Credit Card Lead Prediction

ML Approach Document

Submitted By: Raja Suman Chowdary C

Email ID: [rajasuman09@gmail.com](mailto:rajasuman09@gmail.com)

Table of Contents

[1. Problem Statement: 2](#_Toc73305059)

[2. Objective of the work 2](#_Toc73305060)

[3. Machine Learning Process Flow 2](#_Toc73305061)

[3.1 Understanding Problem Case 2](#_Toc73305062)

[3.2 Data Preparation 2](#_Toc73305063)

[3.3 Modelling 3](#_Toc73305064)

[4. Resources needed for the work 3](#_Toc73305065)

[a. Software Requirements 3](#_Toc73305066)

[**b.** Hardware Requirements 4](#_Toc73305067)

[5. Data Challenges & Risks 4](#_Toc73305068)

[5.1 Missing values 4](#_Toc73305069)

[5.2 Data Skewedness 4](#_Toc73305070)

[6 Data Preprocessing 5](#_Toc73305071)

[6.1 Missing Value Imputation 5](#_Toc73305072)

[6.2 Transforming Skewed Attributes to Gaussian 5](#_Toc73305073)

[7 Machine Learning Modelling & Techniques Applied 6](#_Toc73305074)

[7.1 Baseline Model Design 6](#_Toc73305075)

[7.1.1 Pre-processing Steps Applied: 6](#_Toc73305078)

[7.1.2 Data Splitting 6](#_Toc73305079)

[7.1.3 Feature Engineering 7](#_Toc73305080)

[7.1.4 Model Fitting 7](#_Toc73305081)

[8 Conclusion/Recommendations 8](#_Toc73305082)

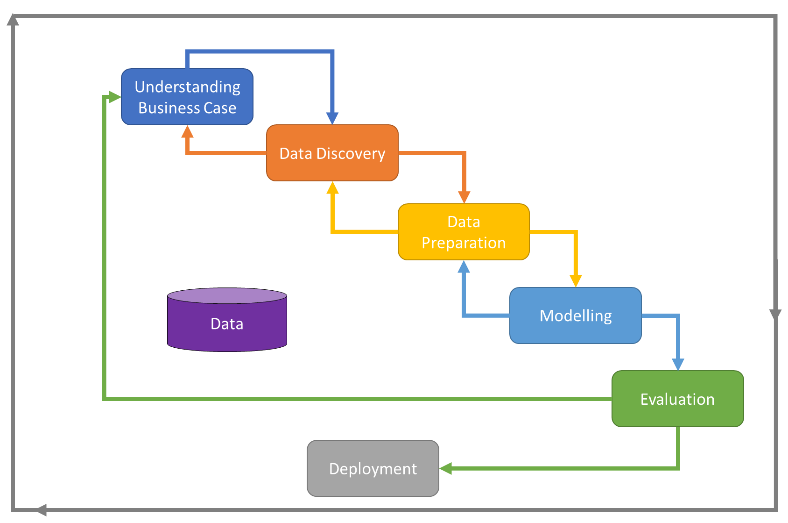
# Problem Statement:

**Happy** Customer Bank wants to cross sell its credit cards to its existing customers and wants to identify who among them could show higher intent towards a recommended credit card given the customer and their relationship details with the bank.

# Objective of the work

To utilize Machine Learning algorithms and help the bank in identifying the required customers who has a lead for credit card recommendation. As the output is to predict whether a customer has a lead or not, this can be considered as a binary classification problem and corresponding classification ML algorithms can be utilized and identify the lead.

# Machine Learning Process Flow

I have chosen to work in an iterative setting while loosely adopting a process model called CRISP-DM (Cross-industry Standard Process for Data Mining) for executing the current work. The process model describes a set of phases that would allow us to execute our work in an iterative manner by allowing bi-directional transition between the phases. This implies that when we reach the final stage of the process model we can start over again and/or refine the existing work.

Given in the figure here is the CRISP-DM process flow diagram

## Understanding Problem Case

The problem here is clearly mentioned in the hackathon which is to predict the Credit Card Lead. The success metric for this project has been chosen as ROC\_AUC and a target ROC\_AUC of > 0.85 can be considered to be a successful outcome for a predictive model which means to find the probability of lead for each customer which combined contributes to the target metric.

