UDACITY

Capstone Project

Report submitted for the partial fulfillment of nano-degree program

Divya Dayashankar Jaiswal

10/31/2020

Contents

1. Definition:

1.1 Project overview:

Machine learning (ML) has become an increasingly important part of IT today. This effect is seen both in how IT leverages machine learning to improve operations and in how IT supports and enables the lines of business (LOBs). Still, organizations have limited understanding on its effective use and have made limited progress in associating it with business outcomes.

Admittedly, The Companies which will lead in the future are those who will be interested in implementing the machine learning algorithms on the enormous amount of data base which they have, they will be the pioneers in their field.

STARBUCKS is one of flagship Worldwide companies which has been established since 31st March 1971 and have worldwide coffeehouse chain, and has a tremendous database of users , that is why I am interested in implementing my capstone project for STARBUCKS Capstone Challenge as I believe that I can implement a good Machine Learning Model for one of the most Worldwide prestigious companies.

Customers' Concerns are the goal for all companies all over the world , what people like? , how much they want to pay?, when do they are capable to pay? , what is the gender and age of those people who are interested and capable to pay? are very important questions, and the answer comes from Historical data which we have to implement a deep learning algorithms to it , and building machine Learning Algorithms according to those Historical data to maximize Companies s' profits.

1.2 Problem Statement:

STARBUCKS is one of flagship Worldwide companies which has been established since 31st March 1971 and have worldwide coffeehouse chain, and has a tremendous database of users , that is why I am interested in implementing my capstone project for STARBUCKS Capstone Challenge as I believe that I can implement a good Machine Learning Model for one of the most Worldwide prestigious companies.

Customers' Concerns are the goal for all companies all over the world , what people like? , how much they want to pay?, when do they are capable to pay? , what is the gender and age of those people who are interested and capable to pay? are very important questions, and the answer comes from Historical data which we have to implement a deep learning algorithms to it , and building machine Learning Algorithms according to those Historical data to maximize Companies s' profits.

1-The Below flow chart for the users Whom received, viewed, completed the offer and make transaction within the offer period and those customers are our target .

2-we will track the amount of money which has been spent by customers within the offer period and till the offer completed, to track the profits that can be gained by each customer for each offer.

3-we will do our statistics analysis and data visualization to understand the role of the features which controlling our model ,such as : Customers ‘s gender , Customers ‘s age ,customers ‘s membership , Customers ‘s income , offer duration …..etc.

4-We will do assumptions that all transactions executed within the offer period -for the customers whom completing the offers- will be through utilizing offers.

1.3 Problem Solving Strategy

1.3.1 Solution strategy:

We will Follow the below process in our Problem Solution:

- **Fetching the Data:**
  The Data sets mentioned in the previous slide to be converted to CSV Files , and to be ready for next step.

- **Clean /preparation Data:**
  1. Wrangle data and prepare it for training
  2. Remove duplicates, correct errors, deal with missing values, normalization, data type conversions, …etc.)

- **Data Visualizing and analysis:**
  1. Visualize data to help detect relevant relationships between variables.
  2. Split into training and evaluation sets

- **Training Model:**
  The goal of training is to make a prediction correctly as often as possible.

- **Evaluating the Model:**
  1. Uses some metric or combination of metrics to measure the performance of model.
  2. Shuffling the data and selecting 20/80 ratio for test/train data set.
  3. Hyper-parameter tuning, which is a corner stone for Model efficiency and performance improvement.
  4. Using test set data which have to predict the output.

1.3.2 Models to be developed

As this is a classification problem, we will try following models (more details about them is described in subsequent section):

- LightGBM Model
- CatBoost Model
- Random Forest Classifier
- Decision Tree Classifier
- K-neighbours Classifier

We will also try a **Benchmark Model** based on Logistic Regression.

1.3.3 Evaluation Metrics:

Our Problem is Classification Problem with imbalanced nature that will lead us to use the following Metrics:

Precision: The proportion of positive cases that were correctly identified.

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDR}$$

Recall: The proportion of actual positive cases which are correctly identified.

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

Below is the schematic, explaining confusion matrix and various ratios which can be derived from it

		True condition			
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
		False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

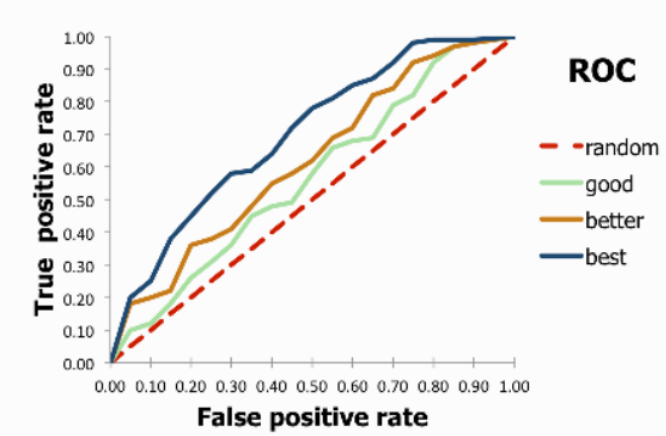
roc_auc_score : We will compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.

It should be noted that an ROC space is defined by FPR and TPR as *x* and *y* axes, respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). Since TPR is equivalent to sensitivity and FPR is equal to 1 − specificity, the ROC graph is sometimes called the sensitivity vs (1 − specificity) plot. Each prediction result or instance of a confusion matrix represents one point in the ROC space

The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives)

The diagonal divides the ROC space. Points above the diagonal represent good classification results (better than random); points below the line represent bad results (worse than random).

Below is the sample ROC curve:



1.3.4 Expectation from the models

We will compare the metric of each model with base model. It is expected that boosting based models will give better results and thus we expect improved performance as compared to the base model. However, to ensure that we are not overfitting our model, we will test their performance on test dataset.

Finally, we will recommend best model on the basis on precision, recall and roc-auc score.

2. Analysis:

2.1 Data Exploration:

2.1.1 Data Sets and Inputs:

Our Data consists of three data sets (three json files) , we will follow the hereunder process till reaching to our Modelled data which we will be used in our Model training and testing.

