Approach to Credit Card Loan Predictor – Analytics Vidhya JOB-A-THON - May 2021

I. Loading of Libraries & Datasets:

1. Required Python libraries have been loaded.
2. After loading the train and test datasets, the data has been inspected for shape, availability of missing values, first few records.

II. Data Wrangling:

3. It was found that the 11.93% (29,325) and 11.89% (12,522) of missing values are available in "Credit_Product" column of Train and Test data respectively.
4. Total number of categorical features and numerical features were identified in each of the train and test datasets.
5. Missing values in "Credit_Product" column of Train and Test data have been imputed with their corresponding mode values.
6. Label encoding has been done for all categorical variables of train and test data.
7. Standardization (Standard Scaler) applied on label encoded data.

III. EDA:

8. Under EDA, value_counts() of all categorical variables ('Gender', 'Region_Code', 'Occupation ', 'Channel_Code', 'Credit_Product', 'Is_Active', 'Is_Lead') have been viewed and found that there are no irregularities in the data of in these columns.

9. Bar plots have been made for the above categorical variables and following have been concluded:
   a) Male customers are more than Female customers.
   b) Self-employed customers are more. Entrepreneurs are very less.
   c) X1 channel code is highest. X4 is the least.
   d) There are more inactive customers than the active customers.
   e) There are less customers who can be lead for the credit card.

10. Also, pandas profiling has been done for all the variables. Also, correlation matrix and heatmap have been plotted among the numerical variables.

IV. Model Building & Results:

Sl No	Model	Data Wrangling	Results	Remarks
1	**Logistic Regression**	Missing values of 'Credit_Product' filled with it's mode value.	AUC of test data split from train data= **0.5309; Leaderboard Score =** 0.56539	Train : Test = 80:20
2	**Logistic Regression**	Missing values of 'Credit_Product' filled with it's mode value.	**AUC** of test data split from train data= **0.5; Leaderboard Score = 0.5**	**Train : Test = 70:30**

Sl No	Model	Data Wrangling	Results	Remarks
3	**XGBoost Classifier**	Missing values of 'Credit_Product' filled with it's mode value.	AUC of test data split from train data= 0.6153; Leaderboard Score = 0.6153	Train : Test = 80:20
4	Pycaret – compare models : **Light Gr Boost Machine Classifier**	Missing values of 'Credit_Product' filled with it's mode value.	AUC = **0.7886; Leaderboard Score** = 0.78959	Train : Test = 70:30 (default)
5	Tried different approaches like **PCA, feature selection** (selecting 'Age', 'Channel_Code', 'Vintage', 'Credit_Product' columns which have **higher correlation to the target variable** of 'Is_Lead') to improve the AUC score. **But, the score didn't improve**.			
6	Pycaret – compare models : **Light Gr Boost Machine Classifier**	Missing values of 'Credit_Product' filled with it's mode value.	AUC = **0.7885; Leaderboard Score** = 0.78959	Train : Test = 80:20
7	Pycaret – compare models : **Light Gr Boost Machine Classifier**	Missing values of 'Credit_Product' put under a separate category during label encoding.	AUC = **0.8735; Leaderboard Score** = **0.87246**	Train : Test = 80:20
8	Pycaret – compare models : **Light Gr Boost Machine Classifier**	1. Missing values of 'Credit_Product' put under a separate category during label encoding. 2. Standardization (Standard Scaler) applied on label encoded data.	AUC = **0.8735; Leaderboard Score** = **0.86802**	Train : Test = 80:20.

V. Final Submission:

The leaderboard score for approach followed in Sl No: 8 decreased compared to Sl No:7.

Hence, the **approach in Sl No: 7 has been used for final submission** since **it has the highest AUC of 0.8735** and leaderboard score of **0.87246.**

VI. Improvements / Future Scope:

The following could be done to improve the AUC score of the model:

1. Feature Engineering
2. Hyperparameter tuning