Approach

1. Understand the Problem Statement and Import Packages and Datasets

Data Dictionary

Train Data

Variable	Definition
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

Now, **in order to predict whether the customer would be interested** in the Credit Card**,** we have **information** about **(gender, age, region code, Vintage, Avg_Account_Balance etc.) for each customer.**

Evaluation Metric used to Check Machine Learning Models Performance:

Here we have ROC_AUC_ScoreS as the Evaluation Metric*.*

The **Receiver Operator Characteristic (ROC)** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR (True Positive Rate)** against **FPR (False Positive Rate)** at various threshold values and essentially **separates the 'signal' from the 'noise'**. The **Area Under the Curve (AUC)** is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

- When **AUC = 1** the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.
- When **0.5<AUC<1** there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.
- When **AUC=0.5** then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.

- Train Data consists of 2,45,725 examples, and the Test Data consists of 1,05,312 examples.

Importing the required Python Packages

1. **Scientific and Data Manipulation –** Used to manipulate Numeric data using Numpy and Table data using Pandas.
2. **Data Visualization Libraries –** Matplotlib and Seaborn are used for visualization of the single or multiple variables.
3. **Data Preprocessing, Machine Learning, and Metrics Libraries –** Used to pre-process the data by encoding, scaling, and measure the date using evaluating metrics like ROC_AUC Score.
4. **Boosting Algorithms –** XGBoost Tree-based Classifier Models is used for Binary as well as multi-Class classification

2. Perform EDA (Exploratory Data Analysis) – Understanding the Datasets

Apply Head and Tail on Data – Used to view the Top 5 rows and Last 5 rows to get an overview of the data.

Apply Info on Data – Used to display information on Columns, Data Types and Memory usage of the DataFrames.

Apply Describe on Data – Used to display the Descriptive statistics like Count, Unique, Mean, Min, Max. etc on Numerical Columns.

Checking the Train Data for Duplicates – Removes the duplicate rows by keeping the first row. No duplicates were found in Train data.

3. Data Pre-processing:

The data type of feature Region_Code is object due to the prefix “'RG”.

So, removed the prefix and converted into numeric data type.

4. Fill/Impute Missing Values:

Categorical – Forward/Backfill

There is Missing Values in 1 Column “Credit_product”.

Apply ffill on Data – Used to forward fill that fills the current missing value with Previous Row value. If the previous row value is Nan (Not a Number) it moves to the next row without filling.

Apply bfill on Data – Used to Backward fill that fills the current missing value with Next Row Value. If the next row value is Nan (Not a Number) it moves to the next row without filling.

5. Split Train Data into Features (Independent) & Target (Dependent)

Split Train Data into Features and Target –Drop the Target column (Is_Lead) from the **DataFrame** to get the other features or independent variables.

6.1 Data Encoding: Label Encoding, OneHot Encoding:

One Hot Encoding will be applied only to Object or Categorical Columns.

Here, we have 5 categorical columns.

'Gender', 'Occupation', 'Channel_Code', 'Credit_Product', 'Is_Active'.

This means that categorical data must be converted to a numerical form. In One Hot Encoding the integer encoded variable is removed and a New Binary variable is added for each Unique label or category value

6.2 *Data Scaling: RobustScaler, StandardScaler, MinMaxScaler, MaxAbsScaler* :

Here we use RobustScaler

Outliers can often influence the sample mean and variance. RobustScaler which uses the median and the interquartile range often gives better results as it gave for this dataset.

7. Create Baseline Machine Learning Model for Binary Classification Problem

1. XGBoost

- **XGBoost (eXtreme Gradient Boosting)** is an implementation of gradient boosted **decision trees designed for speed and performance.**

- XGBoost is an algorithm that has recently been **dominating machine learning Kaggle competitions for tabular data.**

- **XGboost has an implementation that can produce high-performing model** trained on large amounts of data in a very short amount of time.

Key XGBoost Hyperparameter(s) Tuned:

1. 'max_depth':np.arange(2,6) –
   Select the optimum value to build the generalized the tree.

2. 'learning_rate':[0.15,0.1,0.01,0.001] -
   This controls the rate at which boosting learns.

8. Result Submission:

Finally, we make a **Result Submission by converting a DataFrame to a .csv file in the sample submission format** with columns “**Id**” and the predictions that we made using XGBClassifier is passed as values to “**Is_Lead** “.