# **Santander Customer Transaction Prediction**

**Presented by the team:**

**Tommy Entwistle, Shaleen Swarup, Harshitha Kuriminisetty, & Koller Adzick**

**Table of Contents**

[**1. Project Definition & Introduction**](#_7l5ch1k6wn) **1**

[**2. Business Applications**](#_1t4x7hoq4vhk) **2**

[**3. Technical Details (Tech specs)**](#_ou3asz7o52oc) **3**

[**4. Methodology**](#_z7ndrbsp6zvv) **3**

[**5. Analysis**](#_q5lq39tiy38j) **4**

[**5.1 Modeling**](#_1r4xi0ms6rek) **4**

[**5.1.1 Feature Selection**](#_99aifcx03nen) **5**

[**5.1.2 Feature Engineering and Data Augmentation**](#_aqw2jrza20l2) **6**

[**5.1.3 Algorithm Selection**](#_oesx82ybp5q6) **9**

[**5.2 Model Evaluation**](#_dt70hhph6t5o) **14**

[**5.2.1 Training, Testing and Validation**](#_izbmrrqbvyvx) **14**

[**5.2.2 Performance Evaluation**](#_9euzg7oh665f) **15**

[**6. Business Insights**](#_7n8pz07u8ilf) **15**

[**7. Future Analysis and Improvements**](#_gmxhlctb2lem) **17**

[**8. File Locations**](#_vwizyyipp5zn) **17**

[**9. References**](#_5h6148uw38qa) **17**

# 1. Project Definition & Introduction

Santander Bank is a commercial and consumer bank with branches in various countries in central Europe as well as the Northeastern United States. As a bank they are focused on providing the best service for their customers as well as making sure they prioritize healthy financial habits.

In this scenario the data team at Santander is looking to predict which customers will be making a transaction in the future given certain variables. Although the variables in this dataset are not labeled, there can be value in helping to predict whether a certain customer will be making a purchase or not. The model will be able to help the bank predict what types of customers should be targeted for promotions and what new types of customers they should be targeting.

Currently, we are assuming that Santander is using a majority classifier. This model’s performance is deceivingly high, with an accurate prediction of about 90%. This is due to the nature of the imbalanced data set. Roughly 90% of the data are labeled as non-purchasers, meaning that guessing all of them as non-purchasers yields a model accuracy of about 90%. The key challenge here is to not only accurately predict the non-purchasers, but also the purchasers as well. So, the evaluation metric as designated by Santander will be AUC, a measure between 0 and 1, which essentially measures the quality of a model’s predictions over different classification thresholds. What is important here is that this measure isn’t a deceptive measure in this case like accuracy. Santander’s current AUC score is 50%. We will use this as the current base model score for comparison.

# 

# 2. Business Applications

Santander will be able to use our model in order to lower the amount of wasted advertisement spending and make sure they are targeting the right people for their products. Being able to predict who will make a transaction or purchase of a product with the bank will help them to design their products in order to specifically target those individuals who have similar characteristics to those who usually purchase. As banks have begun to use more and more online marketing resources it will be crucial to make sure the banks efforts are directed towards the right people. Banking marketing budgets are stretched by trying to reach the most people possible as well as making sure that these are the same people who will be most likely to make a purchase.

Although the data and outcome of the analysis is irrespective of the amount of money in a given transaction with the bank it can be assumed that the customer may wish to continue doing business with Santander. This type of loyalty is crucial but is very difficult to calculate but will be very valuable to Santander in the long run. According to a 2020 article the average cost of a bank to acquire a new customer is around $300 (3). A proper model to predict the most likely customers will help Santander to slash this cost and get more return for their marketing efforts.

# 

# 3. Technical Details

The technical environment used is as follows:

|  |  |
| --- | --- |
| **OS** | **Windows** |
| **Language** | **Python** |
| **Version** | **3.10.7** |
| **Distribution** | **Anaconda** |
| **year** | **2021** |
| **month** | **04** |
| **day** | **21** |

We used **Anaconda Python Distribution** of **Python 3.10.7** and used the following Python packages:

|  |
| --- |
| **Packages** |
| Numpy |
| Pandas |
| sklearn |
| lightgbm |

# 4. Methodology

The first step in our analysis after defining the problem was to look into the Santander customer data. Our goal was to explore the data by looking for the percent of missing values, attribute characteristics, noisiness, target attributes, correlations between attributes, and identify any promising transformations that could be applied. Next, we prepared the data for modeling. The first step here was to clean the data. Things like missing values and outliers if any were dealt with here. We then upsampled due to the imbalanced nature of the dataset. Feature selection, engineering, and scaling were considered as well. After data preparation, the core modeling processing started. We began with some more basic models and moved to more complex, tuning hyperparameters along the way. Interestingly, all features were uncorrelated with one another, so all of them were used in the modeling process. After rounds of testing and comparing, the best model was chosen based on Santander’s evaluation criteria. Finally, the model was tested and it’s possible business applications were found.

