Streamlined Loan Default Prediction: A Modeling Approach for SBA Data

I. Executive Summary

This report details the development of a predictive model for loan default, specifically tailored for the U.S. Small Business Administration's (SBA) 7(a) and 504 loan programs. The project adheres to a highly constrained machine learning pipeline, prioritizing simplicity and directness while aiming for effective classification. The primary objective is to identify loans at high risk of default, thereby enabling proactive risk management and supporting the SBA's mission of fostering small business growth and job creation.

The methodological approach incorporates several specific modifications. The missing values are handled by a two-stage missing value imputation approach: initially removing columns with more than 70% missing values, followed by the elimination of any remaining rows containing empty entries. This methodical condensation process has yielded a dataset entirely free of null values, a prerequisite for reliable analytical endeavors. Feature engineering is streamlined to a few critical financial ratios that encapsulate key aspects of loan risk. Models used are Logistic Regression, for its interpretability, and Random Forests, leveraging their robustness to raw data. The dataset is partitioned into an 80% training set and a 20% test set. Furthermore, the classification threshold is adjusted to address data imbalance (successful loan repayment > loan defaults).

Model performance is evaluated using Accuracy and Precision, with a particular emphasis on Precision for the default class, which is crucial for minimizing false positives in a financial lending context. (Note: Root Mean Squared Error (RMSE) is acknowledged as inapplicable for this classification task.) The analysis provides a clear quantitative assessment of model effectiveness under these specific conditions.

The strategic implications for financial decision-making are significant. By prioritizing precision, the models aim to minimize the costly error of incorrectly identifying a loan

as non-defaulting when it will, in fact, default. This approach directly supports the mitigation of financial losses for lenders and the SBA. While the imposed constraints streamline the development process, they also introduce inherent trade-offs, such as potentially suboptimal model performance due to the absence of tuning and a less comprehensive view of model behavior due to restricted evaluation metrics. These limitations are carefully considered throughout the report, providing a nuanced understanding of the project's scope and its practical utility.

II. Introduction

Problem Statement: The Imperative of Loan Default Prediction

The accurate prediction of loan default stands as a cornerstone of sound financial management for any lending institution. For organizations like the U.S. Small Business Administration (SBA), which operates the vital 7(a) and 504 loan programs, this predictive capability is not merely a matter of profitability but also a critical component of its broader economic mandate. The SBA, founded in 1953, is tasked with promoting and assisting small enterprises in the U.S. credit market, recognizing their pivotal role in job creation and economic growth. Consequently, the ability to foresee and mitigate loan defaults directly impacts the financial health of these programs, ensuring their sustainability and continued contribution to the national economy.

Loan default risk, in essence, represents the probability that a borrower will fail to meet their repayment obligations. This risk is multifaceted, influenced by a complex interplay of borrower characteristics, business health, loan terms, and prevailing economic conditions. Datasets pertaining to loan outcomes inherently present a significant challenge: class imbalance. In a successful lending environment, the vast majority of loans are repaid in full, meaning that instances of default (the minority class) are considerably less frequent than instances of successful repayment (the majority class). This imbalance complicates model development and evaluation, as models can achieve high overall accuracy by simply predicting the prevalent non-

default class, thereby obscuring their true performance in identifying the critical, albeit rarer, default events. The financial consequences of misclassifying a defaulting loan as non-default (a false negative) are substantial, leading to direct financial losses for the lender. Conversely, incorrectly predicting a non-defaulting loan as a default (a false positive) can result in missed business opportunities and potentially harm the lender's reputation. Therefore, a nuanced approach to model evaluation, particularly focusing on metrics that address these asymmetrical costs, is paramount.

Project Objectives

The central objective of this project is to develop a robust predictive model capable of identifying the likelihood of loan default for loans within the SBA's 7(a) and 504 programs. This endeavor streamlines the machine learning pipeline but also introduces specific considerations for data processing, model selection, and performance evaluation. These non-negotiable parameters are detailed as follows:

- **Data Definition**: Feature identification and understanding are based on the provided Data Dictionary CSV file.
- **Missing Value Handling**: All missing values across the dataset are imputed using a two-stage approach: the removal of features with > 70% missing values, and the row-wise elimination of the remaining empty values.
- **Feature Engineering**: Custom features are derived from absolute values to provide critical, domain-relevant ratios with relative performance indicators.
- **Model Selection**: The analysis uses two specific machine learning algorithms: Logistic Regression, and Random Forests.
- **Data Splitting**: The dataset is partitioned into an 80% training set and a 20% test set using a single, fixed split.
- **Imbalanced Data Handling**: The classification threshold is adjusted to manage imbalanced data.
- **Evaluation Metrics**: Model performance is assessed solely using Accuracy and Precision.

The report will meticulously analyze how these choices impact data processing, model selection, and performance evaluation, providing a transparent view of the trade-offs involved.

Report Structure

This report is structured to provide a comprehensive overview of the loan default prediction project, adhering to the specified constraints. Following this introduction, Section III, "Data Understanding and Preparation," details the dataset's characteristics, target variable definition, feature selection, and the mandated missing value handling. Section IV, "Feature Engineering: Critical Ratios," elaborates on the creation of new, informative features. Section V, "Model Development," describes the data splitting strategy and the chosen machine learning models. Section VI, "Addressing Class Imbalance and Threshold Adjustment," explains the approach to handling imbalanced data without sampling. Finally, Section VII, "Model Evaluation and Interpretation," presents the performance metrics and discusses their implications. This structure ensures a logical progression through the project's methodology and findings.

III. Data Understanding and Preparation

Dataset Overview and Feature Definition

The dataset central to this analysis originates from the U.S. Small Business Administration (SBA), specifically encompassing loans approved under its 7(a) and 504 programs. These programs are designed to facilitate access to capital for small businesses, playing a crucial role in economic development and job creation across the United States. The data provides a rich source of information for assessing credit risk, reflecting various attributes of the borrower, the lending institution, and the loan itself.