We have three JSON Files :
• portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)
• profile.json - demographic data for each customer
• transcript.json - records for transactions, offers received, offers viewed, and offers completed

portfolio.json: shape (10 rows x 6 columns)

- id (string) - offer id
- offer_type (string) - type of offer ie BOGO, discount, informational
- difficulty (int) - minimum required spend to complete an offer
- reward (int) - reward given for completing an offer
- duration (int) - time for offer to be open, in days
- channels (list of strings)

profile.json:shape (2175 rows x 5 columns) with 17000 unique users.

- age (int) - age of the customer
- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)

- id (str) - customer id
- income (float) - customer's income

transcript.json: (306534 rows x 4 columns)

- event (str) - record description (ie transaction, offer received, offer viewed, etc.)
- person (str) - customer id
- time (int) - time in hours since start of test. The data begins at time t=0
- value - (dict of strings) - either an offer id or transaction amount depending on the record

2.1.2 Data Sets Cleaning and reframing:

Profile Data Set:

1-Dividing the age Column to five age groups:

-Child : less than 18 years old.
-Teen :between 30 and 18 years old.
-Young adults : between 50 and 30 years old.
-Middle age adults : between 70 and 50 years old.
-Elderly : between 70 and 50 years old.

2-Transform the became_member_on Column to Month / year Format.

3-Claculating the Customer subscription cumulative number of days since the customer has been started his subscription.

4-Dropping the NA Values.

Transcript Data Set:

1-Dividing the value Column to offer id and amount columns.

2-changing the name of person Column to Customer Column.

Portfolio Data Set:

1-Dividing the Channels Column to Web, email , Mobile and Social media Columns .

2-changing the name of id Column to offer id Column to mange the merging between the data set .

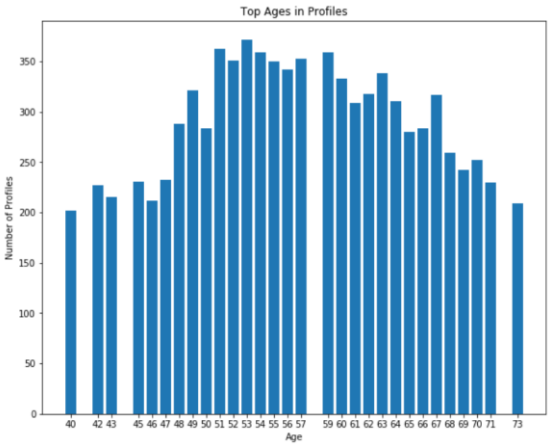
3-Dropping the Channels Column.

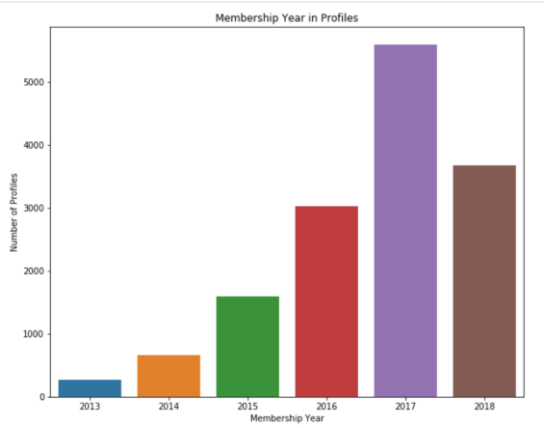
2.2 Exploratory Visualization:

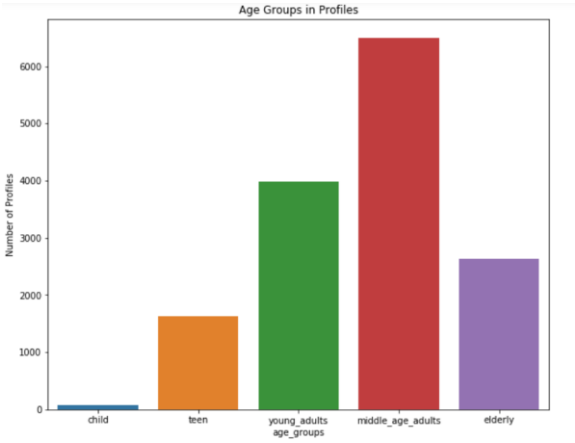
We will do data visualizing for the data sets before Combination and after combination, we will follow the below Process:

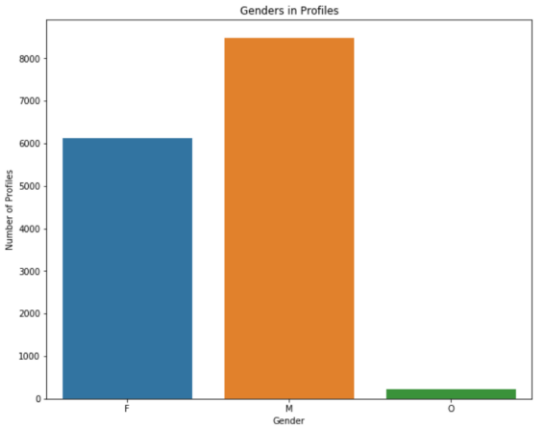
2.2.1 Exploratory Visualization before merging the Data sets:

