Loan Approval Prediction

Final Project Report

Course: CSE 572 – Data Mining Instructor: Prof. Yeonjung Lee Team: Data Corsair

Mohammad Hamza Choudhry
Arizona State University
mchoud26@asu.edu

Sai Gangaraju
Arizona State University
sgangar6@asu.edu

Ryan Sam Varghese
Arizona State University
rsvargh2@asu.edu

Aswath Sridhar
Arizona State University
asridh44@asu.edu

Abstract

This project investigates supervised learning methods for predicting bank loan approval decisions from applicant and loan characteristics. Manual credit assessment is often slow, inconsistent, and difficult to audit, motivating a data-driven, reproducible decision-support system. Using a publicly available Kaggle bank-loan dataset with demographic, financial, and credit-history attributes, we formulate loan approval as a binary classification task and build an end-to-end pipeline for preprocessing, exploratory data analysis, feature engineering, model selection, and evaluation.

We first establish a logistic-regression baseline within a scikit-learn pipeline with scaling and one-hot encoding, then compare a range of classifiers including Gaussian Naive Bayes, k-Nearest Neighbors, Support Vector Machines, Decision Trees, Random Forests, AdaBoost, Multilayer Perceptrons, and XGBoost. Class imbalance is addressed through stratified sampling and class-weighted loss functions. Hyperparameters for Random Forest and XGBoost are tuned via cross-validated grid search, and the strongest models are combined in a soft-voting ensemble. Beyond these standard approaches, we design a cost-sensitive stacked ensemble and a segment-specific expert ensemble that optimize a business-inspired profit function.

Tree-based ensembles substantially outperform the linear baseline in terms of ROC AUC and F1 score: tuned Random Forest achieves the highest F1, while tuned XGBoost yields the best ROC AUC. The ensembling strategies provide robust precision–recall trade-offs and higher expected profit, and our analysis highlights the importance of engineered ratio features such as loan-to-income, careful treatment of class imbalance, and evaluation metrics aligned with business risk.

1 Introduction

1.1 Background and Motivation

Credit risk assessment is a fundamental operation for financial institutions. Banks must decide whether to approve or reject loan applications based on incomplete and noisy information about applicants income, employment, indebtedness, past defaults, and requested loan terms. Traditionally, these decisions rely heavily on manual underwriting and rule based policies, which can be slow, inconsistent across officers, and difficult to audit or adapt as conditions change.

Data driven models offer the potential to support or partially automate loan decision making by providing consistent probability estimates of default or approval outcomes. When integrated into a broader decision support workflow, such models can enable pre screening of applications, prioritize human review, and support what if analyses for policy changes.

In this project, we study loan approval prediction using a structured tabular dataset containing both applicant level and loan level variables. Rather than designing a bespoke credit scoring model from scratch, our focus is on constructing a rigorous data mining pipeline that compares several standard classification algorithms, quantifies performance differences, and reflects on the trade offs between interpretability and predictive power in this application domain. We also explore a cost sensitive and segment specific modeling layer that more directly reflects business trade offs than a simple accuracy objective.

1.2 Problem Description

We consider the task of predicting whether a loan application will be approved (label 1) or rejected (label 0) given a vector of features describing the applicant and the loan. The dataset includes demographic attributes such as age, financial indicators such as annual income, loan amount, loan interest rate, and credit history length, and categorical descriptors such as loan intent, loan grade, home ownership status, and prior default history. The goal is to learn a mapping from these features to the binary approval label that generalizes well to unseen loan applications.

The core question is:

Can we predict whether an applicant will be approved for a loan based on historical application data, and can we design a model and decision rule that align with business risk preferences?

1.3 Importance

Accurate loan approval prediction has multiple benefits:

- **Operational efficiency.** A good model can pre screen applications, reducing manual workload for human underwriters and shortening processing time.
- **Risk management.** Probability estimates can be used to calibrate approval thresholds, interest rates, and portfolio composition, helping to manage default risk.
- **Fairness and consistency.** A well documented, data driven model may improve consistency across cases relative to purely manual decisions, which are subject to human biases and fatigue.
- **Auditability and policy analysis.** Models provide a transparent framework for assessing the impact of changing decision thresholds, features, or cost assumptions.

While our project does not address fairness or regulatory aspects in depth, it provides a technical foundation that could be extended with fairness aware or interpretable modeling techniques in future work.

