Week 6 Documentation: Credit Scoring Model for Bati Bank

---

1. Project Overview

This project focuses on building a **Credit Scoring Model** for **Bati Bank**, which is collaborating with an eCommerce company to provide **Buy Now Pay Later** services. The goal is to assess the creditworthiness of potential customers by predicting whether they are likely to default on their payments.

The model will classify customers into two categories:

- **High Risk (Bad)**: Likely to default.
- **Low Risk (Good)**: Likely to repay without default.

To achieve this, data from the eCommerce platform is used to create a **proxy default variable**, select relevant features, and develop machine learning models that assign risk probabilities and credit scores. The final objective is to create a real-time prediction API for serving the model in production.

---

2. Task Breakdown

Task 1: Understanding Credit Risk

Objective:

To gain an understanding of **credit risk** and its role in assessing customer creditworthiness, as well as review relevant literature on methods and approaches for building a credit scoring model.

Key Concepts:

- **Credit Risk**: The likelihood that a borrower will default on a loan.
- **Basel II Capital Accord**: Regulatory framework that defines the parameters for measuring credit risk.
- **Credit Scoring Models**: Techniques to evaluate and predict default probability, which can include statistical and machine learning approaches.

Progress:

- **Research Completed**:

- Explored key references about credit scoring techniques, the Basel II framework, and alternative methods.
- Gained insights into the importance of features such as transaction history, loan repayment patterns, and customer demographics.

Next Steps:

- Apply this understanding to define the proxy variable for default classification in the dataset.

Task 2: Exploratory Data Analysis (EDA)

Objective:

To analyze the provided dataset, identify patterns, and determine which features are useful for modeling. This includes inspecting the structure of the dataset, checking for missing values, analyzing distributions, and exploring relationships between variables.

Key Steps Taken:

1. **Dataset Overview**:
    - **Shape and Data Types**: Analyzed the number of rows, columns, and data types for each feature.
    - **Summary Statistics**: Calculated basic statistical measures for numerical features (e.g., mean, standard deviation, min, max) and reviewed categorical feature distributions.
2. **Visualization**:
    - Plotted **histograms** for numerical features to understand their distribution.
    - Created **count plots** for categorical features to observe their frequency.
    - Generated a **heatmap** to visualize correlations between numerical variables.
3. **Missing Values**:
    - Identified any missing data within the dataset and outlined strategies to handle them (e.g., imputation or removal).
4. **Outlier Detection**:
    - Used **box plots** to identify potential outliers in numerical features.

Insights:

- Some numerical features, such as `Amount` and `Value`, have a wide range of values, indicating possible outliers.
- Certain categorical features (e.g., `ChannelId`) show imbalances that may need to be addressed through resampling techniques.
- **Correlation**: The heatmap revealed relationships between key features like `Amount`, `Value`, and `FraudResult`, which could be important for predicting default.

Next Steps:

- Proceed with **feature engineering** to generate new, informative features that can improve model performance.

---

3. Data Cleaning and Feature Engineering

Key Steps:

1. **Create Aggregate Features**:
    - **Total Transaction Amount**: The sum of all transaction amounts for each customer.
    - **Average Transaction Amount**: The mean transaction amount per customer.
    - **Transaction Count**: The total number of transactions made by each customer.
2. **Extract Temporal Features**:
    - **Transaction Hour**: The hour of the day when the transaction occurred.
    - **Transaction Day/Month/Year**: Temporal breakdown of when transactions took place.
3. **Encode Categorical Variables**:
    - Use techniques such as **One-Hot Encoding** and **Label Encoding** to convert categorical variables into numerical values.
4. **Handle Missing Values**:
    - Impute missing values using strategies like mean, median, or mode imputation, or consider removing rows/columns with missing data.
5. **Standardize Numerical Features**:
    - Apply **Normalization** (scaling values between 0 and 1) or **Standardization** (scaling to mean 0 and standard deviation 1) to ensure features are on the same scale.

---

4. Default Estimator and WoE Binning

Objective:

To construct a **default estimator** (proxy) variable that classifies customers into **high-risk** or **low-risk** groups based on the likelihood of default.

Steps Taken:

- Visualized the transaction data in the **RFMS space** (Recency, Frequency, Monetary Value, Stability) to establish a boundary between good and bad customers.
- Used **Weight of Evidence (WoE)** and **Information Value (IV)** techniques to segment the data into bins for better interpretation.

---

5. Model Building

Objective:

To develop machine learning models that predict the likelihood of default and assign a credit score to each customer.

Key Models Chosen:

- **Logistic Regression**: A simple and interpretable model for binary classification.
- **Random Forest**: A powerful ensemble model that can capture non-linear relationships.
- **Gradient Boosting Machines (GBM)**: An advanced model that can improve accuracy by iteratively correcting errors.

Model Training and Evaluation:

- **Split the Data** into training and testing sets for unbiased evaluation.
- **Hyperparameter Tuning**: Used grid search to fine-tune model parameters.
- **Model Evaluation Metrics**:
    - **Accuracy**, **Precision**, **Recall**, **F1 Score**, and **ROC-AUC** to assess model performance.

---

6. Model Deployment: API for Real-Time Predictions

Objective:

To deploy the trained model as a **REST API** for real-time credit scoring.

Steps:

1. **Choose Framework**: Selected **FastAPI** due to its high performance and ease of use for deploying machine learning models.
2. **Load Model**: The trained model is loaded and used for prediction within the API.
3. **API Endpoints**:
    - Developed endpoints that accept input data (transaction details) and return predictions (e.g., default probability and credit score).
4. **Deployment**: Deployed the API to a web server or cloud platform for real-time usage.

---

7. Conclusion

This project lays a strong foundation for developing a **Credit Scoring Model** at Bati Bank. The following steps have been completed:

- Gained a comprehensive understanding of **credit risk**.
- Performed **exploratory data analysis** to uncover insights from the dataset.
- Completed initial steps for **feature engineering** and preparing the data for modeling.
- Selected appropriate **machine learning models** for predicting default and credit scores.

The next steps include finalizing the feature engineering process, tuning models, and setting up the real-time API for serving the predictions.

---

8. Future Steps and Recommendations

1. **Feature Enhancement**: Consider incorporating external data sources (e.g., social media activity, payment history) to improve prediction accuracy.
2. **Model Monitoring**: Continuously monitor model performance and retrain when necessary, especially as more data is collected.
3. **Real-Time Integration**: Integrate the credit scoring API into Bati Bank's systems to provide real-time credit assessments for customers applying for the Buy Now Pay Later service.

References

1. Basel II Capital Accord: Basel II Document
2. Understanding Credit Risk: World Bank Report
3. Weight of Evidence and Information Value: WoE and IV Explanation

---