# 5. Analysis

The ultimate goal for this project is to predict with the highest level of performance the customers who are most likely to make a purchase. With this in mind, most of our modeling process was designed, tested, and adjusted to achieve that end. The side effects here are potentially getting a model that fits really well to the training data, but then does poorly when new data arrives. This is called overfitting. To address this concern, model performance was measured only using the training data set given to us from Santander, then finally run on the testing data set, which serves as the “new” data coming in, of which our model has never seen before.

## 5.1 Modeling

This is a classification problem with two possible classes: non-purchaser (0) and purchaser (1). Our goal is to find the model that is the simplest, has the best performance, all the while fast enough to be implemented in a reasonable amount of time.

### 

### 5.1.1 Feature Selection

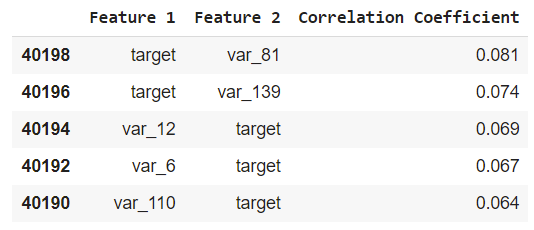
Due to the nature of the dataset, feature selection was difficult. There were no variable names provided by Santander, so an intuitive sense of which features were important could not be used. We therefore used a numeric approach for feature selection.

**Correlations**

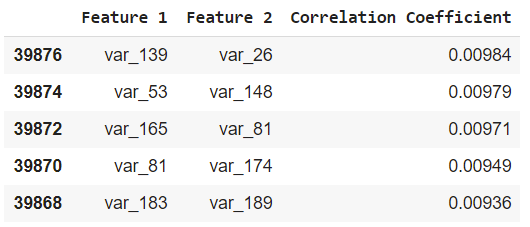
The first approach was to look at the relationship between features. Features from var\_0 to var\_199 have extremely low correlation between each other in both training set and test set. The lowest correlation between variables is 2.7e-8 and it is in the training set (between var\_191 and var\_75). The highest correlation between variables is 0.00986 and it is in the test set (between var\_139 and var\_75). The target variable has slightly higher correlations with other features. The highest correlation between a feature and target is 0.08 (between var\_81 and target).

Because of low correlations amongst features, we could not remove any features on the grounds that they were redundant. This also made us wary of using PCA to find the principal components. Since PCA is mainly useful when there is multicollinearity or in other words redundant features i.e. correlated features, and there is no multicollinearity present, we chose not to use PCA. All of the above considered, we chose to use all of the features present in the data set in our model.