2.2.1.1 Profile Data Set Exploratory Visualization:







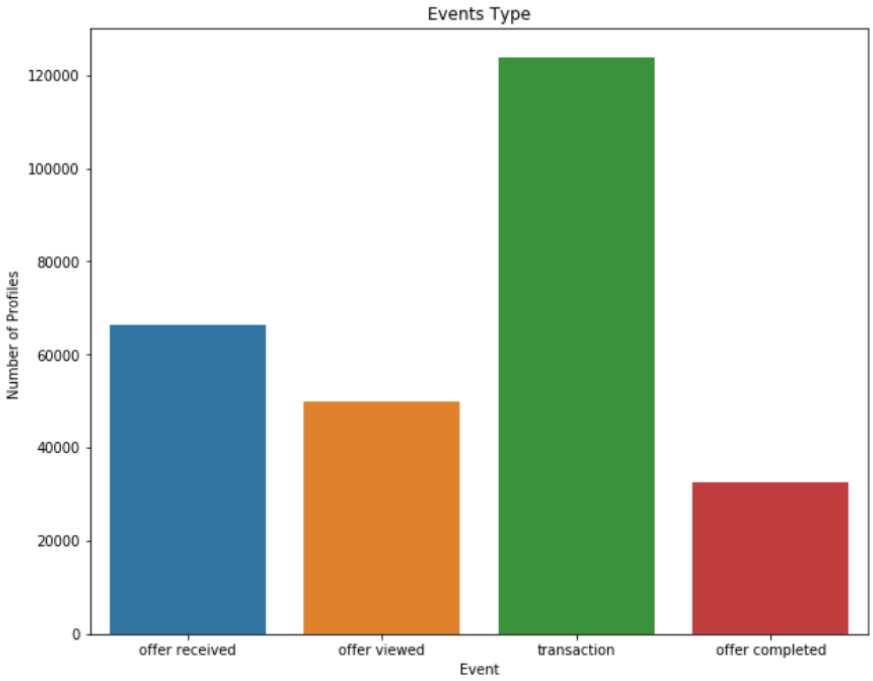


2.2.2 Exploratory Visualization after merging the Data sets:

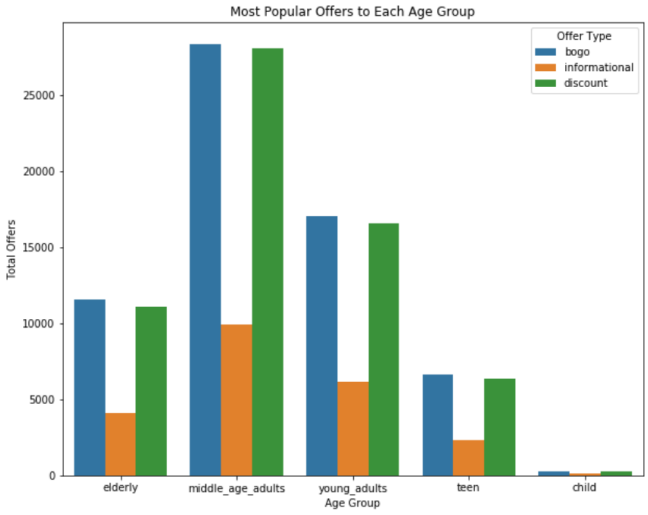
Now we will combine the three data sets, we will follow the below process, then we will do data visualization for the combined data set. We would at first combine transcript and profile into one data set (say, dataset 1). Then, we would combine portfolio and dataset 1into combined data set. Our output will be the below data frame:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 272762 entries, 0 to 272761
Data columns (total 20 columns):
event                     272762 non-null object
customer                  272762 non-null object
time                      272762 non-null int64
offer_id                  272762 non-null object
amount                    272762 non-null object
age                       272762 non-null int64
became_member_on          272762 non-null datetime64[ns]
gender                    272762 non-null object
income                    272762 non-null float64
age_groups                272762 non-null category
member_launch_Cum_days    272762 non-null int64
member_launch_year        272762 non-null int64
difficulty                148805 non-null float64
duration                  148805 non-null float64
offer_type                148805 non-null object
reward                    148805 non-null float64
web                       148805 non-null float64
email                     148805 non-null float64
mobile                    148805 non-null float64
social                    148805 non-null float64
dtypes: category(1), datetime64[ns](1), float64(8), int64(4), object(6)
memory usage: 41.9+ MB
```

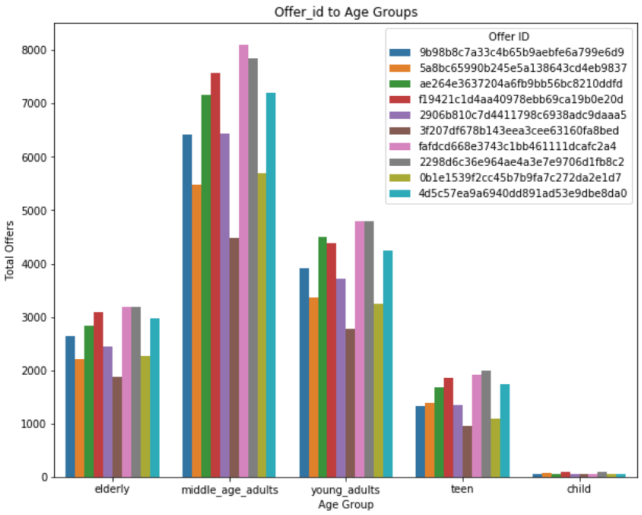
2.2.2.1 Exploratory Visualization for the Combined Data sets:



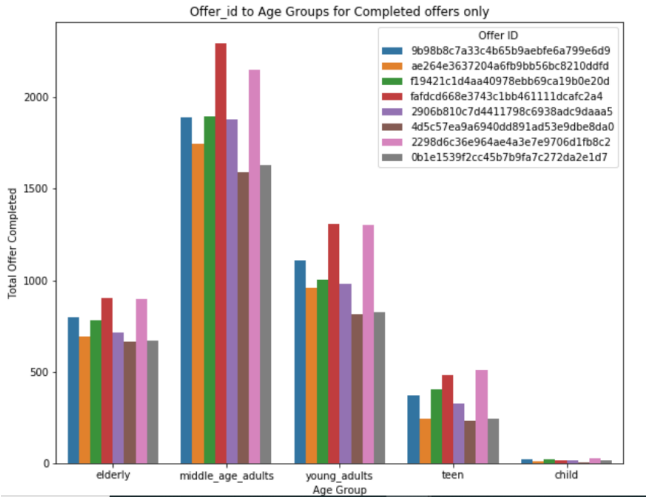
That above figure showing that the completed offers are more than 2,000 offers and less than 4,000, while the received offers are more than 6,000 and less than 8,000 offers , the completed offers after receiving and viewing are only our concern.



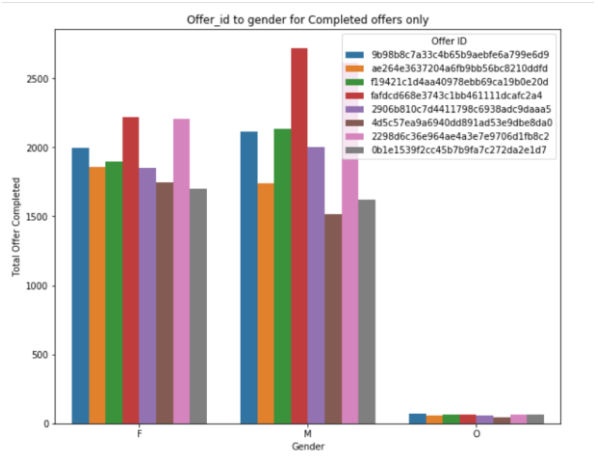
The above figure showing that middle age groups are mainly interested in bogo and Discount offer type.



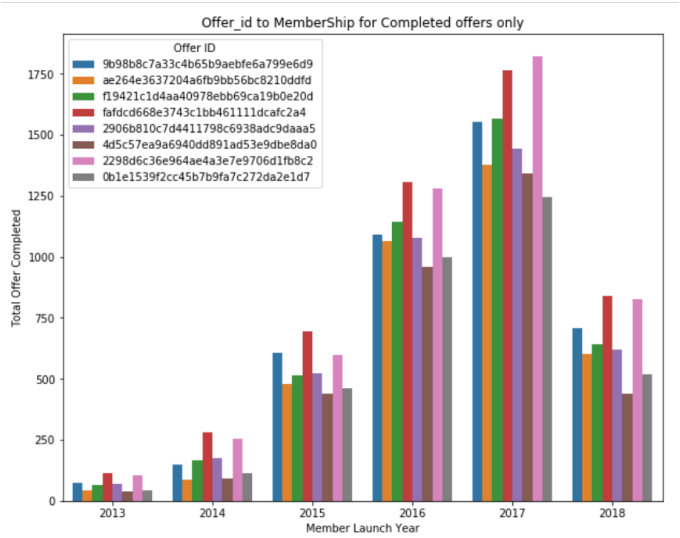
The above figure showing that middle age groups are the main portion whom are interested in offers.



As Shown in the above figure, the most of Completed offers come from middle age groups.



As Shown in the above figure, the most of completed offers come from Males .However the Females are interested in offers as well , and we don't have a big gap between Males and Females.



As shown in the above figure, the most of completed offers come from Customers whom membership have been started in 2017.

2.2.2.2 Statistics for Combined Data Set:

