Customer Transaction prediction

Problem Statement:

Prepare a complete data analysis report on the given data.

Create a predictive model which will help the bank to identify which customer will make transactions in future.

Domain Analysis:

Customer transaction prediction is used to predict whether the customer will make a transaction or not in feature. It is used in banking industry to identify potential customers.

Dataset of consist of 202 columns

1st column is ID_CODE, 2nd is target column and remaining 200 columns are anonymized features with column name from var_1 to var_200

1. Id_code: Unique identifier for each row or record in the data.

2. Target: 0 means the customer will not do a transaction and 1 means the customer will do a transaction.

Dataset Overview:

- The dataset provided no actual feature names — only anonymized numerical columns.
- Target variable: target (1 = transaction made, 0 = no transaction)
- Data shape: (e.g., 200,000 rows × 200 columns)
- Large and high-dimensional dataset with no semantic feature descriptions.

Initial Checks:

- Verified data shape and basic info.
- Checked for missing values → None found.

- No named features → traditional EDA (like pairplots, feature correlation, etc.) not possible.
- No outlier analysis due to lack of domain labels and interpretability.

Scaling:

- Applied StandardScaler to standardize the data for PCA and model input.

Dimensionality Reduction:

- Performed PCA to reduce dimensionality and avoid overfitting.
- Retained N components explaining ~95% variance.

Handling Imbalance:

- The dataset was imbalanced (fewer 1s than 0s).
- Used **SMOTE** (Synthetic Minority Oversampling Technique) to oversample minority class.

Model Building:

- Trained the dataset with multiple models like Logistic Regression, naïve bayes, random forest, decision tree , etc..

Comparing the models:

- Compared the models' performances with each other.

Challenges in Customer Transaction Prediction:

Lack of Feature Names

- No domain context → can't do meaningful EDA.
- Hard to interpret feature importance or explain model behavior.
- Limits insights that can be shared with stakeholders.

High Dimensionality

- Hundreds of features → increased risk of overfitting.
- Longer training time and higher memory usage.
- Need for dimensionality reduction (e.g., PCA), which itself removes interpretability.

No EDA:

- Without real feature names, you can't analyze patterns, distributions, or outliers meaningfully.
- No correlation heatmaps, feature-target trends, or histograms with insights.

No Hyperparameter Tuning:

- Due to large size (especially after SMOTE), tuning takes time and resources.
- Might miss out on a more optimized or better-performing model.

Model Evaluation

- Hard to trust accuracy alone — need precision, recall, F1, and AUC.
- Imbalanced data can cause misleading results (high accuracy but poor recall).

Overfitting Risk

- High feature count , imbalanced data , no tuning → increased risk.
- Must monitor test scores carefully and consider cross-validation.