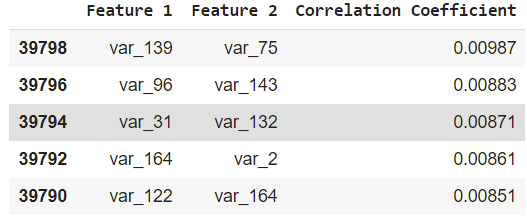
**Top 5 correlations in training set:**



**Top 5 Highest Correlations between variables in the Training Set:**

****

**Top 5 Highest Correlations in the Test Set**

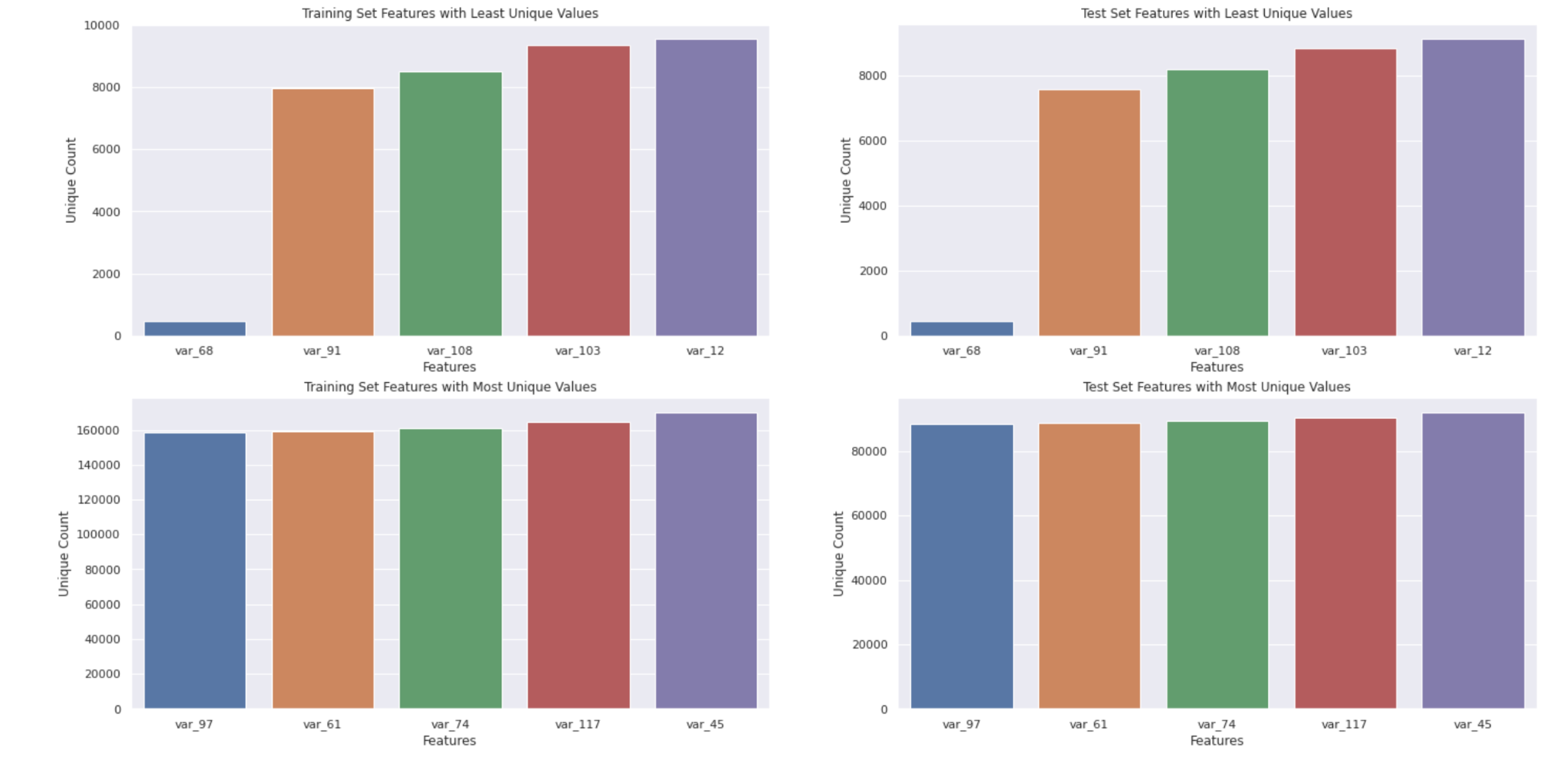
****

### 5.1.2 Feature Engineering and Data Augmentation

**Unique Value Count**

The lowest unique value count belongs to var\_68 which has only 451 unique values in the training set and 428 unique values in the test set. 451 and 428 unique values in 200000 rows are such small amounts that var\_68 could even be a categorical feature. The highest unique value count belongs to var\_45 which has 169968 unique values in the training set and 92058 unique values in the test set. Every feature in the training set has higher unique value counts compared to features in the test set.

The lowest unique value count difference is in the var\_68 feature (Training Set Unique Count 451, Test Set Unique Count 428). The highest unique value count difference is in the var\_45 feature (Training Set Unique Count 169968, Test Set Unique Count 92058). When the unique value count of a feature increases, the difference between training set unique value count and test set unique value count also increases. The explanation of this situation is probably the synthetic records in the test set.

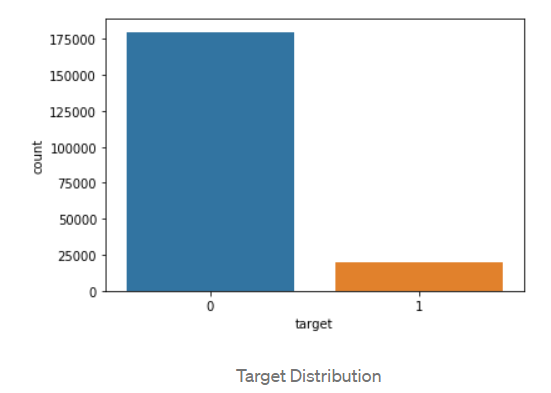


**Feature Distributions in Training and Test Set**

Training and test set distributions of features are not perfectly identical. There are bumps on the distribution peaks of the test set because the unique value counts are lesser than the training set. Distribution tails are smoother than peaks and spikes are present in both training and test sets.

**Separating Real/Synthetic Test Data and Magic Features**

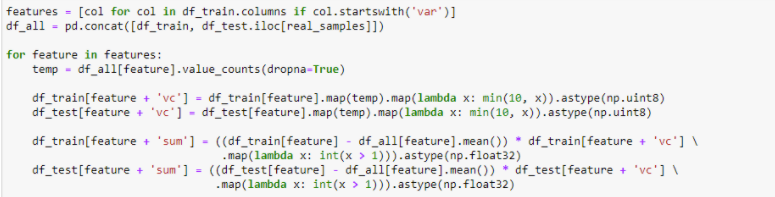
Using unique value counts in a row to identify synthetic samples. If a row has at least one unique value in a feature, then it is real, otherwise it is synthetic. It successfully identifies synthetic samples in the entire test set. This way the unusual bumps on the distribution peaks of test set features are captured. The magic features are extracted from the combination of the training set and real samples in the test set.

