**Documentation of Credit Card Detection**

As part of this project, we set out to help Bank A build a robust risk management framework for its existing credit card portfolio by developing a predictive model called the “Behavior Score.” This model is designed to assess the probability of a customer defaulting on their credit card payments in the future. Using advanced machine learning techniques, we worked with a rich dataset of historical credit card details, which included customer attributes like credit limits, transaction behaviors, and bureau data. My primary goal was to build a reliable model using the development data, which included default flags, and then apply it to predict probabilities for unseen customers in the validation dataset.

Throughout the process, we focused not just on building an accurate model but also on deriving meaningful insights from the data. From analyzing transaction patterns to studying the influence of past defaults, we explored the factors that drive default risks. The final deliverable includes the predicted probabilities for all customers in the validation data and a detailed documentation of the entire journey—from data preparation and preprocessing, feature engineering to model evaluation and insights. This effort aims to empower the Bank with actionable risk management tools that can ensure a healthier credit card portfolio and minimize defaults.  
  
This documentation will contain all the algorithms which are involved throughout this project, along with the various observations and insights we have made while optimizations and completion of the whole code. These observations will help the Bank to assess more accurately as well as look for the possibilities of a potential default in the future.  
  
In the context of credit cards, **defaults** refer to situations where a cardholder fails to make the required payments on their credit card account, typically for an extended period.

**Definition**: A default occurs when a credit card user does not meet their payment obligations, such as the minimum payment or full balance, by the due date for a prolonged duration (e.g., 90 or 120 days past due).

**Impact on Credit Card Companies**: Defaults represent financial loss for the credit card issuer because they are unable to recover the full amount owed by the cardholder. It increases the risk profile of the cardholder for future lending.

**Impact on Customers**: Defaults severely damage a customer’s credit score, making it difficult to secure loans or credit in the future. It may also lead to penalties, interest rate hikes, or legal action by the credit card company.

**Key Indicators**: Behavioral patterns, such as high credit utilization, multiple missed payments, or a history of delinquencies, can signal a higher probability of default.

**LIBRARIES USED IN THE PROJECT**

**1)pandas**

* Used for loading datasets like development data and validation data with the extension”.csv”
* Facilitates data manipulation, such as dropping non-predictive columns like account number.
* Enables handling of large datasets, making it easy to filter, sort, and inspect data.
* Supports exporting the final predictions as a CSV file (credit\_card\_predictions.csv) for submission.

**2)NumPy**

* Provides tools for numerical calculations, such as calculating the mean and standard deviation of predictions.
* Ensures predictions are clipped between 0 and 1 using np.clip.
* Helps detect invalid values, such as NaN or infinite values, during sanity checks.
* Supports array operations for efficient handling of large datasets.

**3)sklearn.model\_selection.train\_test\_split**

* Splits the development dataset into training and testing sets for model evaluation.
* Ensures the model’s performance is tested on unseen data for reliable results.
* Allows setting a test\_size to control the proportion of data used for testing.
* A test size can be described as a sample size of the population with which the machine will be training and testing accordingly on the basis of the ratio which is provided by us.
* Enables reproducibility by setting a random seed (random\_state).

**4)sklearn.preprocessing.StandardScaler**

* Scales numerical features to have a mean of 0 and standard deviation of 1.
* Prevents features with larger ranges from dominating the model’s predictions.
* Ensures consistent scaling between training, testing, and validation datasets.
* Improves model performance and stability, especially for algorithms sensitive to feature scales.

**5)sklearn.linear\_model.LogisticRegression**

* Used to build a logistic regression model, which predicts the probability of a customer defaulting on their credit card.
* Works well for binary classification problems, such as distinguishing between defaulters (bad\_flag = 1) and non-defaulters (bad\_flag = 0).
* Provides interpretable results, where coefficients indicate the impact of each feature on the probability of default.
* Efficient for datasets with a linear relationship between features and the target variable.