1.4 Related Work

Credit scoring and loan approval prediction have been widely studied using both classical and modern machine learning methods. Traditional approaches include logistic regression and linear discriminant analysis, which remain popular due to their interpretability and ease of deployment. Tree based methods such as Decision Trees and Random Forests can capture nonlinear interactions between features while maintaining some interpretability via feature importances and path analysis [1]. Gradient boosting methods, including XGBoost, have become standard tools for tabular classification tasks due to their strong performance on heterogeneous feature sets [2, 3].

Recent work has explored deep neural networks and hybrid architectures, but these models typically require larger datasets, careful regularization, and more complex deployment pipelines. A complementary line of research considers cost sensitive learning, where misclassification costs are explicitly modeled, and segmentation based strategies, where different subpopulations are handled by specialized experts.

Our work aligns with this literature by systematically comparing a linear baseline (logistic regression), probabilistic and distance based methods (Naive Bayes and k Nearest Neighbors), margin based methods (Support Vector Machines), tree based models (Decision Trees, Random Forests, AdaBoost), gradient boosting (XGBoost), and a feed forward neural network (Multilayer Perceptron). As a novel extension of standard ensemble baselines, we develop a soft voting ensemble and two cost aware ensembles, a stacked model and a segment specific expert model, that explicitly reflect the different financial consequences of wrongly approving bad loans and rejecting good borrowers.

1.5 System Overview

At a high level, our loan approval prediction system consists of the following components.

Data ingestion and cleaning. We load the training and test sets from the Kaggle loan dataset, remove irrelevant identifiers such as the unique id column, and standardize feature names and types.

Exploratory data analysis. We examine the class distribution, missing value patterns, and feature distributions for both numerical and categorical variables. The class distribution for the target label is shown in Figure 1. Missing values per feature are summarized in Figure 2. Histograms of numerical features capture the distributions of age, income, loan amount, interest rate, employment length, credit history length, and the engineered loan percent income ratio (Figures 3 to 9). Bar plots describe the marginal distributions of categorical features such as loan intent, loan grade, home ownership, and previous default history (Figures 10 to 13). A correlation heatmap of numeric features and the target (Figure 17) highlights associations between income, loan percent income, and approval.

Preprocessing and feature engineering. We impute missing values, scale numerical variables, one hot encode categorical variables, and construct engineered features such as the ratio of loan amount to income. A boxplot of loan percent income by label (Figure 14) illustrates its separation between approved and rejected loans. Approval rates by loan grade and loan intent are analyzed in Figures 15 and 16.

Modeling. We wrap preprocessing and classifiers in scikit learn pipelines, train multiple models, and perform hyperparameter tuning for key ensembles. Individual and ensemble ROC curves are compared in Figures 18 and 22.

Evaluation and analysis. We use a stratified 80/20 train validation split and 5 fold cross validation to estimate out of sample performance. We compute ROC AUC, accuracy, precision, recall, and F1 score. Logistic regression confusion patterns and coefficient magnitudes are visualized in Figures 19, 20, and 21.

Cost sensitive and segmented decision layers. We build a stacked ensemble and a segment specific expert ensemble on top of the best models and analyze expected profit as a function of decision thresholds. The profit curve for the

stacked model and the profit heatmap for the segmented ensemble are shown in Figures 23 and 24.

Deployment artifact. We produce a final pipeline that accepts raw tabular data and outputs approval probabilities. This pipeline can be applied to the held out Kaggle test set to generate submission files.

1.6 Data Collection

The dataset used in this project is a publicly available tabular loan dataset from a Kaggle playground series on bank loan approval. The training split contains 58,645 labelled loan applications, and the competition test split contains 39,098 unlabeled applications. Each training example is annotated with a binary `loan_status` label indicating whether the loan was approved (1) or not (0). Approved loans constitute about 14% of the training data (8,350 approved vs. 50,295 rejected), so the positive class is a clear minority. The main feature groups are:

- **Numeric features:** applicant age, annual income, requested loan amount, loan interest rate, employment length in years, credit history length, and the ratio `loan_percent_income`.
- **Categorical features:** loan intent (for example debt consolidation, personal, education), loan grade (A to G), home ownership status (RENT, OWN, MORTGAGE), and prior default history (`cb_person_default_on_file`).

We split the data into training and validation sets using a stratified 80/20 split to preserve the empirical class distribution (Figure 1). The separate competition test set is used only for generating final predictions.