## Data Preparation

In this phase I have used the artefacts obtained from the previous phase to work on improving the quality of data.

I have specifically worked on the following areas and arrived at derivatives of the original dataset that could be used for developing different types of models in the subsequent phases:

1. Resolving Inappropriate Data
2. Missing Value Imputation
3. Dropping or encoding categorical values
4. Transform features with skewed distribution to Gaussian distribution
5. Feature derivation

Detailed work on above areas is explained in the relevant section further in the document

## Modelling

The objective of this phase is to build various models using the optimal data features identified as a part of the datasets prepared in the previous phase. I have divided this phase into 3 iterations as below:

1. Basic
2. With Feature Engineering Techniques
3. With Multiple Individual models applied
4. By Stacking various ensemble models

The datasets (raw and derived) obtained from the previous phase to work on creating models using the following machine learning algorithms:

* + Logistic Regression
  + Support Vector Classifier
  + Deep Neural Network
  + Tree based Classifiers
  + Ensemble Methods
  + AutoML

# Resources needed for the work

## Software Requirements

* Conda installation of Python 3.7 + with bundled packages
* Python Package Manager
* Jupyter Notebook (used as IDE)
* Google Colab/Kaggle platform (To setup a temporary 32 GB memory GPU backend for faster execution of models)
* Microsoft Office (For report preparations)

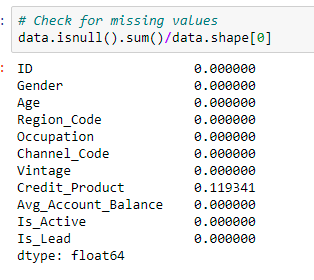
## Hardware Requirements

* A CPU with architecture of minimum 2 cores
* 8 GB RAM

# Data Challenges & Risks

## Missing values

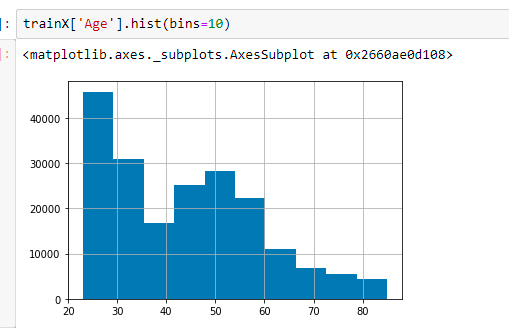
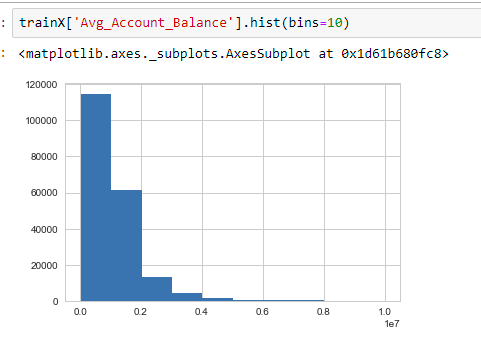
Every statistical package expects the data to be in certain format and conform to certain guidelines. If the data has missing values in it, some algorithms either drop those rows or treat them as zeros which can have a negative effect on the performance of the model. In our dataset there is 1 such feature with missing values in it. The following result of a code snippet depicts the quantity of missing values in that feature –



## Data Skewedness

Data sampling also plays an important role in building a good predictive model. Several underlying mathematical principles of statistical packages assume a Gaussian distribution of data and approximate the equations accordingly. Although the practical data might not follow a perfect Gaussian distribution all the time, even a nearly Gaussian distributed data helps several predictive algorithms especially parametric based algorithms to formulate an equation to predict more accurately.

Now, observing the distribution of features present in our input dataset, we have found some features which are skewed.



Sometimes during the process of data preparation, some features gets transformed to its square, exponential etc. and hence they require some kind of re-transformation again in order to reveal their actual distribution. Looking at the dataset, it is clearly evident that a couple of features are skewed as shown above and they might require certain transformation before we process them to the model.