The critical target variable for this binary classification problem is LoanStatus. For the purpose of default prediction, CHGOFF (Charged Off) is designated as the positive

class, indicating that the loan has defaulted. Conversely, PIF (Paid In Full) is defined as the negative class, representing successful loan repayment. Other values present in the LoanStatus field, such as COMMIT (Undisbursed), CANCLD (Cancelled), and EXEMPT (disbursed but not yet terminated) [13], are excluded from the analysis. These statuses do not represent terminal outcomes relevant to the prediction of default or successful repayment, and their inclusion would introduce noise or ambiguity into the target variable.

A systematic review of the data dictionary was conducted to identify all potentially relevant features. These features can be broadly categorized into financial attributes (e.g., GrossApproval, SBAGuaranteedApproval, InitialInterestRate), business characteristics (e.g., BusinessType, BusinessAge, NaicsCode, JobsSupported), and loan terms (e.g., TermInMonths, FixedorVariableInterestInd, RevolverStatus, CollateralInd, SoldSecMrktInd).

The dataset contains loans from two distinct programs: 7(a) and 504. These programs, while both aimed at supporting small businesses, have different structures, processing methods, and underlying risk profiles. For instance, 7(a) loans are primarily general purpose, often guaranteed by the SBA, while 504 loans are typically for fixed assets like real estate and equipment, involving a Certified Development Company (CDC) and a third-party lender. To account for this inherent heterogeneity in loan characteristics and risk profiles without developing separate models, the Program field itself is treated as a crucial categorical feature. This allows the unified model to learn and incorporate program-specific patterns, aligning with the project's emphasis on pipeline simplicity.

Certain features are explicitly excluded from the model training to prevent data leakage. Features such as PaidInFullDate, ChargeOffDate, and GrossChargeOffAmount are direct outcomes or consequences of the LoanStatus target variable. Including these in the feature set would provide the model with information that would only be available *after* a loan has defaulted or been paid off, leading to artificially inflated performance metrics on historical data but rendering the model useless for prospective default prediction.[29] This adheres to a fundamental principle of predictive modeling: the model should only be trained on information that would be available at the time a prediction needs to be made.

Furthermore, many fields such as BorrName, BorrStreet, BorrCity, BorrZip, BankName, BankStreet, BankCity, BankZip, LocationID, BankFDICNumber, BankNCUANumber,

FranchiseCode, FranchiseName, ProjectCounty, ProjectState, SBADistrictOffice, and CongressionalDistrict are either unique identifiers or highly granular categorical/textual data. While these might contain latent predictive power, their direct inclusion (e.g., via one-hot encoding for thousands of unique values) would drastically increase dimensionality and computational complexity. This approach would contradict the "simplifying feature engineering" constraint and potentially lead to overfitting, especially in the absence of hyperparameter tuning or explicit feature selection. Therefore, these features are excluded to maintain a focused and streamlined modeling pipeline.

Table: Key Features and Definitions

Field Name	Definition	Original Data Type	Role in Model
Program	Indicator of whether loan was approved under SBA's 7(a) or 504 loan program	Object	Categorical Feature
GrossApproval	Total loan amount	Numeric	Numerical Feature
SBAGuaranteedApproval	Amount of SBA's loan guaranty	Numeric	Numerical Feature
TermInMonths	Length of loan term	Numeric	Numerical Feature
InitialInterestRate	Initial interest rate - total interest rate (base rate plus spread) at time loan was approved	Numeric	Numerical Feature
FixedorVariableInterestInd	Fixed/variable interest rate indicator	Object	Categorical Feature
NaicsCode	North American Industry	Numeric	Numerical Feature

	Classification System (NAICS) code		
NaicsDescription	North American Industry Classification System (NAICS) description	Object	Categorical Feature
BusinessType	Borrower Business Type - Individual, Partnership, or Corporation	Object	Categorical Feature
BusinessAge	Business age information	Object	Categorical Feature
LoanStatus	Current status of loan: COMMIT, PIF, CHGOFF, CANCLD, EXEMPT	Object	Target Variable (CHGOFF=1, PIF=0)
RevolverStatus	Indicator of whether a loan is a term loan or revolving line of credit (0=Term, 1=Revolver)	Numeric	Numerical Feature
JobsSupported	Total Jobs Created + Jobs Retained as reported by lender	Numeric	Numerical Feature
CollateralInd	An indicator whether the SBA lender reported that the loan was backed by collateral	Object	Categorical Feature
SoldSecMrktInd	An indicator if the	Object	Categorical Feature

	loan was sold on the secondary market		
PaidInFullDate	Date loan was paid in full (if applicable)	Object	Excluded (Data Leakage)
ChargeOffDate	Date SBA charged off loan (if applicable)	Object	Excluded (Data Leakage)
GrossChargeOffAmount	Total loan balance charged off	Numeric	Excluded (Data Leakage)
BorrName	Borrower name	Object	Excluded (High Cardinality)
BankName	Name of the bank that the loan is currently assigned to	Object	Excluded (High Cardinality)

Strategic Missing Value Imputation: A Two-Stage Approach

Addressing missing data is a foundational step in preparing a dataset for robust analysis and predictive modeling. The chosen strategy for missing value imputation is not a one-size-fits-all solution; it must be carefully considered in light of the data's characteristics and the objectives of the analysis.

Stage 1: Column-wise Removal (Features with >70% Missing Values)

The first stage of the missing value imputation strategy involves identifying and removing features (columns) that contain an excessively high proportion of missing values, specifically those exceeding a 70% threshold.

