MODEL DOCUMENTATION

Statement of Purpose (for the model)

The purpose of this project is to build a predictive model to assess the likelihood of a loan being fully repaid or charged off. This model will assist lending institutions, such as Lending Club, in evaluating the risk associated with loan applications and making data-driven decisions during the underwriting process.

Foundational Data

The Data Used for Training / Testing

The dataset used for training and testing the models is derived from Lending Club's publicly available loan data. This dataset includes features related to loan applications, borrower financial profiles, and loan performance. The original dataset consisted of over **2 million rows and 145 columns**. For practical purposes, the dataset was reduced to a **subset of 200,000 rows**, while maintaining the proportions of the target variable (fully_paid).

To improve processing efficiency, columns that were irrelevant, redundant, or contained excessive missing values were removed. This included features that were only available after the loan issuance (e.g., recoveries, collection_recovery_fee) and features unrelated to the loan decision-making process (e.g., url, desc).

Any Filtering / Subsetting of the Data

1. **Loan Status Filtering:**
    - The dataset was filtered to include only loans with a final status of **"Fully Paid"** or **"Charged Off"**. This resulted in a binary classification problem, where:
        - fully_paid = 1 indicates a loan that was fully repaid.
        - fully_paid = 0 indicates a loan that was charged off (defaulted).
2. **Timeframe Filtering:**
    - Only loans issued after **August 2012** were retained to avoid shifts in patterns due to changing economic conditions or Lending Club policies.
3. **Missing Value Threshold:**
    - Features with more than **60% missing values** were excluded to maintain data quality. Columns with missing rates between **60%-70%** were manually inspected for relevance, and their earliest available dates were evaluated.
4. **Sampling:**

- A **stratified sampling technique** was used to create a 5% subset of the data, ensuring that the proportions of the target variable (fully_paid) remained consistent with the original dataset.

5. **Exclusions:**
    - Features with high cardinality (e.g., zip_code) or features that provided little to no predictive value (e.g., url, policy_code) were excluded.

Data Splitting

1. **Training and Testing Split:**
    - The dataset was split into **training (70%)** and **testing (30%)** sets. This split ensures that the model is trained on the majority of the data while retaining a significant portion for unbiased performance evaluation.
2. **Stratified Splitting:**
    - The splitting was stratified based on the target variable (fully_paid) to ensure that the class distribution was consistent across the training and testing sets.
3. **Validation Setup:**
    - For some experiments, a further split of the training data into training and validation subsets was performed to fine-tune hyperparameters and evaluate model performance on unseen data.

By following these steps, the foundational dataset was refined to ensure it was robust, representative, and suitable for building predictive models.

Features & Feature Engineering

A Description of the Data Used as Predictors:

1. **Numerical Predictors:**
    - Numerical features like loan_amnt, fico_range_low, fico_range_high, avg_cur_bal, and dti were included. These represent loan amounts, credit scores, and financial balance details that directly relate to the borrower's creditworthiness.
    - Feature correlation analysis was conducted to identify potential multicollinearity. For example, int_rate and fico_range_high were highly correlated; hence int_rate was dropped.
2. **Categorical Predictors:**
    - Categorical features such as addr_state, purpose, and application_type were retained.
    - Features with high cardinality, such as zip_code, were excluded to avoid overfitting.

- The categorical feature emp_length was grouped into broader categories (e.g., 0-1 years, 2-4 years, etc.) for simplification.

3. **Feature Selection:**
    - Features like sub_grade, installment, and tot_hi_cred_lim were removed as they were redundant or had high variance without substantial contribution to the model.

Handling Missing Values:

1. **Imputation of Missing Values:**
    - Columns with a high percentage of missing data (>60%) were analyzed further, and many were excluded due to their lack of informative value.
    - Missing continuous variables were imputed using linear regression predictions based on the relationship with other features in the dataset. For example:
        - Features like revol_util and mths_since_recent_bc had their missing values filled using model predictions.
2. **Categorical Feature Handling:**
    - Missing categorical values like emp_length were filled with a default category (0-1 years).
3. **Custom Processing:**
    - Columns like revol_util were preprocessed to standardize their numeric representation (e.g., converting percentages to decimals).