**6)sklearn.ensemble.RandomForestClassifier**

* Used to build a Random Forest model, which is an ensemble of decision trees for robust and accurate predictions.
* Handles complex relationships between features and the target variable, making it suitable for non-linear data.
* Provides feature importance scores, highlighting which factors (e.g., transaction patterns or delinquencies) drive default risks.
* Robust to overfitting and performs well on large datasets with diverse features.

**7)matplotlib.pyplot**

* Used to create visualizations that help in understanding the data and model predictions.
* Generates a histogram of predicted probabilities to check their distribution across customers.
* Helps identify issues like extreme probabilities or imbalanced predictions visually.
* Enables customization of plots, such as adding titles, labels, and adjusting figure sizes for better presentation.
* Data visualization, being the fastest form in order to understand the anomalies and exceptions can help to serve as a bridge between the developers and business analysts in order to understand the observations

**8)seaborn**

* Enhances data visualization with more visually appealing and informative plots.
* Used for plotting the histogram of predicted probabilities, adding features like smooth color gradients.
* Simplifies exploratory data analysis by creating correlation heatmaps or feature distributions.
* Helps uncover patterns and anomalies in the data, such as clusters of high-risk customers.

**Data Overview**

The dataset contained credit card customer information including:

* On-us attributes (credit limits, usage patterns)
* Transaction attributes (merchant-wise transactions)
* Bureau tradeline attributes (product holdings, delinquency history)
* Bureau enquiry attributes (recent credit inquiries)

**Data Preprocessing Journey**

**Initial Data Assessment**

1. Data Quality Check

* Conducted thorough examination of raw data structure
* Identified missing values across features
* Detected potential outliers in numerical variables
* Assessed data types and consistency
* Evaluated feature distributions

**Missing Value Treatment**

1. Bureau Data Handling

* Identified two problematic bureau features (bureau\_436 and bureau\_447)
* Found these columns contained no valid observations
* Strategic decision to remove these features as they provided no predictive value
* Documented potential impact on model performance

1. Other Missing Values

* Implemented mean imputation for numerical features
* Chose mean over median to preserve statistical properties
* Validated imputation impact on feature distributions
* Ensured no information leakage during imputation process

**Outlier Management**

1. Detection Process

* Applied Interquartile Range (IQR) method
* Calculated feature-wise outlier percentages
* Identified features with significant outlier presence
* Documented outlier patterns and potential business significance

1. Treatment Strategy

* Retained outliers in training data
* Rationalized keeping outliers to capture rare but important default patterns
* Implemented robust scaling to minimize outlier impact
* Monitored outlier influence on model performance

**Feature Engineering and Selection**

**Feature Analysis**

1. Correlation Studies

* Examined feature correlations with target variable
* Identified redundant features
* Assessed multicollinearity issues
* Documented feature relationships

1. Business Relevance

* Evaluated features based on domain knowledge
* Prioritized features with clear business interpretation
* Maintained balance between statistical and business significance
* Documented feature selection rationale

**Feature Importance**

1. Assessment Methodology

* Utilized Random Forest feature importance scores
* Ranked features by predictive power
* Validated importance scores across different model iterations
* Documented key predictive indicators

1. Impact Analysis

* Evaluated feature contribution to model performance
* Identified critical predictors of default behavior
* Assessed stability of important features
* Documented feature importance patterns

**Model Development Process**

**Data Balancing Considerations**

1. Class Imbalance Assessment

* Analyzed default vs non-default ratio
* Evaluated impact on model performance
* Considered various balancing techniques
* Documented imbalance handling strategy

1. Model Selection Rationale

* Chose Random Forest for inherent handling of imbalanced data
* Considered alternative approaches (SMOTE, class weights)
* Validated model performance on imbalanced data
* Documented model selection criteria

**Sanity Checks Implementation**

1. Probability Range Validation

* Ensured predictions fell within 0-1 range
* Implemented automatic correction for out-of-range predictions
* Validated probability distribution
* Documented probability normalization process