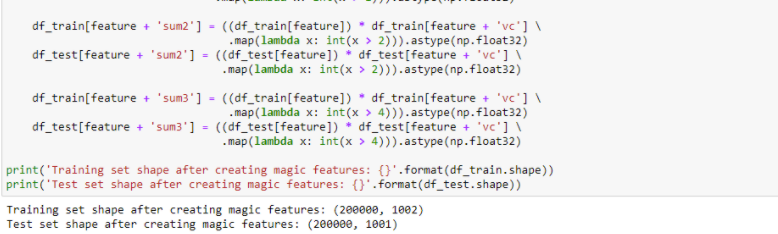
****

Only 10% of the features are Target = 1, which implies that there is imbalance in the data. To overcome the imbalance in the data, we have two options Undersampling and Oversampling the data. We didn’t want to sacrifice the information in the data, so we chose the oversampling of the data. As target = 1 are in less numbers, we are oversampling Target = 1 by 3 times which increases our number of samples to train on by increasing the percentage of samples where Target = 1, solving the imbalance problem. We also doubled Target = 0 samples, so the new sample distribution is similar to the original.

**Creating new features**

We extracted real samples from test data and combined them with training data to create new features based on how many times values repeat themselves.



****

**Data augmentation**

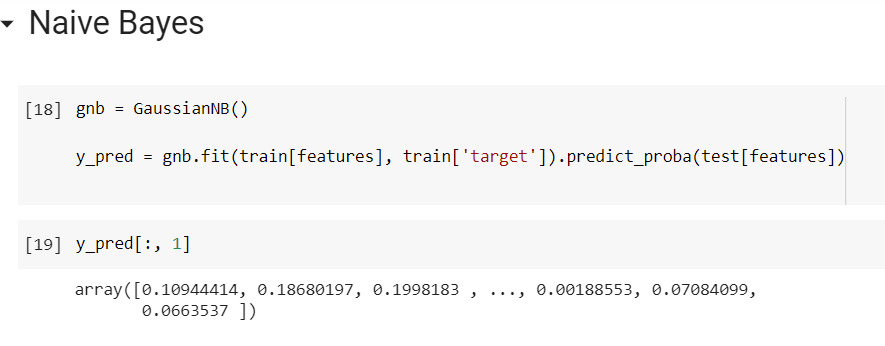




### 5.1.3 Algorithm Selection

**Model 1: Naive Bayes**

To begin with algorithm selection, we first tried a Naive Bayes model. We chose this first due to its simplicity and speed in handling large datasets. The training data set is very large and we ideally wanted to use as much of it as possible. The more data the better. Other models like SVM and Decision tree failed to produce results in a reasonable amount of time, having to be stopped after hours with no results. Code snippets for Naive Bayes are shown below:

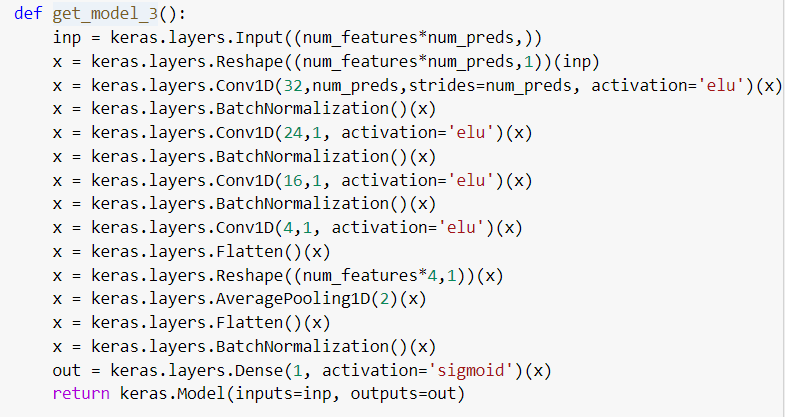


Our Naive Bayes model achieved a AUC score of 0.896, performing better than the baseline model. Though there was better performance here, having researched other approaches, we decided a more complex model would be better.

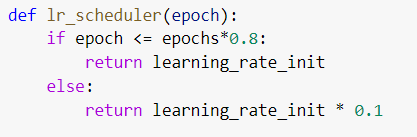
**Model 2: Neural Network**

We tried the Neural network model with different layers. The optimum value is reached when we used 4 layers with batch size of 4000 and used all the features after doing the data augmentation.

