Credit Card Lead Prediction

# Problem Statement:

Happy Customer Bank is a mid-sized private bank that deals in all kinds of banking products, like Savings accounts, Current accounts, investment products, credit products, among other offerings.

The bank also cross-sells products to its existing customers and to do so they use different kinds of communication like tele-calling, e-mails, recommendations on net banking, mobile banking, etc.

In this case, the Happy Customer Bank wants to cross sell its credit cards to its existing customers. The bank has identified a set of customers that are eligible for taking these credit cards.

Now, the bank is looking for your help in identifying customers that could show higher intent towards a recommended credit card, given:

Customer details (gender, age, region etc.)

Details of his/her relationship with the bank (Channel\_Code,Vintage, 'Avg\_Asset\_Value etc.)

**Data Dictionary**

\*ID : Unique Identifier for a row

\*Gender : Gender of the Customer

\*Age : Age of the Customer (in Years)

\*Region\_Code : Code of the Region for the customers

\*Occupation : Occupation Type for the customer

\*Channel\_Code : Acquisition Channel Code for the Customer (Encoded)

\*Vintage : Vintage for the Customer (In Months)

\*Credit\_Product : If the Customer has any active credit product (Home loan, Personal loan, Credit Card etc.)

\*Avg\_Account\_Balance : Average Account Balance for the Customer in last 12 Months

\*Is\_Active : If the Customer is Active in last 3 Months

\*Is\_Lead(Target) : If the Customer is interested for the Credit Card

0 : Customer is not interested

1 : Customer is interested

# Procedure Followed:

* Import Libraries
* Data Inspection
* Data Cleaning
* Exploratory Data Analysis
* Encoding Categorical Features
* Split Data to Independent and Dependent Variables
* Split Data to Train and Valid sets
* Scale the data
* Model Building with Classification Algorithms
* Evaluating the model’s performance
* Balanced the data and build the models
* Hyperparameter Tuning to achieve better Model

## Import Libraries:

To start with any Machine Learning project, we first import all required libraries and packages that perform required tasks.

## Data Inspection:

From the data provided to us

* We have 245725 rows with 11 columns of data in Train set
* We have 105312 rows with 10 columns of data in Test set

Data columns of Train Data (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 ID 245725 non-null object

1 Gender 245725 non-null object

2 Age 245725 non-null int64

3 Region\_Code 245725 non-null object

4 Occupation 245725 non-null object

5 Channel\_Code 245725 non-null object

6 Vintage 245725 non-null int64

7 Credit\_Product 216400 non-null object

8 Avg\_Account\_Balance 245725 non-null int64

9 Is\_Active 245725 non-null object

10 Is\_Lead 245725 non-null int64

------------------------------------------------------

Data columns of Test Data (total 10 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 ID 105312 non-null object

1 Gender 105312 non-null object

2 Age 105312 non-null int64

3 Region\_Code 105312 non-null object

4 Occupation 105312 non-null object

5 Channel\_Code 105312 non-null object

6 Vintage 105312 non-null int64

7 Credit\_Product 92790 non-null object

8 Avg\_Account\_Balance 105312 non-null int64

1. Is\_Active 105312 non-null object

* Of the 11 columns of train data, we have 3 Numerical features and 8 Categorical features. Though Is\_Lead has dtype of int64, we consider it as categorical feature as a binary
* Of the 10 columns of test data, we have 3 Numerical features and 7 Categorical features

## Data Cleaning:

### Handling Missing values:

Missing values may be due to multiple reasons like unrecorded observations or data corruption.

Handling missing data is important as many machine learning algorithms do not support data with missing values.

In our data Credit\_Product column is having missing values in both train and test data. There are 29325 missing values in train data and 12522 missing values in test data.