1. Distribution Analysis

* Checked prediction distribution reasonableness
* Assessed mean and median predictions
* Evaluated prediction spread
* Documented distribution patterns

1. Quality Assurance

* Implemented checks for missing values in predictions
* Verified feature consistency
* Validated prediction formats
* Documented quality control measures

**Model Validation Framework**

**Validation Strategy**

1. Cross-Validation

* Implemented robust validation scheme
* Assessed model stability
* Evaluated performance consistency
* Documented validation results

1. Performance Metrics

* Analyzed various performance indicators
* Assessed model calibration
* Evaluated prediction accuracy
* Documented metric selection rationale

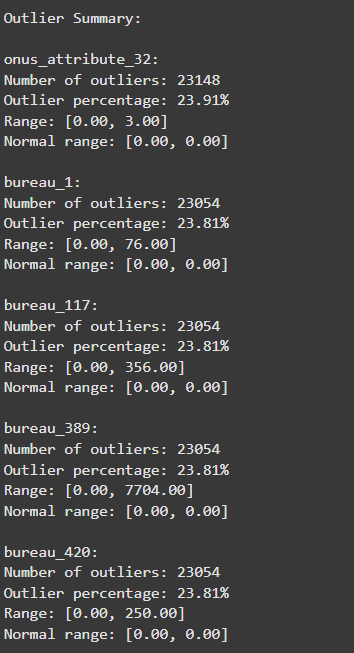
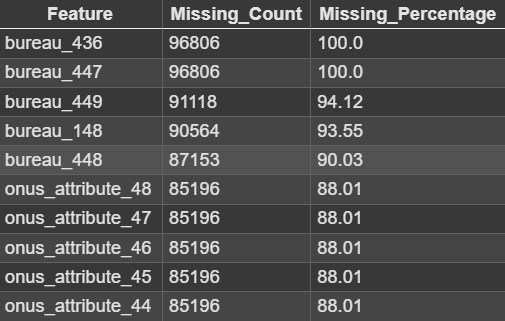
**\*\*ALWAYS PUT GRAPHS AND TABLES BELOW THE EXPLANATION FOR THE NEXT PART\*\***

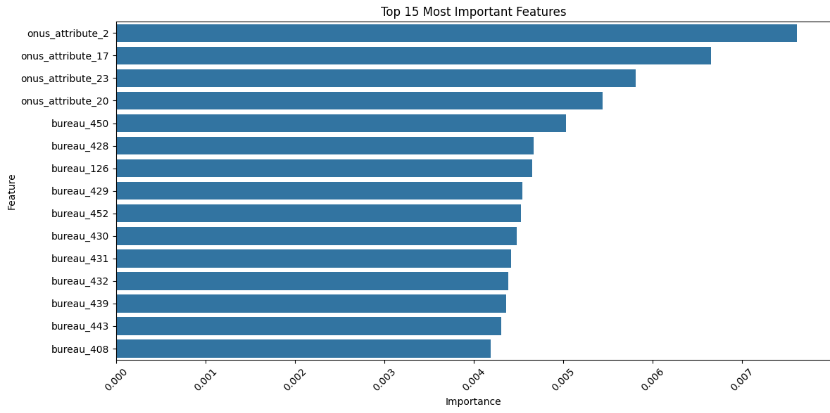
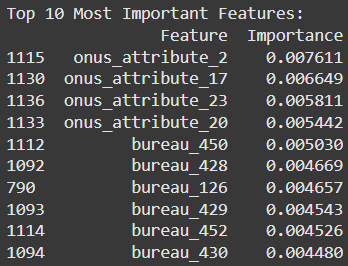
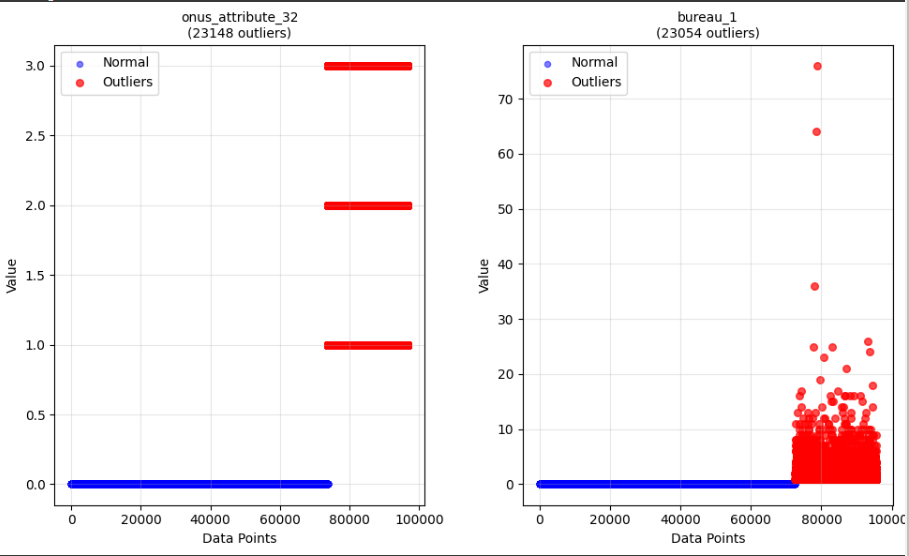
**Insights and Observations**