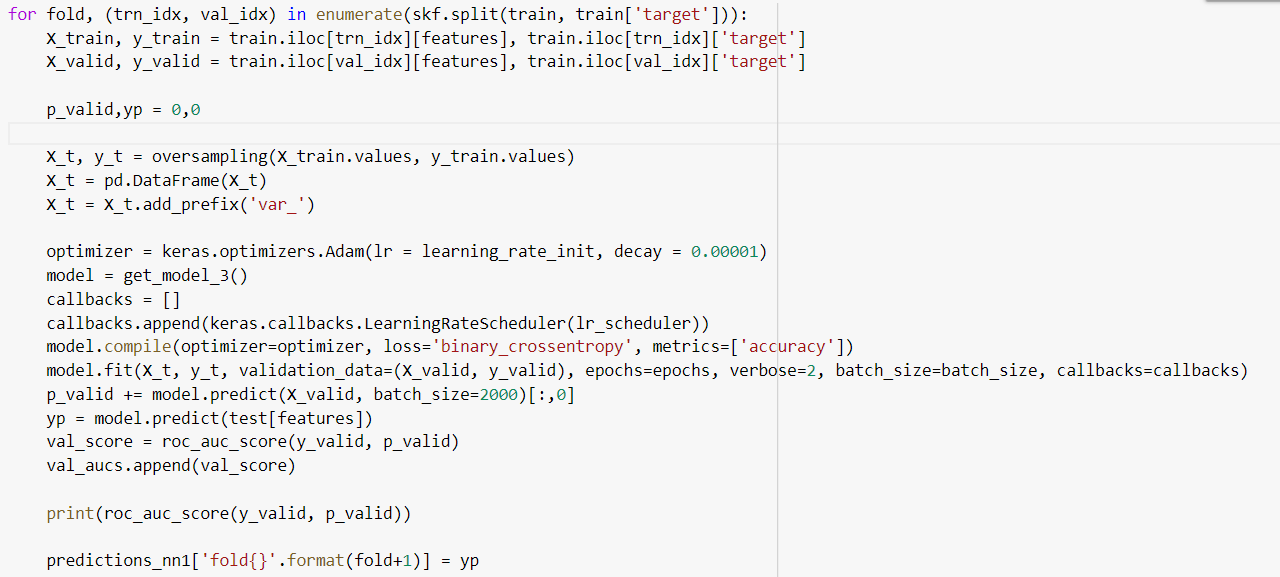
The model with the best set of layers and parameters is given below.



Instead of doing a fixed learning rate, learning rate is taken as the function of the epoch.