```
For Females:
Number of offer received: 27456 43.1% of total offers.
Number of offer viewed: 20786 32.6 % of total offers.
Number of offer completed: 15477 56.4 of received offers.

For Males:
Number of offer received: 38129 46.0 % of total offers.
Number of offer viewed: 28301 34.1 of total offers.
Number of offer completed: 16466 43.2 % of received offers.

The Maximum value to Complete offer for Females: 428.0 Hours and the Value by days is: 17.8 days
The Maximum value to Complete offer for Males: 434.0 Hours and the Value by days is: 18.1 days
```

2.3 Algorithms and Techniques:

As we are implementing a Classification Problem, we will implement the models in the following slides, and by comparing the results and our Evaluation metrics to our Benchmark model, we can know which is the best model to be implemented to our Problem.

Admittedly, we will concentrate on the Gradient Boosting Models like XGBoost, Cat Boost and LightGBM which Often provides predictive accuracy that cannot be beat , Lots of flexibility can optimize on different loss functions and provides several hyperparameters tuning options that make the function fit very flexible , No data pre-processing required - often works greatwith categorical and numerical values as is and Handles missing data .

2.3.1 Amazon Sage maker XG-Boost built in Algorithm:

XGBoost (extreme gradient boosting) is a popular and efficient open-source implementation of the gradient-boosted trees algorithm. ***Gradient boosting*** is a machine learning algorithm that attempts to accurately predict target variables by combining the estimates of a set of simpler, weaker models. By

applying gradient boosting to decision tree models in a highly scalable manner, XGBoost does remarkably well in machine learning competitions. It also robustly handles a variety of data types, relationships, and distributions. It provides a large number of hyperparameters—variables that can be tuned to improve model performance. This flexibility makes XGBoost a solid choice for various machine learning problems.

2.3.2 LightGBM Model:

LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel and GPU learning.
- Capable of handling large-scale data.

2.3.3 CatBoost Model:

CatBoost is a recently open-sourced machine learning algorithm from Yandex. It can easily integrate with deep learning frameworks like Google's TensorFlow and Apple's Core ML. It can work with diverse data types to help solve a wide range of problems that businesses face today. To top it up, it provides best-in-class accuracy.

It is especially powerful in two ways:

It yields state-of-the-art results without extensive data training typically required by other machine learning methods, and Provides powerful out-of-the-box support for the more descriptive data formats that accompany many business problems.

2.3.4 Random Forest Classifier:

Ensemble learning method for classification and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

2.3.5 Decision Tree Classifier

A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter

2.3.6 K-neighbours Classifier

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

2.4 Benchmark Model:

We will use Logistic Regression model as a Benchmark in which to compare our models's performance to, because it is fast and simple to implement. We will implement the roc_auc_score , Precision and Recall Metrics to Compare other Models 's Results.

3. METHODOLOGY

3.1 Data Pre-processing:

3.1.1 Combined Data Preparation for Models training and testing:

A)Dividing our Combined Data to three data sets :
1-received: extracting the items with event= offer received.
2-Viewed: extracting the items with event = offer viewed.
3-completed: extracting the items with event = offer completed.
4-transaction: extracting the items with event = transaction .

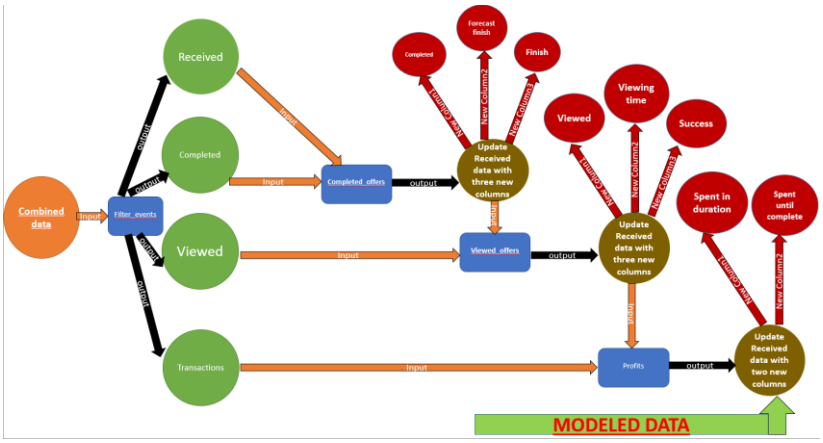
B)(1st output)extracting the persons who completes the received offers ,two new columns to be added to updated data set :
-(forecast_finish) column which equals to (received offer time + offer duration) .
-(finish) column which equals to (forecast_finish) value and received time value in case of the offer not completed or equals to completion time in case of offer completed.
-(completed) column which equals to (1) in case of offer completed and equals to (0) in case of offer not completed.

C)(2nd output) extracting the person who completed the received offer (1st ouput) after viewing the offer within the offer period , three columns to be added
-(success) Column which equals to (1) in case of offer completed after viewing the offer other wise equals to (0).
-(viewing_time) Column which equals to viewed offer time
-(Viewed) Column which equals to either (1) or (0).

D)(3rd output) profits calculation for the amount of money which is spent within the offer forecast completion time assuming that all transaction executed within the offer duration are using the offers.

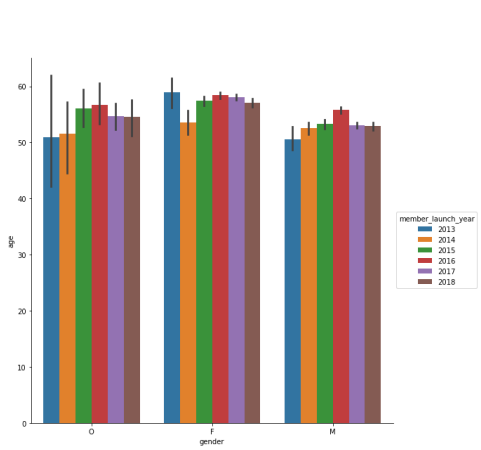
Eventually, we will get our Modelled data which will be used in our Models training and testing.

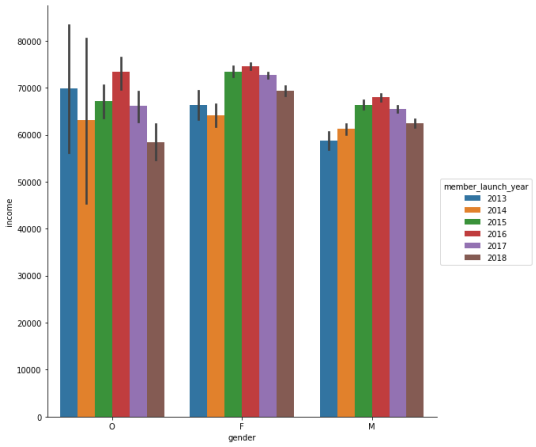
We will follow the below Process to get our modelled data starting from Combined data:

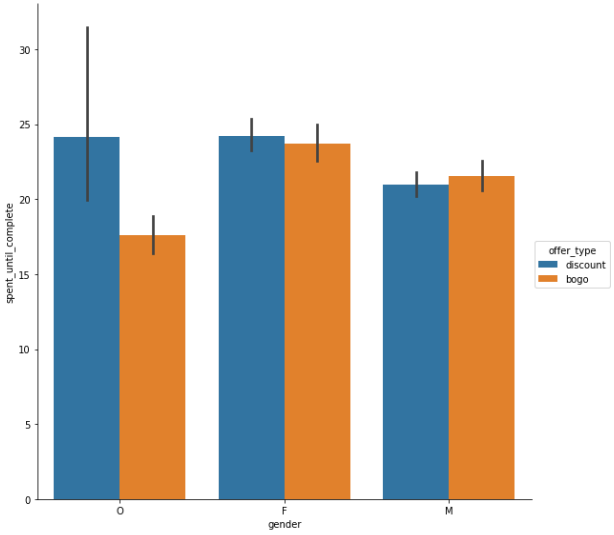


In the Modelled data we have a new column (‘’ success’’) , which will be our output label and will be equal to “1” in case of offer Completed after receiving and viewing or ” 0” otherwise.

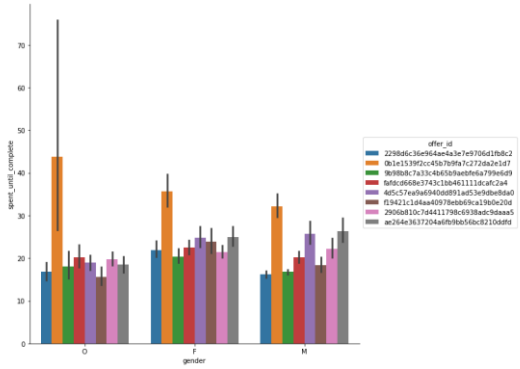
3.1.2 modelled data Exploration:

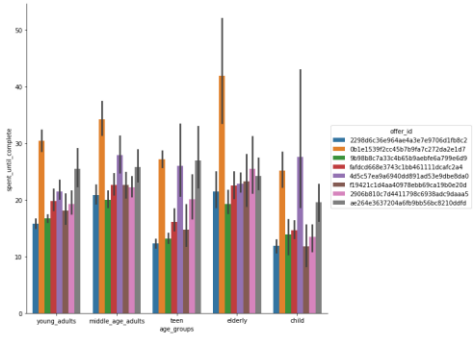


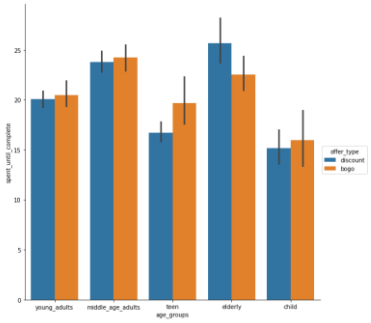




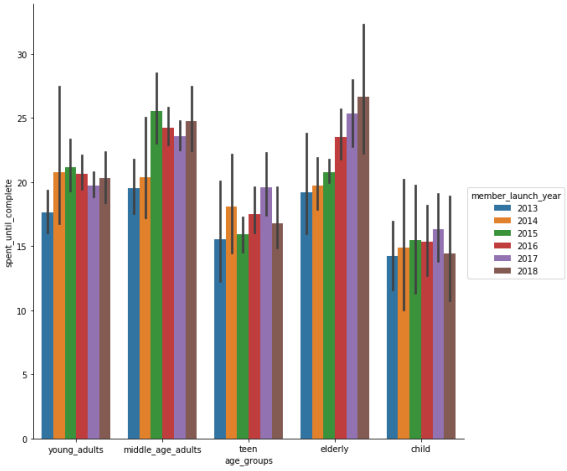
For the successful offers , the Females are more interested in discount type rather than bogo type , while Males are interested than bogo type rather than discount type.







The elderly age group are more interested in discount offer type an which they spent are more than bogo offer type.



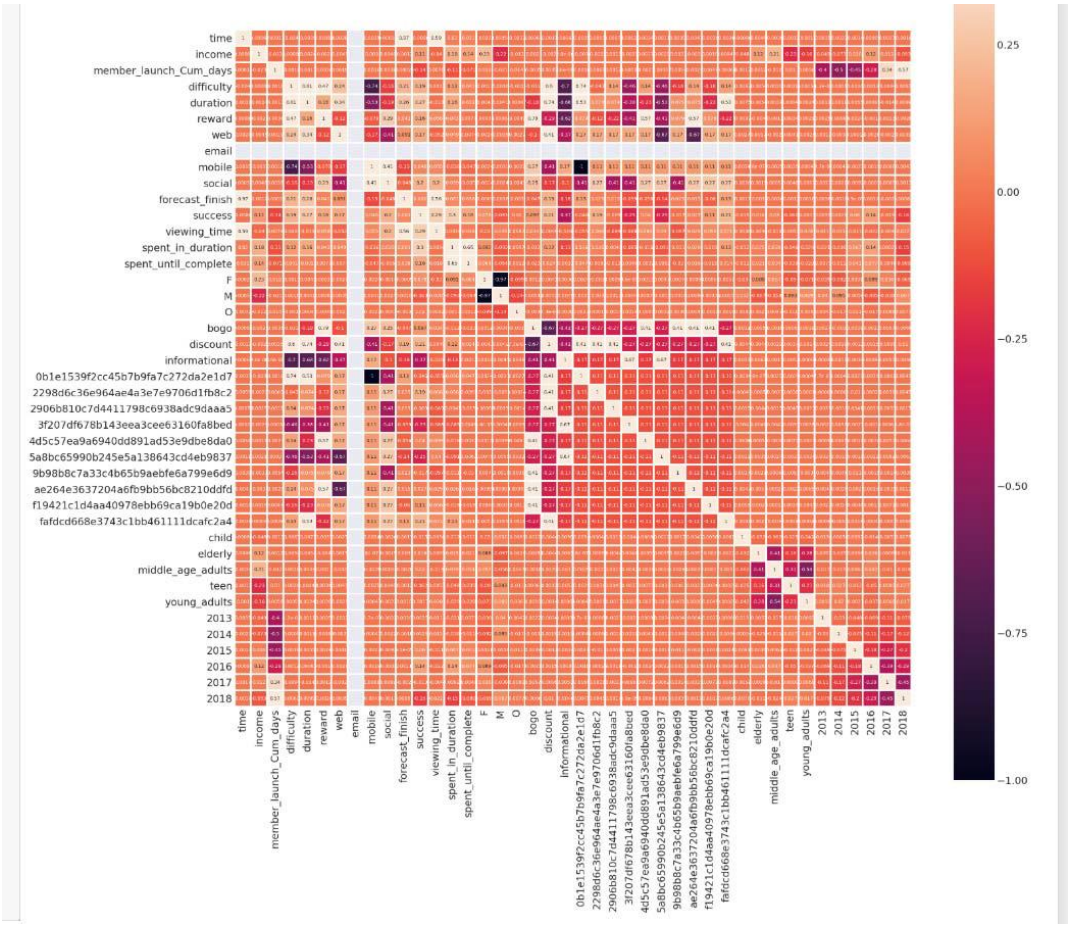
The customers with membership in 2018 are utilizing Discount offer rather than bogo offers

3.1.3 Modelled Data statistics:

For Females:
Number of offer Succeeded: 11107 offer, 40.5% of Female received offers.
Number of offer Succeeded: 11107 offer, 16.7% of Total received offers.
For Males:
Number of offer Succeeded: 12413 offer, 32.6% of Male received offers.
Number of offer Succeeded: 12413 offer, 18.7% of Total received offers.

For Females:
Total Spent until offer Complete: 266395 USD, 53.8% of Total Female received offers.
For Males:
Total Spent until offer Complete: 263321 USD, 50.6% of Total Male received offers.
For Females:
Total Spent until offer Complete: 266395 USD, 25.9% of Total received offers.
For Males:
Total Spent until offer Complete: 263321 USD, 25.6% of Total received offers.

The heat map for the modelled data:



As shown in the above heat map the best features for the modelled data which they have a good bond with our target ('success') are as below in ascending order:

```
#Correlation with output variable
cor_target = abs(C_mat["success"])
#Selecting highly correlated features
relevant_features = cor_target[cor_target>0.15].sort_values()
relevant_features
```

```
spent_until_complete                0.155922
2018                                0.158226
reward                              0.163495
web                                 0.167122
2298d6c36e964ae4a3e7e9706d1fb8c2    0.186246
difficulty                          0.190465
social                              0.197064
discount                            0.209410
fafdcd668e3743c1bb461111dcafc2a4    0.210728
5a8bc65990b245e5a138643cd4eb9837    0.249718
3f207df678b143eea3cee63160fa8bed    0.250011
duration                            0.265026
viewing_time                        0.291399
spent_in_duration                   0.299669
informational                       0.374796
success                             1.000000
```

3.1.4 Modelled Data Preparation for training and testing sets:

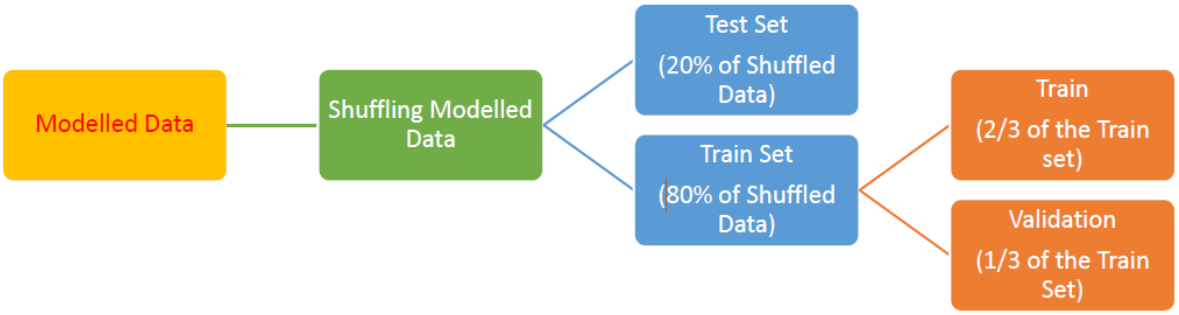
The Modelled data which will be utilized in our Model training and testing :shape (66501 rows x 42 columns)

- Input features : 41 Features
- Output Label: 1 Column (“success”) It will be either (“1”) or (“0”).

