Report: Personal Loan Analytics

By Adarsh Pandey

Problem Statement Overview:

TVS has recently introduced Personal Loans (PL) services for customers facing urgent financial needs. However, the challenge lies in identifying customers with a higher likelihood of defaulting on Personal Loans, aiming to mitigate associated risks. The goal is to use Analytics and Data Science techniques to develop a Supervised Machine Learning model that provides acceptable accuracy in assessing the risk associated with potential Personal Loan applicants.

Documents and Deliverables:

1. Brief understanding of Case Study – 1 with Data Dictionary (PDF Format – Provided By TVS).
2. Labeled Dataset for Training (CSV File – Provided by TVS)
3. Code File - IPython Notebook (IPYNB Format – Code file involved in training and evaluating the Machine Learning Model)

Coding Platform details:

1. Language Used: Python 3.11.4
2. Platform: Jupyter Notebook

Approach:

1. Importing the Dataset and required libraries for Data Pre-Processing.
2. Data Pre-Processing and Feature Selection
   - Encoding Categorical features of the Dataset
   - Removing unwanted features (e.g., Customer ID, Pin Code, Date of Birth)
   - Eliminating features with more than 80% missing data
   - Evaluating correlation between Dependent (V30) and independent variables
   - Dropping features with low significance based on correlation
   - Updating Training data with Significant Features only
3. Defining dependent and Independent variables
4. Splitting the dataset into Train Dataset and Test Dataset (75% Training Data, 25% Test Data)
5. Feature scaling of independent variables to avoid errors during training of machine learning model.
6. Fitting the Pre-Processed dataset into classification models and evaluating the metric score (Accuracy) of each model:
   - Random Forest
   - Decision Tree
   - Gradient Boosting
   - Gaussian Naïve Bayes
   - Logistic Regression
7. The model with the best metric score (Accuracy) is used to assess whether the customer seeking a loan from TVS is risky (Bad/1) or not (Good/0).

Expected Benefit:

- The best metric score achieved is 97.87% accuracy using the Gradient Boosting Algorithm.
- The trained model can be deployed on Real-Time platforms to promptly identify defaulters and prevent the company from approving Personal Loans to such customers.

Algorithms Used:

1. For feature selection, correlation was used to identify significant features for training the machine learning model.Using all the features of a dataset for training purposes may be time consuming and may produce less accuracy. The features that were finally involved for training purposes based on correlation are:

V30	1.000000	Target variable (1: Bad / 0: Good Customer)
V26	0.142721	Number of times defaulted in last 6 months
V25	0.141551	Number of times defaulted in last 3 months
V27	0.139536	Number of times defaulted in last 12 months
V5	0.071633	Number of bounces with TVS Credit
V3	0.058623	Number of bounces in last 3 months Outside TVS
V19	0.027094	Number of Live loans
V24	0.021416	Number of enquiries
Employment_Type_SELF	0.020701	Employment type of customer SELF: Self-employed
Gender_MALE	0.015864	Gender: Male
V2	0.012339	First EMI Bounce (0: No, 1: Yes) (existing loan)
Employment_Type_HOUSEWIFE	-0.010992	Employment type of customer- HOUSEWIFE
V23	-0.011540	Number of closed loans
Gender_FEMALE	-0.015802	Gender: Female
Employment_Type_SAL	-0.016787	Employment type of customer SAL: Salaried,
V22	NaN	Number of new loans taken in last 3 months

2. Classification algorithms used:
   - Random Forest Algorithm (Advanced Ensemble Algorithm)
   - Decision Tree Algorithm (Primitive Ensemble Algorithm)
   - Gradient Boosting (Algorithm based on advanced ensemble techniques)
   - Gaussian Naïve Bayes (GNB) (Probabilistic Classifier Algorithm based on Bayes Theorem)
   - Logistic Regression (Basic algorithm for Binary Classification)

Final Evaluation Metric:

- Accuracy is calculated using Confusion Metrics.
- Gradient Boosting algorithm yielded the best results with an accuracy of 97.87%.

Accuracy = 100*(True Positive +True Negative)/Total

```
RandomForest : 0.9761060170002008
DecisionTree : 0.9759721571514625
GradientBoosting : 0.9777792651094305
GNB : 0.924569975235928
LogisticRegression : 0.9776119402985075
```

The highest accuracy is obtained using GRADIENT BOOSTING, which is 97.87%