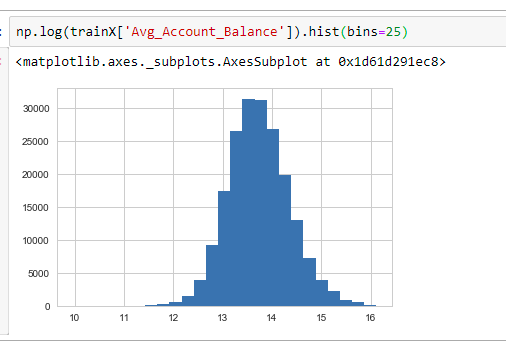
# Data Preprocessing

## Missing Value Imputation

For the feature ‘Channel\_Product’ which have got around 12% of the rows with missing values in it. I first tried with imputing with the mode of the column (value which is repeating most number of times). However, this degraded the performance by a little due to which I have made considered those missing values as a third category which actually improved the model performance.

## Transforming Skewed Attributes to Gaussian

Data skewedness in ‘Avg\_account\_Balance’ feature is removed by log transforming the values. For the remaining attributes ‘Age’ and ‘Vintage’, log transformation did not remove their skewedness and hence treated with encoding them and tried in the model as it is. The plot of this column after transformation looks like below:



# Machine Learning Modelling & Techniques Applied

Following sections explain in detail about execution of this work at each step along with the graphical results. Screen shots of source code are also provided wherever necessary. As mentioned in ‘Machine Learning Process Flow’ section above, my work involved establishing a baseline model with raw data and then building a refined model on top of it. Here, we present all the work performed at various levels of building a predictive model.

## Baseline Model Design

In this phase, I built the model with raw data as it is with only few essential data preparation steps in order to fit the data to the model.



### Pre-processing Steps Applied:

* Dropping ‘Id’ column
* imputation for missing values with a third category in ‘Credit\_Product’ feature

### Data Splitting

In order to train and test the model, data needs to be split into testing and training sets so that the model built will not have any bias towards the data and should be able to predict accurately on unseen data as well.

In order to achieve this, I have opted for a K-Fold Cross validation technique which basically splits the data into N parts, trains the model on N-1 parts and tests the model built with 1 part. This process is repeated for K times and mean of all the results is then calculated. This ensures that we don’t end up with over-fitted models

### Feature Engineering

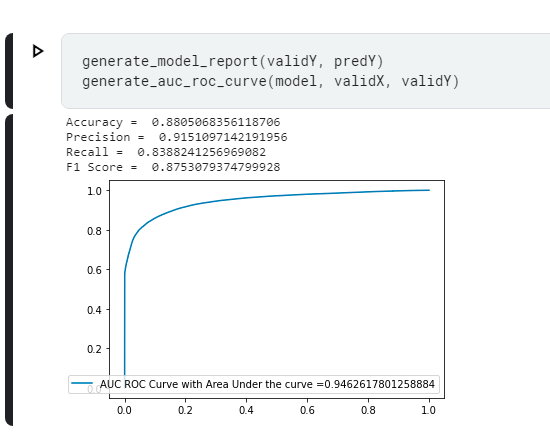
* Occupation Groups were assigned ordinal encoding instead of default based on the customer occupation profile
* Tried with creating bins for ‘Age’ attribute to reduce the variance in the data.
* Same for ‘Vintage’
* Regions were classified as 4 tiers based on the number of users in that area for that bank. Top 3 were the highest tiers and last tier combines the remaining regions

Tried the above in multiple combinations and removed those which did not improve any performance.

### Model Fitting

As my very first experimental model fitting, I have used the following classification methods to build various predictive models:

* Logistic Regression
* SGD Classifier
* K-NN
* Lasso Regression
* Tree Based Models

Among the above models, tree based models such as XGBoost, LGBM, HistogramGradientBoosting, CataBoost, ExtraTreeClassifier, RandomForestClassifier gave an ROC\_AUC score of 0.85.

Fine tuning these models with various hyperparameters and building a stacked model together with all the above tree models resulted in boosting the score to 0.865.

# Conclusion/Recommendations

Ensemble models have always shown superiority when it comes to classifying table based data and here they proved it again. There is still a scope for hyperparameter tuning of these ensemble models which I couldn’t do due to hardware (GPU) and time availability for this hackathon.

I have much more to do and explain on improving the score further using gradient boosting algorithms and AutoML if I get shortlisted.