

70%) with down sampling observed for both models but it comes at the cost of reduction in Precision (below 30%). There is also a reduction in overall Accuracy from around 90% down to about 78%.

Model	Accuracy	Sensitivity	Precision
Rpart	0.8995031	0.0000000	NA
glm	0.9139089	0.2702114	0.6805012
lda	0.9134715	0.2792289	0.6656783
glm + down sample	0.7765070	0.7754975	0.2794711
lda + down sample	0.7796631	0.7714552	0.2820280

- **Final Model Selection** - Selection is made of a model with Down sampling because it gives much better Sensitivity. Therefore, **glm with down sampling** is selected as the final model because it gives the highest Sensitivity of 0.7754975. While it has a low Precision (high proportion of False Positives), this is an acceptable trade off to allow the bank to identify almost 80% of customers who are likely to do a transaction (True Positives). Overall Accuracy for the selected model is close to 80% which implies that it also does a decent job with identifying Customers who will not do a transaction (True Negatives) and therefore will not be targeted by sales team.

3. Model Performance with Validation Data Set

- The final model performance is now tested on the Validation data set, i.e the final hold-out set.
- As shown below, the Overall Accuracy, Sensitivity, and Precision are close to what we saw for our final model with training data. Infact, we get a higher Sensitivity (0.7937811) as compared to what we saw in training (0.7754975)

Model	Accuracy	Sensitivity	Precision
Validation Results - glm + down sample	0.7806805	0.7937811	0.2865739

4. Conclusion

- The stated objective of building a customer transaction prediction model is achieved as presented in this report.
- The model is built using the Santander Bank dataset available on kaggle.com.
- The dataset is explored for any visible trends in its feature variables for the two classes, however no distinct trends are found. Given the large number of feautres in the data, an attempt is made to reduce the dimension through couple of techniques but is concluded to be not feasible.
- The following models are evaluated: Rpart, glm and lda. Subsequently all these except Rpart are evaluated with down sampling technique. glm with down sampling is selected as our final model as it gives the best Sensitivity score.
- The final model is tested on validation dataset and the model performance is consistent with what is observed on training data.
- Banks are increasingly using data analysis techniques such as those presented in this report to achieve better outcomes for the bank as well as their customers. There is a strong push within the banking community to leverage the troves of transaction information available within their systems to gain useful insights.

4.1 Limitations and future work

- The dataset makes one appreciate the computation capacity constraints faced by data analysts in dealing with large datasets having large number of variables. More computationally intensive algorithms such as Random Forests could be explored.
- The dataset is also a good example of real world prediction problems where model building is challenging when data does not provide any any black and white trends. Other models and techniques could be explored which specifically cater to such scenarios.
- In this project, the down sampling technique is used to handle the class imbalance problem. There are other techniques to handle this which could be evalulated to see if they give better results.

5. References

https://www.r-bloggers.com/2016/12/handling-class-imbalance-with-r-and-caret-an-introduction/