Methods tried to fill the missing values:

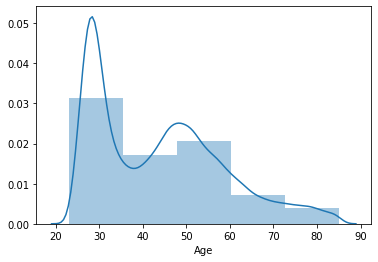
* Filled the missing values with Most Frequent datapoint (Mode) of Credit\_Product using function fillna()
* Filled the missing values with method of ‘ffill’ of function fillna()

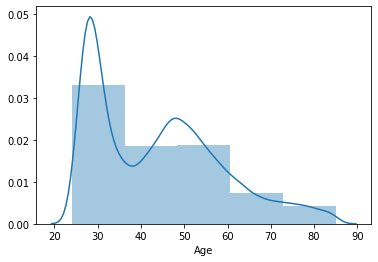
Tried both the methods and trained the model and observed that by using ‘ffill’ method of fillna() worked better.

## Exploratory Data Analysis:

**EDA** is often the first step of the data modelling process. It allows us to uncover patterns and insights, often with visual methods, within data.

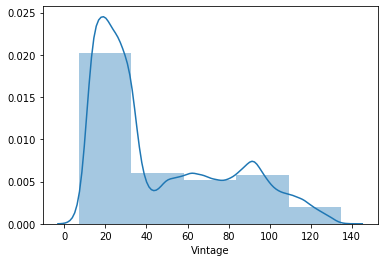
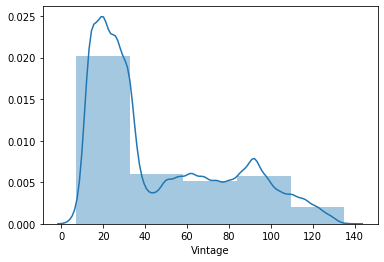
Lets check the Visualization of each column:

 **Age Distribution-Train data**  **Age Distribution-Test data**



We observe the customers that are eligible for taking the recommended credit cards were mostly in the age range of 20 to 35 years in both train and test data.

**Vintage distribution-Train data Vintage distribution-Test data**

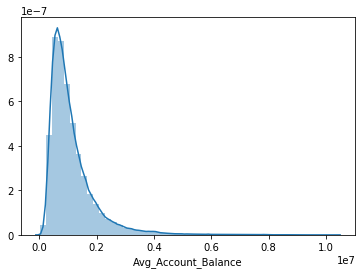
 

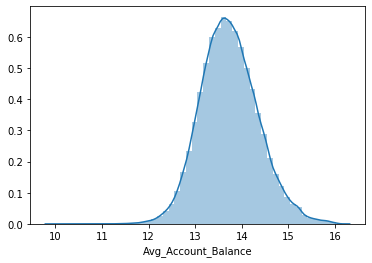
We observe that majority of customers selected as eligible for taking the recommended credit card has Vintage b/w 0-30 months in both train and test data

Here we observe that, customers have most of the Avg account balance b/w 0-200000 and it is positively skewed and same in both train and test data.

We try to normalise them using numpy log() function.

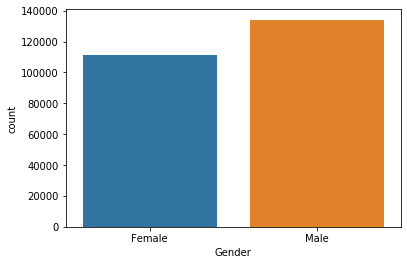
**Avg\_Acc\_Bal Dist- Train, Test data Normalized Avg\_Acc\_Bal Dist-Train,Test**



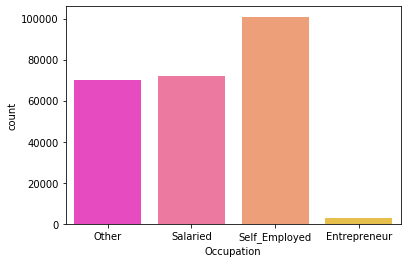


\*\*The countplot of all categorical features for train and test data are same.