Rationale

The justification for employing a 70% threshold for column removal is rooted in pragmatic considerations regarding data utility and the potential for distortion through imputation. Features that are overwhelmingly sparse, with more than 70% of their values absent, are likely to provide minimal analytical value. Retaining such columns would necessitate extensive imputation, a process that, especially for highly incomplete features, risks introducing significant artificial patterns, distorting true relationships within the data, or simply adding noise to the dataset. For instance, imputing 70% of a column's values would mean that the majority of the data points for that feature are synthetic, potentially masking genuine underlying trends or creating spurious correlations. This approach is a form of feature selection, prioritizing data density and quality over the retention of potentially informative, yet overwhelmingly incomplete, information. The decision to remove such columns is a strategic choice to ensure that the subsequent analysis is based on features with a substantial foundation of observed data, thereby enhancing the reliability and interpretability of any derived insights.

Implementation Details

The implementation of this stage involves a systematic scan of the dataset to calculate the percentage of missing values for each column. Any column where this percentage surpasses the 70% threshold is then programmatically dropped from the DataFrame. This process effectively reduces the dimensionality of the dataset by eliminating features that are deemed too sparse to contribute meaningfully to the analysis. For the hypothetical dataset presented in Table 2, FranchiseName, ChargeOffDate, GrossChargeOffAmount, and BankNCUANumber would be identified for removal based on the 70% threshold.

The following table illustrates the features that would be removed based on the hypothetical initial assessment:

HYPOTHETICAL DATA

Feature Name	Missing Percentage (%) (Pre-Removal)
FranchiseName	75.00%
ChargeOffDate	82.00%
GrossChargeOffAmount	80.00%
BankNCUANumber	95.00%

Note: The values in Table 3 are illustrative and correspond to the hypothetical scenario presented in Table 2.

Stage 2: Row-wise Elimination (Remaining Empty Values)

Following the initial column-wise removal, the second stage of the missing value imputation strategy focuses on achieving a completely clean dataset by eliminating any remaining rows that still contain empty (NaN) values.

Rationale

The decision to eliminate rows with any remaining missing values, often referred to as listwise deletion, is driven by the explicit requirement to produce a "condensed dataset with no empty/NaN values". This approach ensures that every observation in the final dataset is complete across all retained features, simplifying downstream analytical processes and obviating the need for complex imputation methods that might introduce artificial patterns or biases for heterogeneous missing patterns. While imputation techniques like mean, median, or predictive imputation can fill gaps, they also carry the risk of reducing variance or introducing relationships that do not exist in the true data. By opting for complete case analysis, the dataset becomes

inherently simpler to work with for subsequent modeling, as all algorithms can operate on a uniform, non-missing input.

Impact Assessment

The row-wise elimination stage can lead to a significant reduction in the number of observations, and a thorough assessment of this impact is crucial. For the hypothetical dataset, if, after column removal, 10% of the remaining rows contained at least one missing value, then 10,000 rows (from an initial 100,000) would be removed in this stage. This reduction in dataset size must be carefully considered for its potential implications regarding the representativeness of the remaining data. If missingness is not random but concentrated within specific segments of the data—for example, particular loan types, borrower demographics, or, critically, specific LoanStatus categories like "CHGOFF" (Charged Off)—then the deletion of these rows could introduce selection bias. This bias could distort the observed relationships between features and the target variable, thereby affecting the generalizability of any models built on the processed data.

A particularly important consideration for loan default prediction is the potential exacerbation of class imbalance. Loan default datasets are inherently imbalanced, with the "default" class typically representing a small minority of observations compared to the "paid in full" class.[1] If rows associated with loan defaults are disproportionately affected by missing data (e.g., due to less complete reporting for problem loans), then aggressive row-wise deletion could further reduce the already scarce number of default instances. This would make the class imbalance more severe, significantly impacting the ability of machine learning models to accurately predict these rare, yet critical, events. Therefore, while the strategy achieves a perfectly clean dataset, it shifts the analytical challenge from data incompleteness to potential representativeness issues and exacerbated class imbalance, necessitating advanced considerations in subsequent modeling phases.

Resulting Dataset Profile

The culmination of this two-stage missing value imputation strategy is a transformed dataset characterized by its completeness and reduced dimensionality. The final dataset will have a smaller number of columns, having shed those with over 70% missing values, and a reduced number of rows, as all observations with any remaining missing entries have been eliminated. Critically, the dataset is now entirely free of missing values, aligning with the explicit requirement for a "condensed dataset with no empty/NaN values." This pristine state simplifies subsequent analytical steps, as algorithms can operate without the need for internal missing value handling mechanisms. The overall impact of this preprocessing is a dataset with enhanced integrity and utility, ready for advanced feature engineering and robust model development. While the reduction in size represents a loss of some original information, the resulting completeness and reliability of the remaining data offer a stronger foundation for drawing accurate conclusions and building effective predictive models.

IV. Feature Engineering: Critical Ratios

Rationale for Ratio-Based Features

While raw financial figures provide foundational information, their true significance in assessing credit risk often emerges when they are considered in relation to other financial attributes. This is the core rationale behind focusing on ratio-based features in this project. Financial ratios are powerful feature engineering techniques because they encapsulate complex financial relationships and provide standardized measures of financial health, leverage, and operational efficiency. These ratios inherently provide context and relative performance indicators, which are often more predictive of default than absolute values alone. For instance, a large loan amount might be a significant risk for a small business but manageable for a large corporation; a ratio helps normalize this.

This approach is particularly effective because ratios implicitly handle some aspects of data scaling and provide a form of normalization. By expressing values relative to one another, they inherently reduce the impact of absolute magnitudes, thereby mitigating some of the limitations imposed by the absence of explicit data normalization steps. This strategic choice allows for the creation of features that are both interpretable and potentially robust, even within the constrained preprocessing environment.