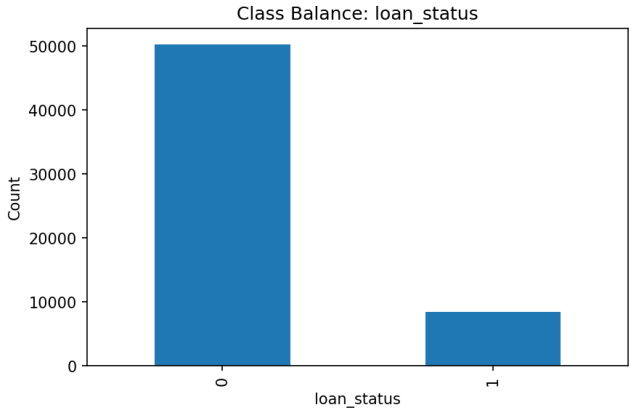

Figure 1. Class balance for the `loan_status` label.

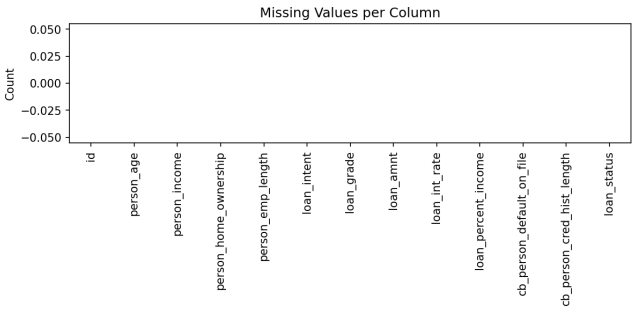

Figure 2. Fraction of missing values per feature.

1.7 Machine Learning System Components

The main components of our machine learning system are as follows.

Preprocessing. We treat identifier columns such as `id` as non-predictive and exclude them from the main feature transformers, then apply median imputation for numeric features, most frequent imputation for categorical features, standardization of numerical variables, and one-hot encoding of categorical variables with `handle_unknown="ignore"`.

Modeling. We consider a logistic regression baseline with `class_weight="balanced"` and a collection of candidate models: Gaussian Naive Bayes, k Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, AdaBoost, Multilayer Perceptron, and XGBoost. Hyperparameter tuning is performed for Random Forest and XGBoost. We then construct a soft voting ensemble, a stacked ensemble, and a segment specific expert ensemble.

Evaluation. We use a stratified train validation split and 5 fold cross validation, with metrics including ROC AUC, accuracy, precision, recall, F1 score, and business inspired expected profit. Visualizations include ROC curves, approval rate plots, profit curves, and profit heatmaps.

1.8 Initial Results

The initial baseline experiment uses a logistic regression classifier with scaling and one hot encoding. On the validation split, the baseline model achieves a ROC AUC around 0.90 to 0.91 and an accuracy around 0.85. The ROC curve for this baseline is shown in Figure 18, and the corresponding confusion matrix is shown in Figure 19. Recall for the positive approved class is reasonably high, but precision is more modest, indicating a tendency to over predict approvals when optimized primarily for ranking quality.

These preliminary findings suggest that while the linear model captures meaningful structure in the data, more expressive models such as Random Forest and XGBoost have the potential to yield better precision recall trade offs, especially for the minority class.

2 Problem Formulation

2.1 Key Definitions

Let $X \subset \mathbb{R}^d$ denote the feature space and $Y = \{0, 1\}$ the label space, where $y = 1$ corresponds to an approved loan and $y = 0$ to a rejected loan. Each observation consists of:

- a feature vector $x = (x_{\text{num}}, x_{\text{cat}})$, where x_{num} contains numerical attributes such as income, loan amount, interest rate, credit history length, and loan to income ratio, and x_{cat} contains categorical attributes such as loan intent, loan grade, home ownership, and prior default;
- a binary label $y \in \{0, 1\}$ representing the loan approval outcome.

We assume the data are independently and identically distributed samples from an unknown joint distribution $P(X, Y)$.

2.2 Formal Problem Statement

The objective is to learn a function $f : X \to [0, 1]$ that maps a feature vector x to an estimate $\hat{p} = f(x)$ of $P(Y = 1 \mid X = x)$, the probability that the loan is approved given the input features. At decision time, a threshold $\tau \in (0, 1)$ is applied to convert probabilities into hard approval decisions:

$$\hat{y} = \begin{cases} 1 & \text{if } f(x) \geq \tau, \\ 0 & \text{otherwise.} \end{cases}$$

We train f by minimizing a classification loss, such as cross entropy, on the training data, with class weighting to mitigate label imbalance. Evaluation focuses on performance metrics that account for ranking quality (ROC AUC) and class specific performance (precision, recall, F1 score), given the asymmetric costs associated with false positives and false negatives. In our cost sensitive layer, we explicitly model these costs to choose thresholds that maximize expected profit.