**Gender Count**

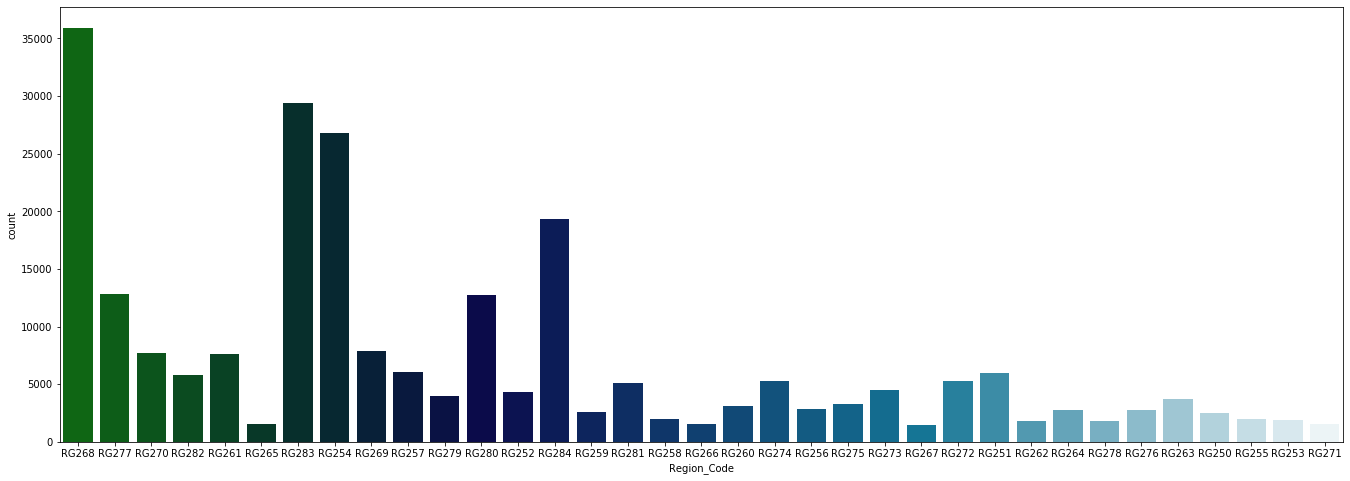


Here we can observe that among the customers selected by the Happy Customer Bank, Male customers are more compared to Female customers.

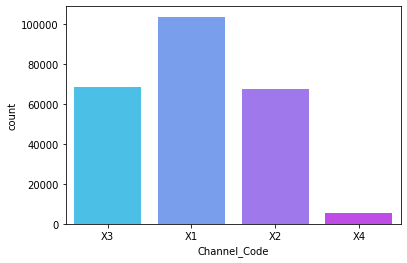
 **Occupation**

Here we can observe that Self\_employed customers are more and Entrepreneur customers are least in count in both train and test data

**Region\_Code**

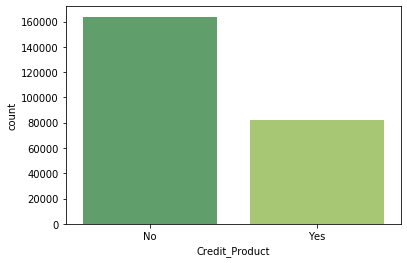


Customers selected by the bank are mostly from region with code of RG268 in both train and test data.

 **Channel\_Code**

Acquisition Channel\_code of X1 is more in count and X4 is in least in both train and test data.

**Credit\_Product**



We Observe that majority of Customer has NO active credit product in both train and test data.

**Is\_Active**



Majority of Customers are InActive in last 3 Months in both train and test data.

 **Is\_Lead**

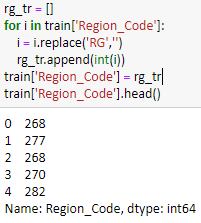
We can observe class-0 dominance in target variable Is\_Lead.

In this way we visualized our data and got some insights about the data. Now we proceed further by Encoding the categorical features.