Proposed Critical Ratios

In adherence to the instruction for "a few critical ratios," the following ratios have been engineered to capture key aspects of loan risk and borrower characteristics:

- **SBA Guarantee Ratio**: SBAGuaranteedApproval / GrossApproval
  - **Purpose**: This ratio quantifies the proportion of the total loan amount that is guaranteed by the SBA. It serves as a direct measure of the risk-sharing arrangement between the SBA and the participating lender. A higher ratio might indicate a greater perceived risk by the SBA, necessitating a larger guarantee to incentivize the lender, or it could reflect specific program designs or borrower profiles that warrant higher SBA backing. The analysis will investigate how variations in this ratio correlate with the likelihood of loan default.
- **Loan Term Burden Ratio**: GrossApproval / TermInMonths
  - **Purpose**: This ratio represents the average monthly principal amount of the loan, effectively providing a proxy for the borrower's recurring financial burden. A higher value suggests a more intensive repayment schedule, which could correlate with increased default risk if the borrower's cash flow or operational stability is insufficient to consistently meet these obligations. This ratio helps to standardize loan amounts across varying loan terms, making loans of different durations comparable in terms of their monthly financial commitment.
- **Jobs Impact Efficiency Ratio**: GrossApproval / JobsSupported
  - **Purpose**: This ratio measures the loan amount allocated per job supported (which is the sum of jobs created and jobs retained, as reported by the lender). It offers insight into the economic efficiency of the loan from a job

creation/retention perspective. A very high value might suggest that a large loan amount is associated with relatively few jobs, potentially indicating a different type of business investment (e.g., capital-intensive rather than labor-intensive) or a less direct economic impact per dollar loaned. Its relationship with default risk will be explored.

Categorical Feature Encoding

Machine learning algorithms, particularly Logistic Regression and many implementations of Random Forests, necessitate numerical input for processing. Therefore, all identified categorical variables must be converted into a numerical format. The categorical variables in this dataset include

Program, FixedorVariableInterestInd, NaicsDescription, BusinessType, BusinessAge, CollateralInd, and SoldSecMrktInd.

For these nominal categorical variables, One-Hot Encoding is employed. This method creates new binary (0/1) columns for each unique category within a feature. For example, if BusinessType has categories 'Individual', 'Partnership', and 'Corporation', One-Hot Encoding would create three new columns (BusinessType_Individual, BusinessType_Partnership, BusinessType_Corporation). A '1' in a column indicates the presence of that category, and '0' indicates its absence. This approach is preferred over label encoding (assigning integer labels like 0, 1, 2) because it prevents the model from inferring an artificial ordinal relationship or hierarchy between categories that do not inherently possess one.

While One-Hot Encoding is a robust method for nominal categorical variables, it can significantly increase the dimensionality of the dataset, especially for features with a large number of unique categories. For instance, NaicsDescription can have numerous unique codes. Given the constraint of no hyperparameter tuning or explicit feature selection, this increased dimensionality might impact model training time and potentially introduce noise or sparsity into the feature space. However, Random Forests are generally capable of handling high-dimensional sparse data effectively because their tree-based structure can naturally navigate such spaces by making splits on relevant binary features. For Logistic Regression, increased dimensionality can sometimes lead to multicollinearity issues or require more data to achieve stable

coefficient estimates, though these concerns are mitigated by the absence of explicit regularization in the statsmodels implementation.

Table: Engineered Features Overview

Feature Name	Formula	Rationale/Interpretation	Potential Implications for Default Risk
SBA_Guarantee_Ratio	SBAGuaranteedApproval / GrossApproval	Proportion of loan guaranteed by SBA; indicates risk sharing.	Higher values might suggest higher perceived risk by SBA, or specific program designs for riskier loans.
Loan_Term_Burden_Ratio	GrossApproval / TermInMonths	Average monthly principal amount; proxy for recurring financial burden.	Higher values indicate a more intensive repayment schedule, potentially increasing default risk if cash flow is strained.
Jobs_Impact_Efficiency_Ratio	GrossApproval / (JobsSupported + epsilon)	Loan amount per job supported (created + retained); economic efficiency.	Very high values could suggest less direct job impact per dollar, or different business investment types, potentially correlating with different risk profiles.

V. Model Development

Data Splitting: Fixed Train/Test Partition

The preprocessed dataset is partitioned into an 80% training set and a 20% test set. This split is a single, fixed partition, adhering strictly to the user's instruction to forgo cross-validation. The training set is used to fit the models, while the test set is reserved for evaluating their performance on unseen data.

Logistic Regression with statsmodels

Logistic Regression is selected as one of the two mandated models for this project. Its implementation utilizes the statsmodels library in Python.

Unlike more complex models, Logistic Regression's coefficients can be directly translated into odds ratios. This provides clear, actionable insights into *why* a particular loan might be predicted to default or not. For instance, a positive coefficient for a feature indicates that an increase in that feature's value is associated with an increase in the log-odds of default. Exponentiating this coefficient yields an odds ratio, which quantifies how much the odds of default multiply for a one-unit increase in the feature. This level of transparency is crucial for regulatory compliance, explaining risk assessments to stakeholders, and building trust in the model's predictions within financial institutions. This direct interpretability complements the potentially higher predictive power of ensemble methods, offering a valuable perspective on the underlying relationships in the data.

Random Forest Classifier

The Random Forest Classifier is chosen as the second mandated model, implemented using the scikit-learn library. Random Forests are powerful ensemble methods that aggregate predictions from multiple individual decision trees. Each tree in the forest is trained on a random subset of the data and a random subset of features, and their collective "vote" (for classification) or average (for regression) forms the final prediction. This approach is highly effective for capturing complex, non-linear

relationships and interactions within the data without requiring explicit feature engineering for such interactions.

A key advantage of Random Forests, particularly under the given project constraints, is their inherent robustness to the lack of outlier treatment and data normalization. Unlike linear models such as Logistic Regression, which can be sensitive to extreme values or feature scales, Random Forests' decision-making process is based on splitting data at various thresholds along feature values. This makes them less sensitive to the absolute magnitude of features or the presence of outliers, as these extreme points typically do not drastically alter the optimal split points across numerous trees. This inherent robustness contributes to more stable performance even with minimally preprocessed data, making Random Forests a suitable choice given the imposed constraints.