We make the following assumptions:

- The training data are representative of future loan applications, with no substantial distribution shift.
- All features are observable at decision time and do not depend on the model output.
- The primary objective is predictive performance and risk alignment; fairness and regulatory constraints are left to future work.

3 Overview of Proposed Approach

Our approach follows a standard data mining workflow adapted to the specifics of the loan approval task.

Exploratory Data Analysis. We begin by examining the class distribution, missing value patterns, and feature target relationships. The dataset exhibits clear class imbalance: approved loans comprise about 14% of the training data (Figure 1).

Histograms of numerical features and bar plots of categorical features reveal skewness in financial variables and uneven usage of some loan intents and grades (Figures 3 to 13). A correlation heatmap highlights that income and loan to income ratio are strongly associated with approval (Figure 17).

Preprocessing and pipeline design. We design a modular preprocessing pipeline using scikit learn ColumnTransformer and Pipeline abstractions. Numerical and categorical features are processed separately and the transformed feature vectors are passed to downstream classifiers. This design reduces the risk of data leakage and ensures that hyperparameter tuning and cross validation apply consistently to both preprocessing and modeling steps.

Baseline Models and Comparative Evaluation. We establish a logistic regression baseline and then systematically compare a diverse set of models. All models are wrapped inside the preprocessing pipeline and evaluated under identical splits and metrics.

Hyperparameter tuning and ensemble construction. We identify Random Forest and XGBoost as the strongest individual models, and perform grid search to tune their key hyperparameters. We then construct a soft voting ensemble combining tuned Random Forest, tuned XGBoost, and logistic regression with specified weights. ROC curve comparisons show that this ensemble provides a competitive and robust trade off across metrics (Figure 22).

Cost sensitive and segmented extensions. To better align with business objectives, we construct a stacked cost sensitive ensemble that optimizes a profit based objective by sweeping decision thresholds (Figure 23). We further experiment with a segment specific expert ensemble that splits borrowers based on loan percent income and uses specialized models and thresholds in each segment, analyzing the resulting profit surface (Figure 24).

This pipeline provides a technically sound and practically motivated framework for loan approval prediction.

4 Technical Details

4.1 Feature Engineering

We first remove unique identifier columns that carry no predictive information and may encourage overfitting. The remaining features are partitioned into numerical and categorical subsets.

Numerical features include:

- `person_age` (applicant age),
- `person_income` (annual income),
- `loan_amnt` (requested loan amount),
- `loan_int_rate` (loan interest rate),
- `person_emp_length` (employment length in years),
- `cb_person_cred_hist_length` (credit history length),
- `loan_percent_income` (ratio of loan amount to income).

Histograms of these variables (Figures 3 to 9) show substantial skewness, especially for income, loan amount, and

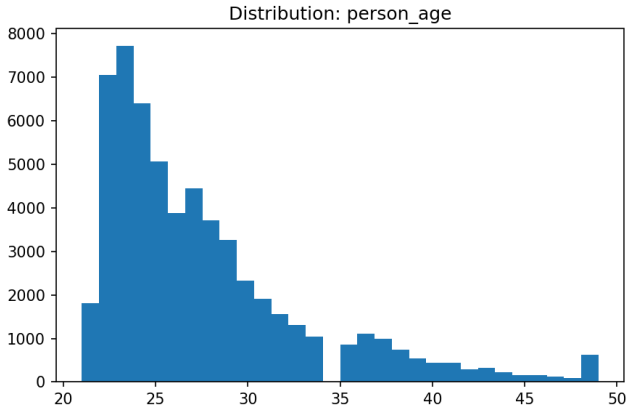

Figure 3. Distribution of applicant age.

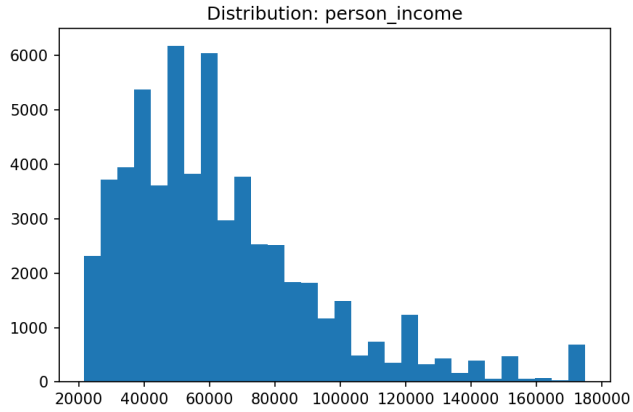

Figure 4. Distribution of annual income.

loan percent income. To reduce the influence of extreme outliers, we clip each numeric feature to the first and ninety ninth percentiles when plotting and apply scaling in the modeling pipeline.