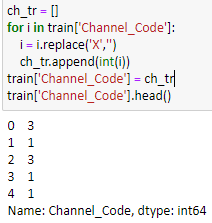
## Encoding Categorical Text Features:

Since we are working with Machine Learning Algorithms which uses mathematical techniques to train models, they don’t accept categorical text data. To overcome this issue, we Encode our text features of train and test data with various methods:

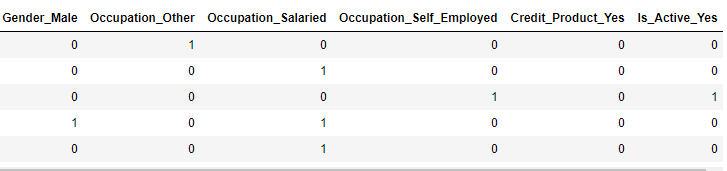
* Converted Regional\_Code to numerical data by replacing ‘RG’ with empty string ‘’ and took the integer form of each datapoint.



* In the same way, converted Channel\_Code to numerical data by replacing’X’ with empty string ‘’ and took the integer form of each datapoint.



* Rest of the categorical features were encoded by One Hot Encoding using Pandas get\_dummies(). The resultant is as follows:



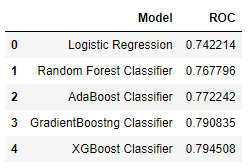
* Dropped ‘ID’ column in both train and test data.
* With the resultant train and test data, we keep test data aside for final evaluation and Split train data further to Independent and Dependent variables.
* The independent data is further split into train and valid sets as x\_train, x\_test, y\_train, y\_test.
* x\_train, x\_test are further scaled using StandardScaler().

## Model Building:

Here to train the model we used Classification algorithms of Logistic Regression, Random Forest Classifier, AdaBoost Classifier, GradientBoosting Classifer, XGBoost Classifer, Light GBM.

## Evaluation:

We were asked to use the metrics of **roc\_auc\_score** to evaluate the model performance. So, on predicting the probability of Customer showing interest (class 1) we got the roc scores as follows:

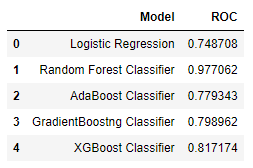


Here we can observe that XG Boost Classifier has got better roc\_auc\_score compared to other classifiers.

## Balancing the data:

Since our data has got Label imbalance (0 category is dominating 1 category), this may be one of the reason for less roc\_auc\_score. So, we balance our label by upsampling the class 1.

After up sampling the data, we again trained models as we had done before and evaluated the models and got the roc\_auc\_scores as follows:



We can now observe drastic improvement in roc\_auc\_score of Random Forest Classifier which is better compared to other models after balancing labels of the data.

Later we tried improving the performance of Random Forest Classifier and XGBoost Classifer by doing some Hyperparameter tuning and also build model with LIghtGBM and after evaluation results are as follows:

|  |  |
| --- | --- |
| **Model** | **Roc\_auc\_score** |
| Random Forest Classifier | 0.97722 |
| XGBoost Classifer | 0.81578 |
| Light GBM Classifier | 0.89744 |

Though we got better roc\_auc\_score for above models, after predicting on test data and submission of the resultant CSV file, we observed the overfitting issue with these models. So tried using CatBoost Classifer.

## CatBoost Classifer:

For Cat Boost Classifier, we need not encode the categorical data. It does by its own. So, we used the data of train and test before encoding the categorial features.

We trained model with CatBoost Classifer by making many changes and checked its roc score after submission:

* Tried dropping some columns assuming they may not give importance while predicting the target like Gender, Region\_Code, Channel\_Code. But the assumption was wrong. So continued with all columns of data.
* Did Hyper parameter tuning by tuning the number of iterations, learning rate and l2\_leaf\_reg
* After all trails, achieved roc\_auc\_score of 0.80727 and 0.800596 after submission.

## Conclusion:

Thus got the probability of 0.80727 and 0.800596 after submission as Customer showing interest (class 1).