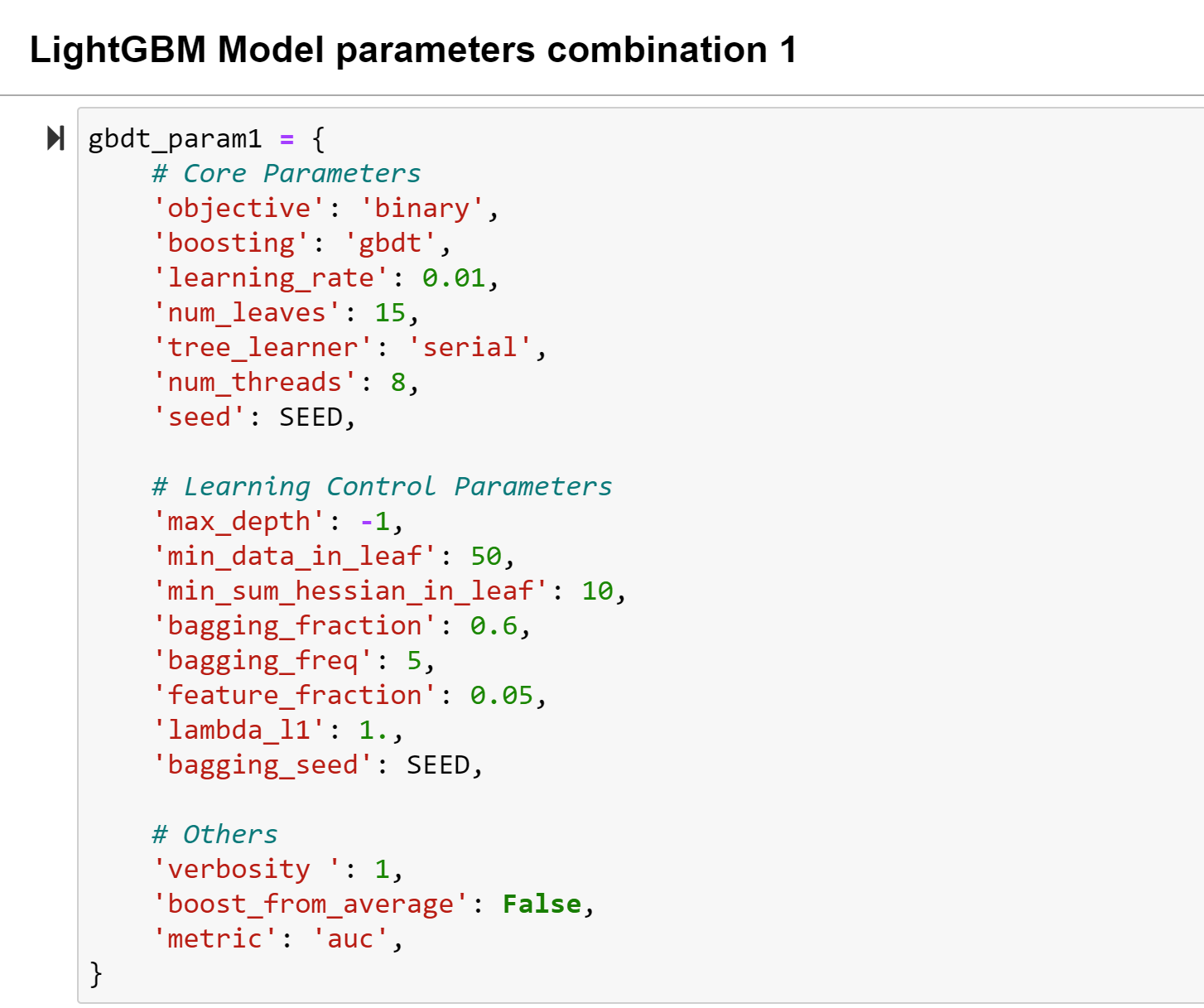


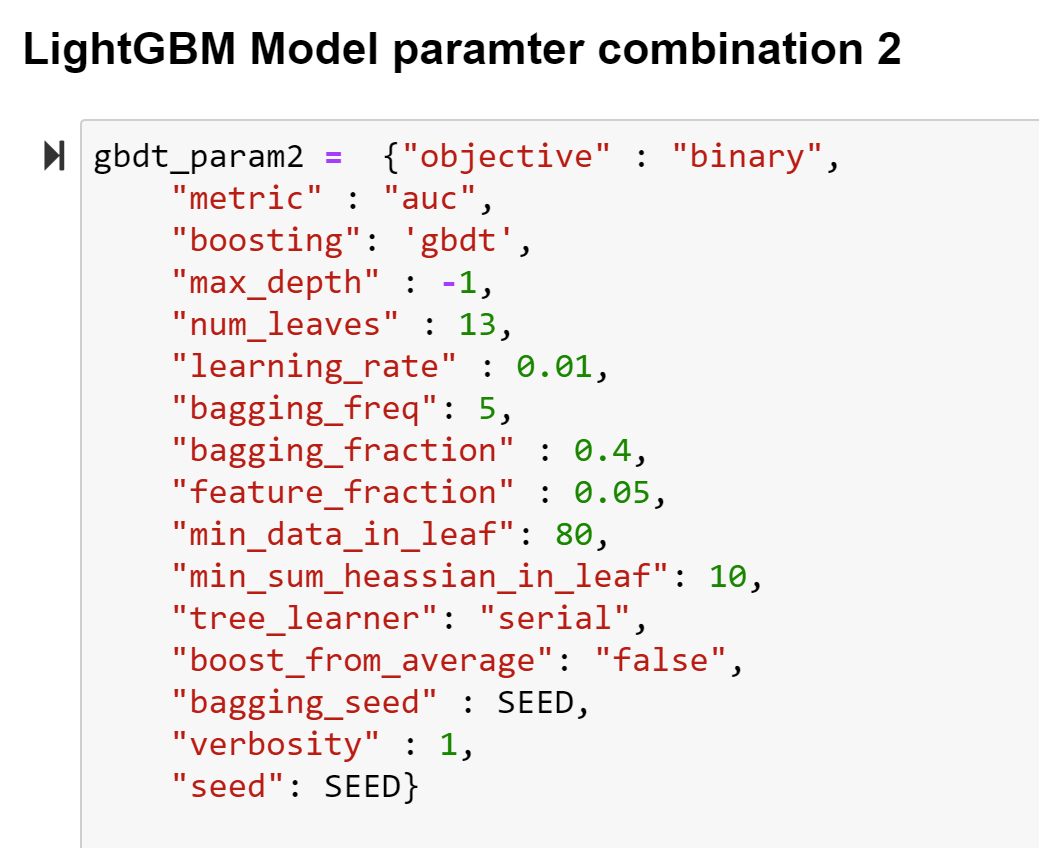
After running the model with 3 fold, stratified sampling using loss function as “binary cross entropy” , the score of the submission is 89.1%