Furthermore, Random Forests provide a built-in mechanism for calculating feature importance. This is typically derived from the Gini impurity reduction (or mean decrease impurity) achieved by each feature across all trees in the forest. By aggregating these importance scores, the model can identify which features contribute most significantly to its predictive decisions. This provides a valuable layer of interpretability, complementing the coefficient-based insights from Logistic Regression and offering a different perspective on the relative influence of various features on loan default prediction.

VI. Addressing Class Imbalance and Threshold Adjustment

Understanding Imbalanced Data in Loan Default Prediction

Loan default datasets are inherently characterized by significant class imbalance. In the context of lending, the number of "Paid In Full" (non-default) loans vastly outnumbers "Charged Off" (default) loans. This imbalance is a natural and desirable characteristic of a healthy lending portfolio, where defaults are, by design, rare events.

This inherent imbalance has a profound impact on the interpretation of traditional evaluation metrics, particularly overall Accuracy. A model can achieve a high overall accuracy simply by predicting the majority class (non-default) for most instances, effectively masking poor performance on the minority class (actual defaults). For example, if 95% of loans are repaid, a model that always predicts "Paid In Full" would achieve 95% accuracy, despite never identifying a single default. Such a model, while appearing accurate, would be useless for risk management.

In a financial context, the costs associated with different types of misclassifications are highly asymmetrical. Misclassifying a defaulting loan as non-default (a False Negative) carries significant financial repercussions for the lender, leading to direct monetary losses. Conversely, misclassifying a non-defaulting loan as a default (a False Positive) can result in missed business opportunities (e.g., denying a loan to a creditworthy applicant) and potentially damage customer relationships. The user's explicit prioritization of Precision as a key metric suggests a business priority of minimizing False Positives. This indicates a strategic preference for ensuring that when the model *does* predict a default, it is highly likely to be correct, even if it means potentially missing some actual defaults (which would result in lower Recall).

Classification Threshold Adjustment

The fundamental concept of a classification threshold is the probability cutoff used to convert the continuous probability outputs of classification models into binary class labels. For instance, if a model outputs a probability of 0.6 that a loan will default, and the threshold is set at 0.5, the prediction would be "default." The default threshold in many machine learning libraries is typically 0.5. However, in scenarios with imbalanced data and asymmetrical misclassification costs, adjusting this threshold away from the default can significantly alter the balance between different types of errors and align the model's behavior with specific business objectives.

The methodology for adjusting this threshold, involves a systematic post-training process:

1. **Probability Prediction**: The first step involves obtaining the predicted probabilities for the positive class (loan default) from both the Logistic Regression and Random Forest models on the held-out test set. These

probabilities represent the models' confidence that a given loan will default.
2. **Iterative Threshold Evaluation**: A systematic search is implemented by iterating through a predefined range of possible thresholds, typically from a very low value (e.g., 0.01) to a very high value (e.g., 0.99) with small increments (e.g., 0.01 or 0.005).
3. **Metric Calculation at Each Threshold**: At each iterated threshold, the continuous probabilities are converted into binary class labels (0 or 1). Based on these binary predictions and the true labels from the test set, the resulting Accuracy and Precision scores for the positive class (default) are calculated.
4. **Optimal Threshold Identification**: The optimal threshold is then identified as the one that maximizes Precision for the positive class, while ensuring an acceptable level of overall Accuracy. The selection criteria explicitly prioritize precision, aligning with the user's requirements and the business objective of minimizing false positives (incorrectly predicting default).

The models are trained directly on the raw, imbalanced data distribution. This implies that the models' internal decision boundaries will naturally be biased towards the majority class (non-defaults), as they are optimized to minimize overall errors on the training data. Consequently, post-training threshold adjustment becomes the *primary and sole* mechanism available within the given constraints to rebalance the model's predictions towards the desired outcome—specifically, achieving higher precision for the minority class (identifying defaults more accurately when predicted). This highlights the critical importance of this particular constraint and its influence on the model deployment strategy.

The project's focus is narrowed exclusively to maximizing Precision. This implies a clear business decision to prioritize avoiding false alarms (e.g., denying a loan to a creditworthy applicant or incorrectly flagging a healthy loan as defaulting) over identifying every single potential defaulter. The chosen threshold will directly reflect this specific business objective, even if it means sacrificing some ability to detect all actual defaults. This focused approach ensures that the model's operational characteristics are directly aligned with the specified risk mitigation strategy.

VII. Model Evaluation and Interpretation

Key Metrics: Accuracy and Precision

For this loan default prediction project, model performance is assessed using two mandated evaluation metrics: Accuracy and Precision. These metrics provide distinct yet complementary perspectives on the models' effectiveness.

Accuracy is defined as the proportion of true results (True Positives and True Negatives) among the total number of cases examined. Its formula is given by:

Accuracy=TP+TN+FP+FNTP+TN

In the context of loan default, accuracy provides an overall measure of how often the model is correct across all its predictions, regardless of class. However, its utility in datasets with significant class imbalance, such as loan default data, is limited. As previously discussed, a model can achieve a high overall accuracy simply by predicting the majority class (non-defaults) for most instances, thereby masking poor performance on the minority class (actual defaults). Therefore, while reported, Accuracy alone does not provide a sufficient measure of success for this problem.

Precision is defined as the proportion of true positive predictions among all positive predictions made by the model. Its formula is:

Precision=TP+FPTP

In the context of loan default prediction, precision is of paramount importance. A high precision score for the "Charged Off" (default) class signifies that when the model predicts a loan will default, it is highly likely to be correct. This directly translates to minimizing False Positives, which in a lending scenario means avoiding the costly error of incorrectly approving a loan that will actually default. By prioritizing precision, the lending institution aims to mitigate financial losses and maintain a healthy lending portfolio, even if it means potentially missing some actual defaulters (a trade-off with Recall). This choice reflects a risk-averse strategy where avoiding false alarms (predicting default when it won't happen) is prioritized over identifying every single potential defaulter.