```
modeled_data.info(0)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66501 entries, 0 to 66500
Data columns (total 42 columns):
time                                66501 non-null float64
income                              66501 non-null float64
member_launch_Cum_days              66501 non-null float64
difficulty                          66501 non-null float64
duration                            66501 non-null float64
reward                              66501 non-null float64
web                                 66501 non-null float64
email                               66501 non-null float64
mobile                              66501 non-null float64
social                              66501 non-null float64
forecast_finish                     66501 non-null float64
success                             66501 non-null int64
viewing_time                        66501 non-null float64
spent_in_duration                   66501 non-null float64
spent_until_complete                66501 non-null float64
F                                   66501 non-null uint8
M                                   66501 non-null uint8
O                                   66501 non-null uint8
bogo                                66501 non-null uint8
discount                            66501 non-null uint8
informational                       66501 non-null uint8
0b1e1539f2cc45b7b9fa7c272da2e1d7    66501 non-null uint8
2298d6c36e964ae4a3e7e9706d1fb8c2    66501 non-null uint8
2906b810c7d4411798c6938adc9daaa5    66501 non-null uint8
3f207df678b143eea3cee63160fa8bed    66501 non-null uint8
4d5c57ea9a6940dd891ad53e9dbe8da0    66501 non-null uint8
5a8bc65990b245e5a138643cd4eb9837    66501 non-null uint8
9b98b8c7a33c4b65b9aebfe6a799e6d9    66501 non-null uint8
ae264e3637204a6fb9bb56bc8210ddfd    66501 non-null uint8
f19421c1d4aa40978ebb69ca19b0e20d    66501 non-null uint8
fafdcd668e3743c1bb461111dcafc2a4    66501 non-null uint8
child                               66501 non-null uint8
elderly                             66501 non-null uint8
middle_age_adults                   66501 non-null uint8
teen                                66501 non-null uint8
young_adults                        66501 non-null uint8
2013                                66501 non-null uint8
2014                                66501 non-null uint8
2015                                66501 non-null uint8
2016                                66501 non-null uint8
2017                                66501 non-null uint8
2018                                66501 non-null uint8
dtypes: float64(14), int64(1), uint8(27)
memory usage: 9.3 MB
```

We will follow the below process for training and testing sets preparation:



3.2 Implementation:

Firstly, after the Preparation of our training and testing data sets -We will implement our Benchmark model (Logistic regression Model) and calculating our Metrics that we have discussed before.

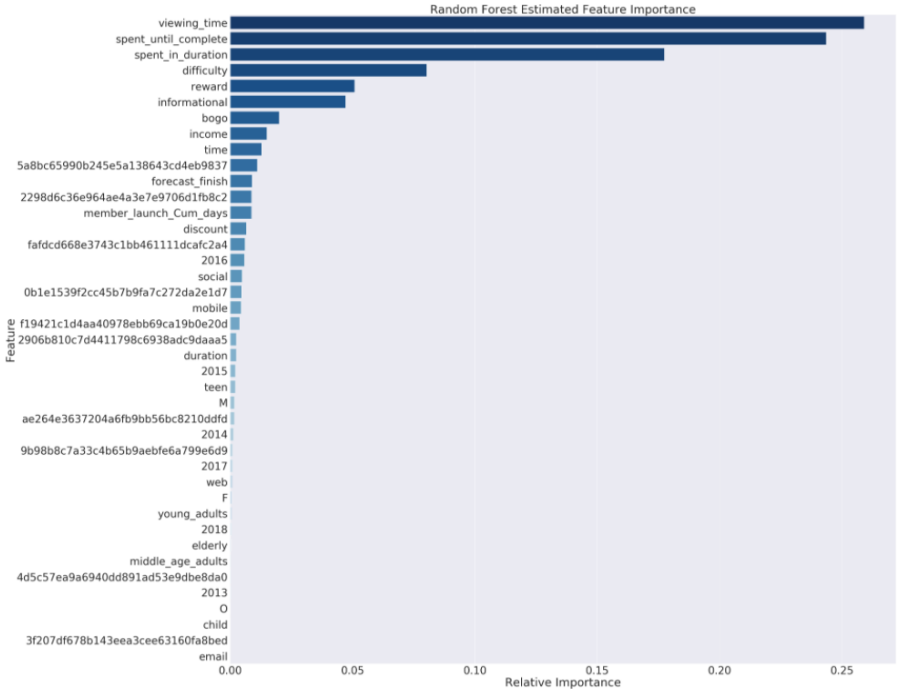
3.2.1 LOGISTIC REGRESSION MODEL (BENCHMARK MODEL):

```
('roc_auc_score:', 0.85317637023094)
('Precision Metric:', 0.8188451115280384)
('Recall Metric:', 0.8092295014421096)
```

3.2.2 Random Forrest Classifier:

```
('roc_auc_score:', 0.9702503342663181)
('Precision Metric:', 0.9492140266021766)
('Recall Metric:', 0.9703337453646477)
```

Feature Importance



3.2.3 Decision Tree Classifier:

```
('roc_auc_score:', 0.9617122693677033)
('Precision Metric:', 0.9550165837479271)
('Recall Metric:', 0.9491141326740832)
```

3.2.4 K-neighbors Classifier:

```
('roc_auc_score:', 0.7807964262392975)
('Precision Metric:', 0.7175152749490835)
('Recall Metric:', 0.7257931602801813)
```

3.3 Refinement:

We will work for improvement of our Models, and we will concentrate on XGB , LGB and CatBoost by tuning the hyper parameters.

3.3.1 Amazon Sage maker XG-Boost built in Algorithm with best parameter:

```
('roc_auc Metric:', 0.9723513549596361)
('AUC Metric:', 0.9723513549596361)
('Precision Metric:', 0.9643886372993001)
('Recall Metric:', 0.9651833539348991)
```

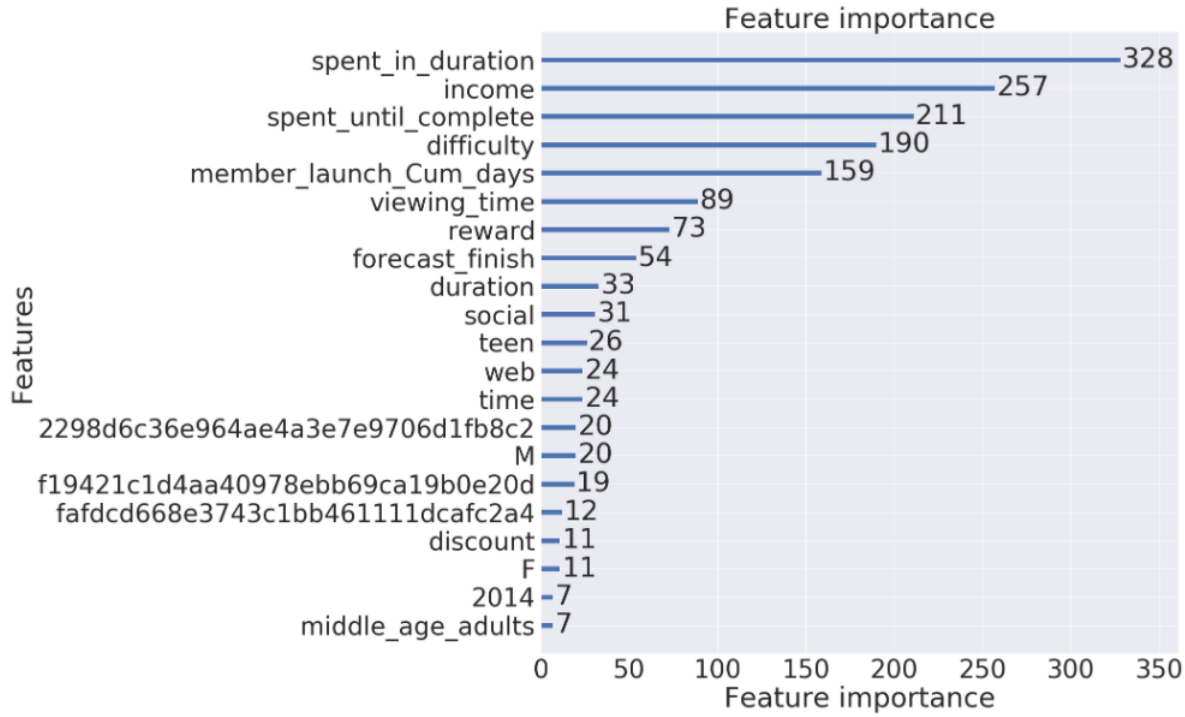
3.2.6 LightGBM Model (after hypertuning):

```
('roc_auc_score of Light GBM model:', 0.9709661209971148)
('Precision Metric:', 0.9631231973629996)
('Recall Metric:', 0.9631231973629996)
```

Hyper-parameters:

```
{'num_leaves': 35, 'colsample_bytree': 0.75, 'learning_rate': 0.1, 'n_estimators': 48, 'subsample': 0.5, 'random_state': 501,
'objective': 'binary', 'boosting_type': 'gbdt'}
0.9744697564807542
```

LightGBM Model(Features importance) :



3.2.7 Cat Boost Model (After hypertuning)

```
('roc_auc_score of CATBOOST model:', 0.9721322788804709)
('Precision Metric:', 0.9653465346534653)
('Recall Metric:', 0.9641532756489494)
```

Hyper parameters:

```
========================================================
 Results from Random Search
========================================================
('\n The best estimator across ALL searched params:\n', <catboost.core.CatBoostClassifier object at 0x7fb2e33728d0>)
('\n The best score across ALL searched params:\n', 0.9753114128605095)
('\n The best parameters across ALL searched params:\n', {'l2_leaf_reg': 7, 'learning_rate': 0.1, 'depth': 6})
```

4. Results:

4.1 Models Evaluation and Validation:

Now, after finalizing the models hyper parameters tuning and getting the best parameters, we will Evaluate the all Models together to get the best Model according to roc_auc_Score:

Classifier Type	True Positive	False Positive	False Negative	True Negative	Precision	Recall	roc_auc_score
Random_Forest_Classifier	4645	264	168	8224	0.946221	0.965095	0.965095
XGBoost	4653	174	160	8314	0.963953	0.966757	0.966757
CatBoost	4634	174	179	8314	0.963810	0.962809	0.962809
LGB	4633	182	180	8306	0.962201	0.962601	0.962601
Decision_Tree_Classifier	4550	232	263	8256	0.951485	0.945356	0.945356
Logistic_regression	3903	893	910	7595	0.813803	0.810929	0.810929
KNeighborsCLassifier	3527	1461	1286	7027	0.707097	0.732807	0.732807

As shown in the above figure, all classifier (except for KNeighborsClassifier) performs better than base model (i.e. Logistic_regression). We can also observe that XGBoost offers best roc-auc score.

4.2 Justification:

From the results shown in section 4.1, we can observe that boosting & bagging based models have better performance as compared to base model (Logistic_regression). This is as expected as these models train on the errors and thus give better results.

From the results, we would recommend XGBoost as the recommended model as it has least false positive and false negative. It should be noted that while training we have used cross validation in order to ensure that we don't overfit our model. Also since, above result are based on the test dataset (which was not used during model training), so we are confident that model will perform well in real scenario.

5. References :

- https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html
- https://aws.amazon.com/blogs/machine-learning/simplify-machine-learning-withxgboost-and-amazon-sagemaker/
- http://uc-r.github.io/gbm_regression
- https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/
- https://en.wikipedia.org/wiki/Receiver_operating_characteristic
- https://www.kaggle.com/arthurtok/introduction-to-ensembling-stacking-in-python
- https://stackoverflow.com/questions/10373660/converting-a-pandas-groupby-objectto-dataframe