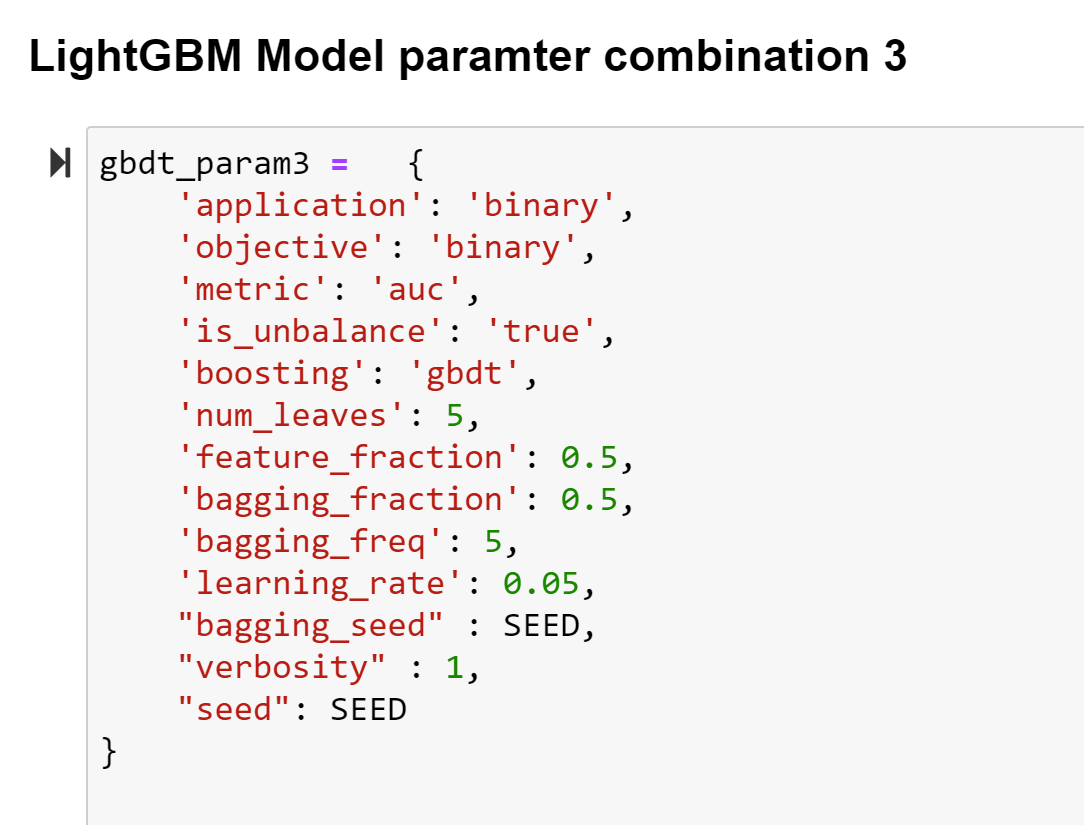


**Model 3: Light GBM**

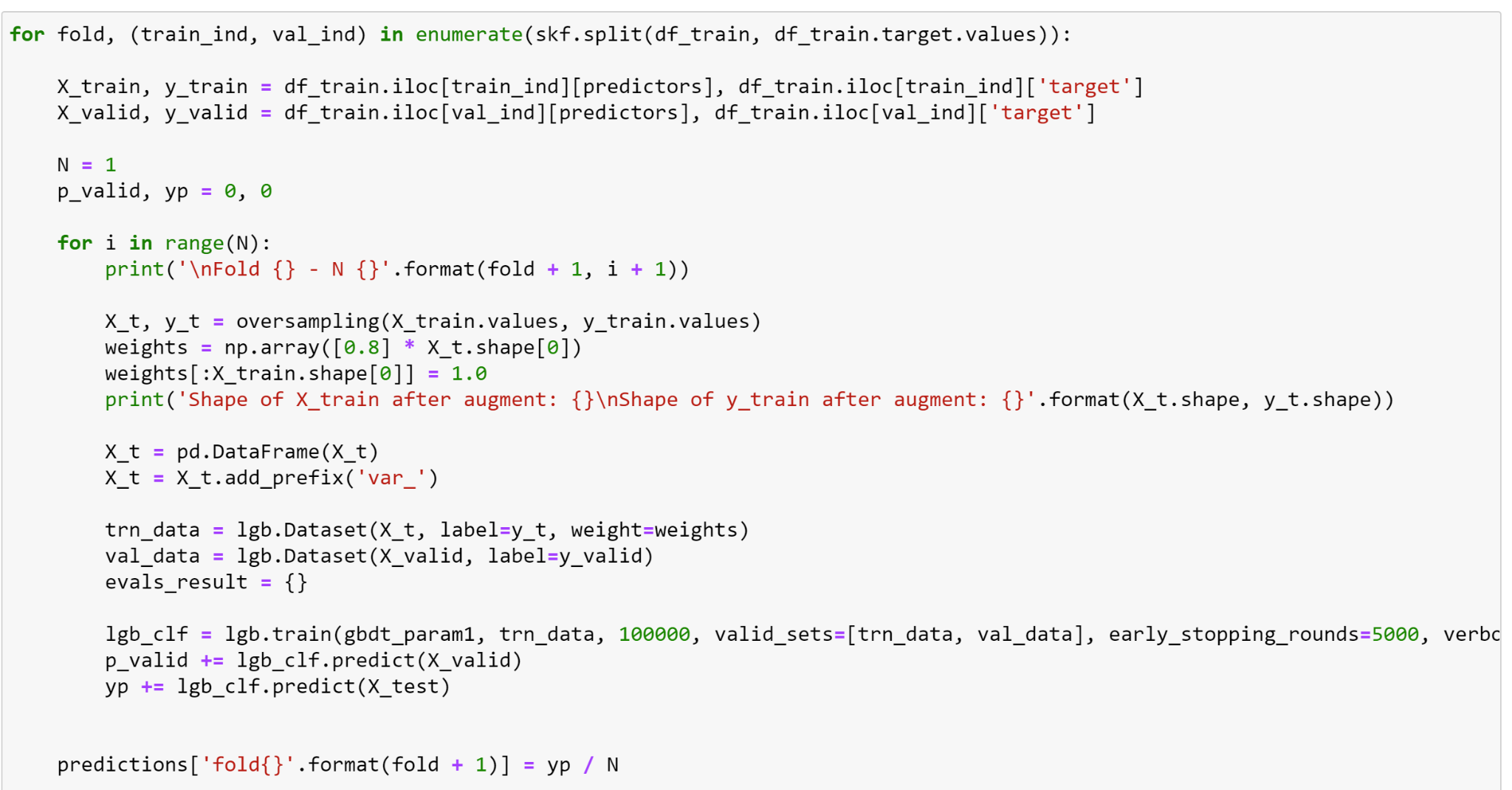
Our next and final model iteration was a LightGBM, short for Light Gradient Boost Machine. It is based on the decision tree algorithm and originally developed by Microsoft to be optimized for performance and scalability. As noted above, it is an ideal algorithm to use with large amounts of data and to make computations in a reasonable amount of time. Code snippets for the hyperparameters are given below:

****

****

****

We ran the LightGBM model through 5 fold stratified cross validation for each of the parameter combinations above.