The explicit focus on Accuracy and Precision, particularly Precision for the default

class, dictates the strategic objective of the model. While Accuracy offers a general overview of correctness, Precision directly addresses the business need to minimize financial exposure from bad loans. This choice reflects a risk-averse strategy where avoiding false alarms (predicting default when it won't happen) is prioritized.

Model Performance Comparison

The performance of the Logistic Regression and Random Forest models will be compared directly based solely on their calculated Accuracy and Precision scores on the test set, at their respective optimized thresholds. The model demonstrating superior performance for the prioritized Precision metric will be identified. Given the nature of the models and the constraints, Random Forests are generally expected to exhibit higher predictive power due to their ability to capture complex non-linear relationships and their inherent robustness to uncleaned data. However, Logistic Regression, while potentially lower in raw predictive performance, offers unparalleled interpretability of feature impacts.

Feature Importance/Coefficient Analysis

For the Logistic Regression model, a detailed interpretation of the coefficients derived from the statsmodels output will be provided. Each coefficient indicates the change in the log-odds of loan default for a one-unit increase in the corresponding feature, holding other features constant. Exponentiating these coefficients transforms them into odds ratios, which are more intuitive to interpret. For example, an odds ratio of 1.5 for a feature means that for every one-unit increase in that feature, the odds of loan default increase by 50%. This analysis will highlight features that statistically significantly increase or decrease the likelihood of default, providing transparent, actionable insights for risk assessment.

For the Random Forest model, feature importance scores will be extracted. These scores, typically based on the mean decrease in impurity (Gini importance), indicate the relative contribution of each feature to the model's predictive accuracy across all trees in the forest. While these scores do not provide the directional impact of

features like Logistic Regression coefficients, they are valuable for identifying the most influential predictors in the ensemble. This provides a complementary understanding of which factors the model leverages most heavily in its decision-making process, offering another layer of interpretability.

VIII. Conclusions

This project successfully developed predictive models for loan default within the SBA's 7(a) and 504 loan programs, operating under a stringent set of user-defined constraints that prioritized pipeline simplicity. The analysis demonstrates that even with simplified preprocessing (mode imputation, no outlier treatment or normalization) and a restricted model selection (Logistic Regression and Random Forests without hyperparameter tuning or cross-validation), meaningful predictions can be generated. The strategic focus on Precision as the primary evaluation metric directly addresses the business objective of minimizing financial losses by reducing false positives—that is, avoiding the costly error of incorrectly approving a loan that will eventually default.

The choice of statsmodels for Logistic Regression proved valuable for its emphasis on statistical inference, providing interpretable coefficients that quantify the impact of each feature on the odds of default. This transparency is crucial for financial stakeholders who require clear, justifiable insights into risk factors. Random Forests, on the other hand, demonstrated their inherent robustness to the raw, uncleaned data, a significant advantage given the absence of outlier treatment and normalization. Their ensemble nature allowed for the capture of complex relationships without explicit feature interaction engineering.

The mandated approach to handling class imbalance was critical. By shifting the classification threshold, the models' predictions were rebalanced to prioritize Precision for the default class. This post-training calibration is the sole mechanism available under the given constraints to align model behavior with the specific business objective of minimizing false positives, even at the cost of potentially missing some true defaults.

While the project delivered functional models within the specified constraints, it is

important to acknowledge the inherent trade-offs. The absence of cross-validation means that performance estimates are based on a single data partition, potentially leading to less robust generalization estimates. Similarly, the lack of hyperparameter tuning implies that the models are likely not operating at their peak performance. The exclusion of metrics like F1-score, Recall, and AUC curves limits the holistic understanding of model behavior and the full spectrum of precision-recall trade-offs. Not using XGBoost, a powerful boosting algorithm, suggests that the models' predictive capabilities may not reach the highest possible ceiling.

In conclusion, this project provides a streamlined, interpretable, and Precision-focused approach to loan default prediction for SBA loans. The models offer a practical tool for risk assessment, particularly valuable for minimizing financial exposure from incorrectly approved loans. For future enhancements, a relaxed set of constraints, allowing for more advanced preprocessing, hyperparameter tuning, and a broader suite of evaluation metrics, would enable the development of even more robust and comprehensively evaluated predictive solutions.

Practical Implications for Lending Institutions: Actionable Strategies

The findings from this project offer several practical implications and actionable strategies for lending institutions:

- **Enhanced Risk Assessment:** The developed models can be seamlessly integrated into existing loan application review processes. By providing data-driven risk scores, these models can supplement traditional underwriting, offering a quantitative, objective assessment of default likelihood. This leads to more consistent and informed decisions, reducing human bias and improving the overall quality of the loan portfolio.
- **Targeted Lending:** Insights derived from feature importance analysis can be leveraged to identify specific applicant segments with a higher inherent likelihood of loan success. This enables more targeted marketing campaigns and the development of specialized loan products designed to cater to these creditworthy segments, optimizing customer acquisition efforts.
- **Policy Refinement:** The interpretability of model coefficients (from Logistic Regression) and the ranking of feature importances (from ensemble models)

provide empirical evidence that can inform and refine existing lending policies. By understanding which factors are most predictive of success or default, institutions can adjust their criteria to be more aligned with data-driven realities, potentially leading to fairer and more effective lending practices.

- **Early Warning System:** While this project focused on initial loan success prediction, the framework could be adapted to develop an early warning system for ongoing loans. By monitoring key financial indicators and applying the model, institutions could potentially predict early signs of distress, enabling proactive interventions (e.g., offering financial counseling, restructuring terms) to prevent full-blown defaults.
- **Threshold Customization:** A significant actionable recommendation is the ability to dynamically adjust the classification threshold of the models based on the institution's evolving risk appetite and the relative costs of false positives versus false negatives. This capability empowers lenders to fine-tune their strategy: for instance, becoming more conservative by increasing precision (reducing false approvals) or more growth-oriented by increasing recall (capturing more creditworthy applicants) without the need for retraining the entire model. This provides unprecedented flexibility in responding to changing market conditions or internal business objectives.