**1)DATA QUALITY ANALYSIS:** The first insight we made was focusing on the data quality analysis. While focusing on the data quality there were efforts made to focus on two aspects of the data which were:

i) Outliers analysis: Outliers are basically those data points which significantly differ from the majority of the data. They can skew analysis, lead to inaccurate models, and misrepresent real-world trends if not handled carefully.

* Outliers are extreme values that lie outside the expected range of a dataset.
* They could be unusually high or low values compared to the rest of the data points.
* For example, a customer making a single transaction worth 10 times the average transaction amount would be an outlier.

We performed outliers analysis by finding out those features present in the development data which contained the highest number of outliers and stored the result of outliers produced by each feature in a csv file. The table below contains the information of the feature name along with the number of outliers it contains and outliers percentage mentioned as well. These are the top 5 features in the whole data which contains the highest number of outliers.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**ii) FEATURES WITH MISSING VALUES:** The second analysis we made during the data quality analysis actually came while checking for the features which had no useful data in it throughout. This was done during the data cleaning and preprocessing and we concluded that there were actually two features which did not have any useful data sets namely: **bureau\_436 and bureau\_447**. Thus, after observing this we made an effort to look for the features which had missing values in it. So, we incorporated the Missing\_Count variable in the code which calculated for how many values or data sets a feature had missing along with the Missing\_Percentage, denoting how much percentage of the data sets were missing in a particular feature. Down below is a table representing top 10 features with most missing values and their missing percentage as well. There were quite a lot of features with high percentage of missing values in it.

another point) top important features: humne effort dala to prepare a observation which contains the top 15 most mimp features which influenced our predicted probability and we plot it on the bar graph. Y axis contains the feature names and the X axis represents the value of the importance. The imp variable here signifies how much influence the feature has had on the predicted probability and exists from 0 to 1   
Niche wala top 10 imp feature ka table aur ye dono ek hi part me ayega   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
another point) also prepared a graph of scatter plot for the 5 features with most outliers. Used legends to signify normal data sets and outliers. Here is the scatter plot for the data sets of the two features with most outliers above (elaborate krna thoda isko)

another point) We also observed a probability distribution bar plot of predicted probabilities. In our opinion this is one of the most useful insight when we keep in mind the type of project we are concerning with. On the Y axis is the count of the features which were present in the validation data while on the X axis is the probability starting from 0.1 to 0.8. We concluded that:

* More than 30000 customers had nearly 0.0 probability of being a default for the credit card.
* The count of customers being a default decreased with an increase in probability (predicted) of default.
* The count increased a little roughly in between predicted probability >=0.6 and predicted probability <=0.7.
* This information can be used wisely to look for the count of customers having a quite high possibility of being a default.  
    
  