Transformations:

1. **Numerical Transformations:**
    - Continuous features were retained in their original scales, as tree-based models like Random Forest and XGBoost do not require standardization or normalization.
    - Feature engineering was applied to calculate credit_line_age, derived from issue_d and earliest_cr_line.
2. **Categorical Transformations:**
    - One-hot encoding was applied to categorical variables like addr_state and purpose to enable compatibility with machine learning algorithms.
    - Boolean features were converted to binary integer format for compatibility with the models.
3. **Additional Features:**
    - A new feature, fully_paid, was created as the target variable, mapping Fully Paid to 1 and Charged Off to 0. This feature ensures a binary classification problem.

These steps were designed to ensure that the dataset is clean, informative, and ready for modeling, with minimal noise and redundancy while retaining critical information for prediction.

MODELS

Describe the Two Models that were Built:

1. **Random Forest Classifier**
    - **Architecture:** A Random Forest classifier was implemented with 100 decision trees (n_estimators=100) and a maximum depth of 10 (max_depth=10).
    - **Data Characteristics:** The training set consists of 95% of the majority class (fully_paid=1) and 5% of the minority class (fully_paid=0), while the test set was (50%-50%).
2. **XGBoost Classifier**
    - **Architecture:** An XGBoost classifier with binary logistic regression as the objective function (binary:logistic), an AUC evaluation metric (eval_metric=auc), and a maximum depth of 6 (max_depth=6).
    - **Data Characteristics:** The training set was balanced with stratified sampling applied to maintain class proportions. Feature engineering and transformations ensured consistency and suitability for predictive modeling.

Describe and Compare the Performance of the Two Models:

1. **Random Forest Classifier :**
    - **Accuracy:** 0.5017
    - **ROC AUC:** 0.9571
    - **Classification Report:**
        - Precision for Class 0: 1.00
        - Recall for Class 0: 0.00
        - Precision for Class 1: 0.50
        - Recall for Class 1: 1.00
    - **Feature Importance:** The model heavily prioritized the last_fico_range_low and last_fico_range_high features, overlooking other potentially informative variables.
    - **Key Observations:**
        - Despite the high ROC AUC, the flawed training process caused the model to fail at predicting the minority class (fully_paid=0) effectively, with a recall of 0.00 for this class.

2. **XGBoost Classifier (Actual Model):**
    - **Accuracy:** 0.9093
    - **ROC AUC:** 0.9518
    - **Classification Report:**
        - Precision for Class 0: 0.77
        - Recall for Class 0: 0.79
        - Precision for Class 1: 0.95
        - Recall for Class 1: 0.94
    - **Feature Importance:** Balanced contributions from a variety of features such as last_fico_range_high, dti, and loan_amnt were observed, reflecting the model's ability to generalize better.
    - **Key Observations:**
        - The balanced dataset and robust architecture allowed XGBoost to perform well across both classes, demonstrating superior precision, recall, and F1-scores compared to the Random Forest model.

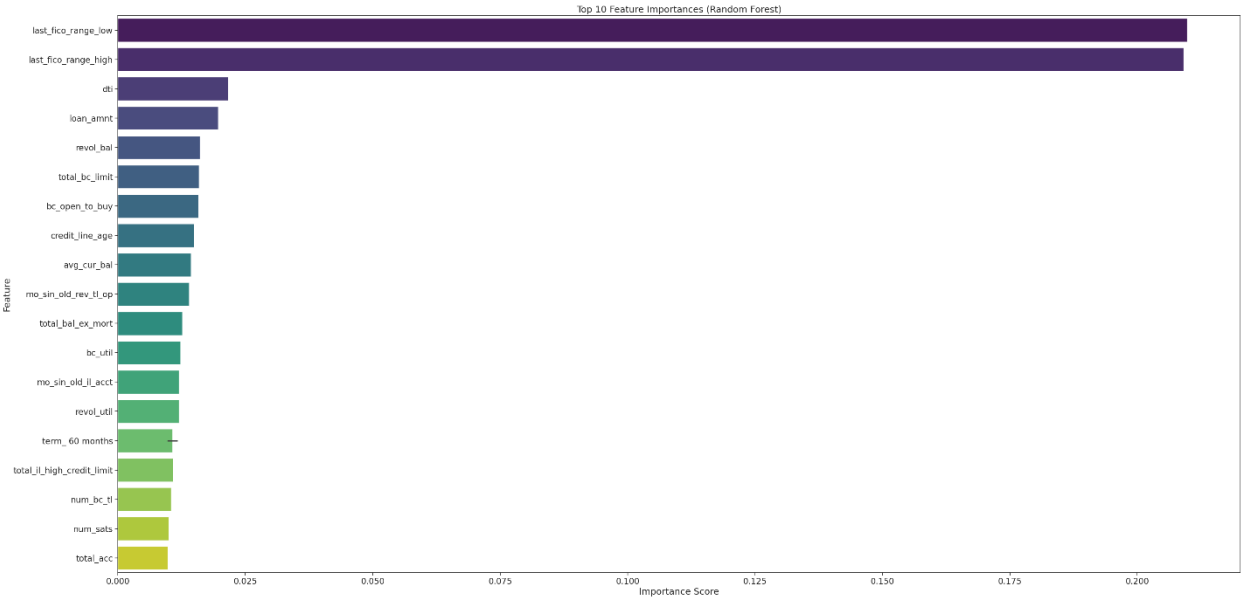
Select the "Winning" Model and Reasoning:

The **XGBoost Classifier** is the "winning" model for the following reasons:

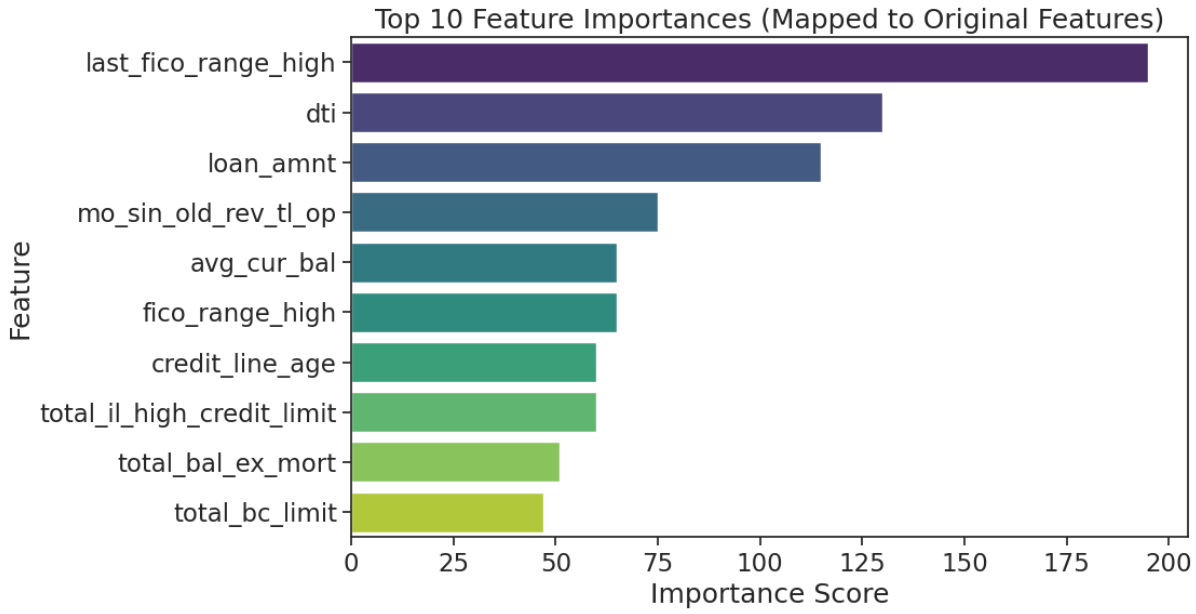
- **Balanced Performance:** The XGBoost model achieved high accuracy and ROC AUC, along with better precision, recall, and F1-scores across both classes.
- **Class Representation:** The model was trained on a balanced dataset, ensuring fair representation of both classes and minimizing bias.
- **Generalization Ability:** XGBoost demonstrated a more nuanced understanding of the feature set, as reflected in the diverse feature importance scores.
- **Reliability:** The XGBoost classifier effectively handled the balanced test set, making it more reliable for real-world deployment.

