**Credit Card Default Prediction Model Documentation**

**Overview**

This document outlines the development of a machine learning model designed to predict credit card defaults. The solution implements an XGBoost classifier with a systematic approach to data preprocessing, model training, and evaluation.

**Technical Approach:**

# 1. *Data Preprocessing Pipeline*

Missing Value Analysis and Handling

The preprocessing pipeline implements specialized handling for different types of features:

- Transaction Features : Missing values are filled with zeros, assuming missing transactions indicate no activity

- Onus Features :

- Missing values are imputed with median values

- Additional binary flags are created to capture missingness patterns

- Bureau Features : Median imputation is applied to preserve distribution characteristics

- Bureau Enquiry Features : Zeros are used for missing values, indicating no enquiries

The strategy is implemented in the `preprocess\_data()` function, which ensures consistent treatment across training and validation datasets.

*# 2. Algorithm Selection and Implementation*

XGBoost Classifier

The solution utilizes XGBoost (eXtreme Gradient Boosting) for several key reasons:

- Robust handling of non-linear relationships

- Built-in handling of missing values

- Excellent performance on tabular data

- Strong regularization capabilities to prevent overfitting

Configuration parameters:

```python

XGBClassifier(

objective='binary:logistic',

eval\_metric='auc',

use\_label\_encoder=False,

random\_state=42,

n\_estimators=100,

learning\_rate=0.1,

max\_depth=6

)

```

*# 3. Feature Engineering and Processing*

1. Data Splitting : Implementation of train-test split (80-20) with stratification

2. Feature Scaling : StandardScaler application to normalize feature distributions

3. Categorical Processing : Implicit handling through XGBoost's capability

*# 4. Model Training Process*

The training pipeline follows these steps:

1. Initial data preprocessing

2. Feature scaling using StandardScaler

3. Model training on the scaled features

4. Probability prediction generation

5. Performance evaluation on both training and test sets

**Key Insights and Observations**

*# Feature Importance Analysis*

The model provides insights into feature importance, revealing the most predictive variables for credit default. The top features are captured through XGBoost's built-in feature importance mechanism.

*# Data Distribution Patterns*

- Missing value patterns vary significantly across feature categories

- Transaction and bureau enquiry features show systematic missingness

- Onus features demonstrate random missing patterns

**Model Evaluation Metrics**

*# Primary Metrics*

1. Area Under the ROC Curve (AUC-ROC)

- Measures model's ability to distinguish between classes

- Robust to class imbalance

- Provides evaluation across multiple threshold values

2. Training vs Testing Performance

- Training AUC is monitored to detect overfitting

- Test AUC provides unbiased performance estimation

- Gap between train and test AUC indicates generalization capability

*# Validation Approach*

1. Cross-Validation : Implementation of k-fold cross-validation for robust performance estimation

2. Hold-out Validation : Final validation on completely unseen data

3. Probability Distribution Analysis : Statistical analysis of predicted probabilities

**Model Output and Deployment**

The final model produces probability scores for each account, saved in a structured format:

- account\_number: Unique identifier for each case

- predicted\_probability: Model's estimated probability of default

**Future Improvements**

*1. Feature Engineering Opportunities :*

- Creation of interaction terms

- Time-based feature aggregation

- Domain-specific feature creation

*2. Model Enhancements:*

- Hyperparameter optimization through grid search

- Ensemble approaches with multiple models

- Implementation of early stopping

*3. Validation Enhancements:*

- Additional cross-validation folds

- Time-based validation splits

- Population stability monitoring

**Technical Requirements**

- Python 3.x

- Key Libraries:

- scikit-learn

- XGBoost

- pandas

- numpy

- seaborn

This documentation provides a comprehensive overview of the credit card default prediction model, its implementation, and evaluation approach. The solution demonstrates robust handling of missing values, appropriate feature processing, and systematic model evaluation.