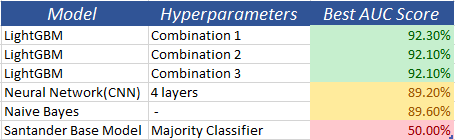
We found that Model parameter **Combination 1** gives us the best results. We were able to get an auc of 92.37% with these parameters.

## 5.2 Model Evaluation

### 5.2.1 Training, Testing and Validation

For predictive modeling in data science, it is very important to split data into training, testing and validation data. If the data isn’t at least split once in this way, the model will be prone to performing well on the data it is trained on and poor on any new data coming in. As a result, we split our data into training and validation data sets, utilizing a 3 fold stratified cross validation approach, ultimately testing our model on the testing data set provided by Santander, which was never seen by our model. Cross validation serves to give each section of the training data a chance to serve as the validation set. What is important here is that cross validation helps to mitigate the overfitting problem.

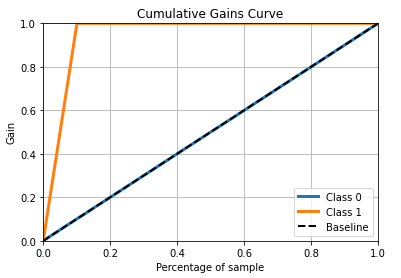
### 5.2.2 Performance Evaluation

****

As we can see from the above table, Santander’s base model performance is very low compared to the rest of the models we tested. The LightGBM model stands out as the winner with a best overall performance of 92.30% AUC.

# 6. Business Insights

To see what it means for business, We split the data as 90/10 split, and use 90% of the data to make the model and 10% to test the model and find the Cumulative response/gain curve(CRC). The plot for CRC is shown below.



In the above plot, X axis has the percentage sample of the total data used and the Y axis represents the percentage of actual transactions(label =1) in the sample to the total number of actual transactions present.

From the plot above, y reaches maximum and unchanging when x reaches 10%, which means that by taking the top 10% users when they are ordered with highest probability to low using our best model, we can accurately target customers who have the highest probability of making a transaction.

From the Santander document that gives us the cost of its services(link given in resources), we assume that the user detected from the model will at least check the balance which incurs a revenue of $2 for santander. Assuming that Santander uses google or facebook for making the ads, the cost of advertising can be taken as $0.5. By using the CRC curve, you can target the customers you need to send the ads which are only the top 10% of the total customers. From this assuming a 100% click through transaction rate, ROI is calculated as

Return of investment = (Revenue - Cost) / Cost = 100% \* ($2 - $0.5)/$0.5 =300%.

So, just by using 10% of the data, we can achieve Return of investment(ROI) of 300%, whereas before Santander’s base model achieved an ROI of -60%. With our model, Santander should be able to more effectively target customers for their marketing campaigns, which will decrease ad spend and ultimately increase profits.

Our model will help the business to gain insight into their customers and the best ways to target customers who are most likely to make a purchase. Santander can also use this insight to make sure their customers are receiving the promotions and advertisements for products that will be the most bene ficial. The model we have created can be implemented in order to automate which customers receive certain content based on their transaction prediction which means less wasted advertising and promotional spending for the bank. By accurately targeting existing customers to make transactions to increase revenue and lowering the cost of acquiring their customers will be crucial if Santander wants to compete with the top commercial banks in the world.

# 7. Future Analysis and Improvements

Although there are a number of challenges with this prediction task and business question involved there are some that could be addressed with future improvements to the process. One of the major challenges that can be addressed is making sure that all of the characteristics of the customers are labeled in the data. Having information about the customers characteristics will help the bank evaluate customer behavior based on common patterns in those that have purchased their products. The process may also be improved by including the lifetime value of the customers purchases with Santander. Knowing who is more likely to spend large amounts of money would help them perform additional cost benefit analysis.

# 8. File Locations

# GitHub: [link](https://github.com/HarshithaKuriminisetty/Santander-Customer-Transaction-Prediction)

# 9. References

1. <https://medium.com/analytics-vidhya/santander-customer-transaction-prediction-an-end-to-end-machine-learning-project-2cb763172f8a>
2. <https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60>
3. <https://www.bankdirector.com/issues/retail/scaling-customer-acquisition-through-digital-account-openings/>
4. For santander charges : <https://www.santanderbank.com/us/documents/22507/131529/Product+and+Fee+Schedule.pdf/09bc92e7-da91-447f-98e0-a39b2c2ee2a2>