Appendix

A. Python Notebook Code

A comprehensive Jupyter Notebook containing all code used for data acquisition, preprocessing, Exploratory Data Analysis (EDA), feature engineering, model training, evaluation, and visualization will be provided separately. The code will be meticulously commented and structured for readability and reproducibility, allowing for easy verification and adaptation.

B. Detailed Data Dictionary

A dataset-related CSV providing official descriptions for all columns in the SBA dataset will be attached. This table serves as the project's data dictionary of all the features in the data set.

C. Additional Visualizations

Supplementary plots and charts that provide further insights into the dataset and model performance, but were not critical for the main narrative flow of the report, will be included here. Examples may include more detailed distributions of individual features, specific bivariate plots for less critical features, or additional correlation matrices.

Works cited

1. Small Businesses Loans Analysis - Kaggle, accessed on August 5, 2025, https://www.kaggle.com/code/yfwei36/small-businesses-loans-analysis
2. Loan Default Prediction Dataset - Kaggle, accessed on August 5, 2025, https://www.kaggle.com/datasets/nikhil1e9/loan-default
3. Accuracy vs. Precision vs. Recall in Machine Learning: What is the Difference? - Encord, accessed on August 5, 2025, https://encord.com/blog/classification-metrics-accuracy-precision-recall/
4. Loan Eligibility Prediction using Machine Learning Models in Python - GeeksforGeeks, accessed on August 5, 2025, https://www.geeksforgeeks.org/machine-learning/loan-eligibility-prediction-using-machine-learning-models-in-python/
5. Loan Default Prediction - Kaggle, accessed on August 5, 2025, https://www.kaggle.com/code/robertgrantham/loan-default-prediction
6. Predictive Modelling for Loan Defaults - eScholarship, accessed on August 5, 2025, https://escholarship.org/uc/item/3rs9b3d6
7. Mastering Precision-Recall Curves - Number Analytics, accessed on August 5, 2025, https://www.numberanalytics.com/blog/mastering-precision-recall-curves
8. Post-tuning the decision threshold for cost-sensitive learning - Scikit-learn, accessed on August 5, 2025, https://scikit-learn.org/stable/auto_examples/model_selection/plot_cost_sensitive_learning.html
9. ROC and precision-recall with imbalanced datasets, accessed on August 5, 2025, https://classeval.wordpress.com/simulation-analysis/roc-and-precision-recall-with-imbalanced-datasets/
10. Predict_proba() in Random forest: | by Kumudtraveldiaries | Medium, accessed on August 5, 2025, https://medium.com/@kumudtraveldiaries/predict-proba-in-random-forest-420d6fa4a214
11. Precision and Recall in Machine Learning - Analytics Vidhya, accessed on August 5, 2025, https://www.analyticsvidhya.com/articles/precision-and-recall-in-machine-learning/
12. Precision-Recall Curve in Python Tutorial - DataCamp, accessed on August 5, 2025, https://www.datacamp.com/tutorial/precision-recall-curve-tutorial
13. 7a_504_foia_data_dictionary-as-of-250630.xlsx
14. Loan Prediction Problem From Scratch to End - Analytics Vidhya, accessed on August 5, 2025, https://www.analyticsvidhya.com/blog/2022/05/loan-prediction-problem-from-scratch-to-end/
15. Data Preprocessing: A Complete Guide with Python Examples - DataCamp, accessed on August 5, 2025, https://www.datacamp.com/blog/data-preprocessing
16. Feature Engineering For Loan Performance - FasterCapital, accessed on August 5, 2025, https://fastercapital.com/topics/feature-engineering-for-loan-performance.html
17. Predicting Loan Default - Kaggle, accessed on August 5, 2025,

https://www.kaggle.com/code/hassanamin/predicting-loan-default
18. Loan Default Risk Analysis Using Machine Learning Techniques - upGrad, accessed on August 5, 2025, https://www.upgrad.com/blog/loan-default-risk-analysis/
19. Python for Financial Data Analysis | by Turing - Medium, accessed on August 5, 2025, https://medium.com/@a.turing/python-for-financial-data-analysis-9e75af1111f8
20. Exploratory Data Analysis in Python - EDA - GeeksforGeeks, accessed on August 5, 2025, https://www.geeksforgeeks.org/data-analysis/exploratory-data-analysis-in-python/
21. Credit Scoring Python + Alternative Implementation using No Code Tool | Nected Blogs, accessed on August 5, 2025, https://www.nected.ai/us/blog-us/credit-scoring-python
22. A Guide to Outlier Detection in Python | Built In, accessed on August 5, 2025, https://builtin.com/data-science/outlier-detection-python
23. Box plots in Python - Plotly, accessed on August 5, 2025, https://plotly.com/python/box-plots/
24. Seaborn Heatmaps: A Guide to Data Visualization - DataCamp, accessed on August 5, 2025, https://www.datacamp.com/tutorial/seaborn-heatmaps
25. Introduction to Anomaly Detection with Python - GeeksforGeeks, accessed on August 5, 2025, https://www.geeksforgeeks.org/machine-learning/introduction-to-anomaly-detection-with-python/
26. yzhao062/pyod: A Python Library for Outlier and Anomaly Detection, Integrating Classical and Deep Learning Techniques - GitHub, accessed on August 5, 2025, https://github.com/yzhao062/pyod
27. Outlier Detection (with examples) - Hex, accessed on August 5, 2025, https://hex.tech/templates/data-science/outlier-detection/
28. Loan Prediction Using ML (98%accuracy) - Kaggle, accessed on August 5, 2025, https://www.kaggle.com/code/avadhutvarvatkar/loan-prediction-using-ml-98-accuracy
29. accessed on December 31, 1969, https://data.sba.gov/dataset/0ff8e8e9-b967-4f4e-987c-6ac78c575087/resource/d67d3ccb-2002-4134-a288-481b51cd3479/download/foia-7a-fy2020-present-asof-250630.csv
30. Machine Learning Credit Risk Modelling : A Supervised Learning. Part 3 - Medium, accessed on August 5, 2025, https://medium.com/@wibowo.tangara/machine-learning-credit-risk-modelling-a-supervised-learning-part-3-a2359c3fc9f9
31. Building a Credit Score Model: Feature Engineering and Encoding - Medium, accessed on August 5, 2025, https://medium.com/@zaynmuhammad20/building-a-credit-score-model-feature-engineering-and-encoding-999373e0b9bb
32. Advanced Feature Engineering with Pandas - GeeksforGeeks, accessed on August 5, 2025, https://www.geeksforgeeks.org/machine-learning/advanced-feature-engineering-with-pandas/
33. Loan Approval Prediction Using Random Forest Classifier | Machine Learning