Interpretability / Explainability

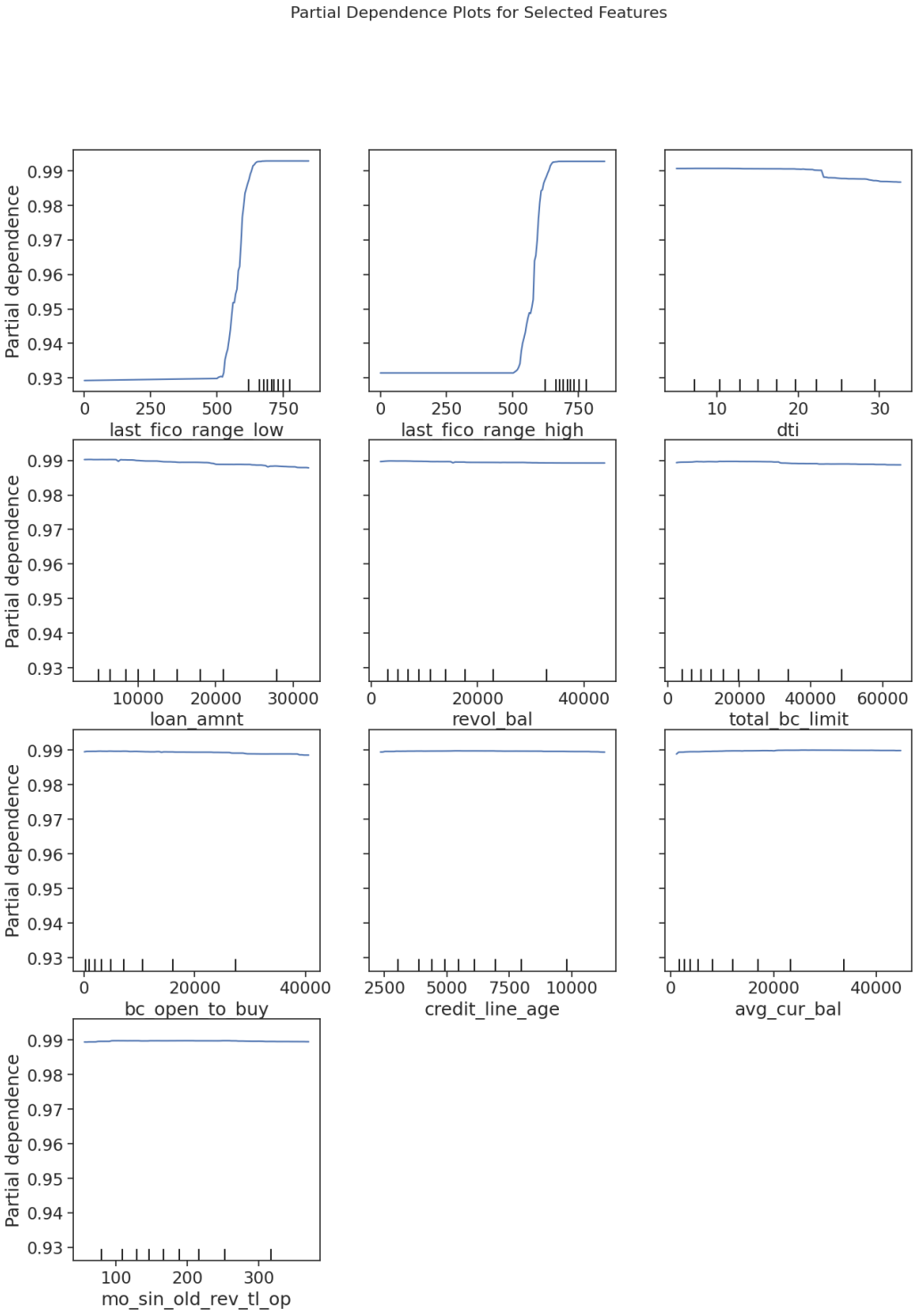
Feature Importance(Random Forest)



Feature Importance (XGBOOST)



Partial Dependency Plots (Random Forest)


Partial Dependence Plots for Selected Features
Partial dependence
last_fico_range_low
last_fico_range_high
dti
loan_amnt
revol_bal
total_bc_limit
bc_open_to_buy
credit_line_age
avg_cur_bal
mo_sin_old_rev_tl_op

Known Weaknesses / Risks of the Winning Model

1. **Bias from Feature Engineering:**
    - Features with high variance, such as total_il_high_credit_limit and total_bc_limit, may introduce biases, potentially affecting predictions for underrepresented groups.
2. **Overfitting Risks:**
    - The XGBoost model, despite its robustness, may risk overfitting if retraining is not regularly performed, especially given its complexity.
3. **Concept Drift:**
    - Economic conditions and lending practices may evolve, reducing model accuracy over time if it is not updated with recent data.

Suggested Controls for the Model

1. **Fairness Audits:**
    - Perform regular audits to detect and address bias in the model's predictions, ensuring fairness across demographic groups.
2. **Periodic Retraining:**
    - Retrain the model on updated datasets to mitigate the effects of concept drift and maintain its reliability.
3. **Explainability Tools:**
    - Use tools like SHAP or LIME to improve transparency and enable stakeholders to understand the model's decision-making process.
4. **Automated Monitoring:**
    - Deploy automated monitoring systems to track the model's performance and detect data distribution changes in real-time.