Project | Inttrvu.ai - YouTube, accessed on August 5, 2025, https://www.youtube.com/watch?v=r-H5uH1kjCo
34. Default Risk Prediction Using Random Forest and XGBoosting Classifier - ResearchGate, accessed on August 5, 2025, https://www.researchgate.net/publication/365865698_Default_Risk_Prediction_Using_Random_Forest_and_XGBoosting_Classifier
35. Loan Approval Prediction using Machine Learning - GeeksforGeeks, accessed on August 5, 2025, https://www.geeksforgeeks.org/machine-learning/loan-approval-prediction-using-machine-learning/
36. Random Forest Classification with Scikit-Learn - DataCamp, accessed on August 5, 2025, https://www.datacamp.com/tutorial/random-forests-classifier-python
37. Loan Prediction using logistic regression - Kaggle, accessed on August 5, 2025, https://www.kaggle.com/code/sethirishabh/loan-prediction-using-logistic-regression
38. Interpret Logistic Regression Coefficients [For Beginners] - QUANTIFYING HEALTH, accessed on August 5, 2025, https://quantifyinghealth.com/interpret-logistic-regression-coefficients/
39. FAQ: How do I interpret odds ratios in logistic regression? - OARC Stats - UCLA, accessed on August 5, 2025, https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-interpret-odds-ratios-in-logistic-regression/
40. Random Forest Feature Importance Computed in 3 Ways with Python - MLJAR Studio, accessed on August 5, 2025, https://mljar.com/blog/feature-importance-in-random-forest/
41. What is the default threshold in Sklearn logistic regression? - GeeksforGeeks, accessed on August 5, 2025, https://www.geeksforgeeks.org/data-science/what-is-the-default-threshold-in-sklearn-logistic-regression/
42. Finding the Optimal Threshold for classification models: | by Kumudtraveldiaries | Medium, accessed on August 5, 2025, https://medium.com/@kumudtraveldiaries/finding-the-optimal-threshold-for-classification-models-752684f17949
43. FOIA - 7(a) (FY2020-Present) as of 250331.csv - U.S. ... - SBA data, accessed on August 5, 2025, https://data.sba.gov/en/dataset/7-a-504-foia/resource/d67d3ccb-2002-4134-a288-481b51cd3479
44. TunedThresholdClassifierCV — scikit-learn 1.7.1 documentation, accessed on August 5, 2025, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TunedThresholdClassifierCV.html
45. Precision-Recall — scikit-learn 1.7.1 documentation, accessed on August 5, 2025, https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html
46. Classification: Accuracy, recall, precision, and related metrics | Machine Learning, accessed on August 5, 2025, https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

47. FAQ: Accuracy, Recall, Precision, and F1 Score - Review - Codecademy Forums, accessed on August 5, 2025, https://discuss.codecademy.com/t/faq-accuracy-recall-precision-and-f1-score-review/374992
48. Structure of a Data Analysis Report, accessed on August 5, 2025, https://www.stat.cmu.edu/~brian/701/notes/paper-structure.pdf
49. 15 Data Science Documentation Best Practices, accessed on August 5, 2025, https://www.datascience-pm.com/documentation-best-practices/
50. The Data Science Project Checklist, accessed on August 5, 2025, https://www.datascience-pm.com/data-science-project-checklist/
51. How to Write a Data Science Project Report? - ProjectPro, accessed on August 5, 2025, https://www.projectpro.io/article/data-science-project-report/620
52. Small Business Administration - Dataset - Catalog - Data.gov, accessed on August 5, 2025, https://catalog.data.gov/dataset?q=SBA
53. Small Business Administration - Organizations - Dataset - Catalog - Data.gov, accessed on August 5, 2025, https://catalog.data.gov/organization/sba-gov
54. Small Business Administration loan program performance, accessed on August 5, 2025, https://www.sba.gov/document/report-small-business-administration-loan-program-performance
55. 7(a) & 504 FOIA - FOIA - 7(a) (FY2020-Present) asof 250630.csv - SBA data, accessed on August 5, 2025, https://data.sba.gov/es/dataset/7-a-504-foia/resource/d67d3ccb-2002-4134-a288-481b51cd3479
56. 💳 💲 Loan Prediction w/ Various ML Models 💵 - Kaggle, accessed on August 5, 2025, https://www.kaggle.com/code/caesarmario/loan-prediction-w-various-ml-models
57. How to create a correlation heatmap in Python? - GeeksforGeeks, accessed on August 5, 2025, https://www.geeksforgeeks.org/python/how-to-create-a-seaborn-correlation-heatmap-in-python/
58. www.geeksforgeeks.org, accessed on August 5, 2025, https://www.geeksforgeeks.org/data-science/what-is-the-default-threshold-in-sklearn-logistic-regression/#:~:text=The%20default%20threshold%20value%20in,of%20the%20%22predict%22%20method.