The engineered ratio feature `loan_percent_income` is particularly informative. A boxplot of this feature by loan status (Figure 14) reveals that rejected applicants tend to request a larger fraction of their income than approved applicants. This supports the inclusion of this engineered feature in all models.

Categorical features include `loan_intent`, `loan_grade`, `person_home_ownership`, and `cb_person_default_on_file`. Bar plots of these variables (Figures 10 to 13) show that some categories are more common than others and that the population is skewed towards certain intents and grades.

We also examine label conditioned statistics. Approval rate by loan grade (Figure 15) shows that higher quality grades A and B have higher approval probabilities than lower grades. Approval rate by loan intent (Figure 16) shows that some intents such as education or home improvement appear to be treated differently from others such as venture or medical. The correlation heatmap for numerical features and the target (Figure 17) confirms that higher income and lower loan percent income are associated with higher approval likelihood, while high interest rate and short credit history may be associated with rejections.

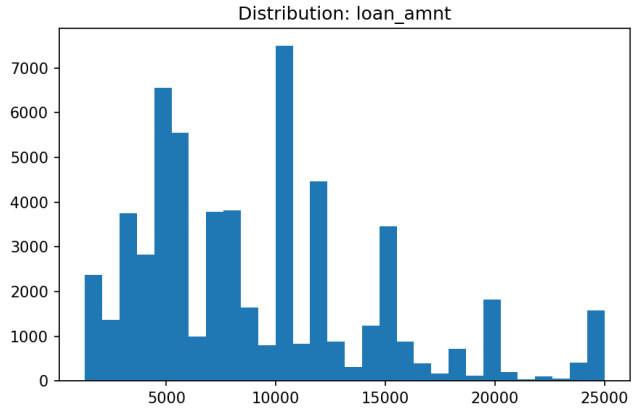

Figure 5. Distribution of requested loan amount.

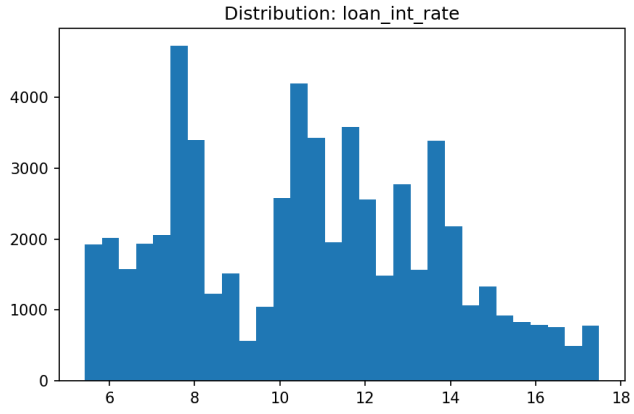

Figure 6. Distribution of loan interest rate.

4.2 Preprocessing Pipeline

We implement preprocessing using a ColumnTransformer with two branches.

Numeric branch. A `SimpleImputer` with median strategy handles missing values, followed by `StandardScaler` to standardize features.

Categorical branch. A `SimpleImputer` with most frequent strategy handles missing categories, followed by OneHotEncoder with `handle_unknown="ignore"` to create dummy variables while maintaining robustness to unseen categories.

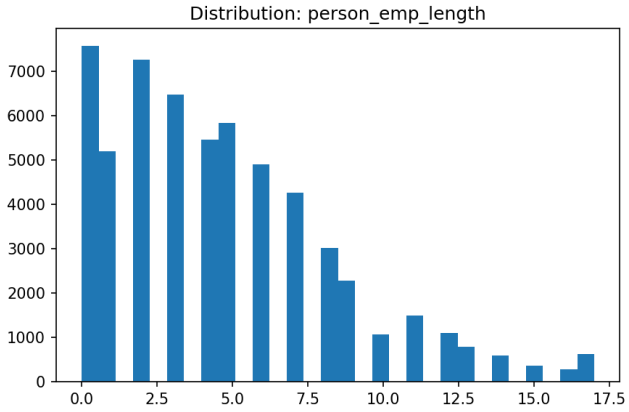

Figure 7. Distribution of employment length.

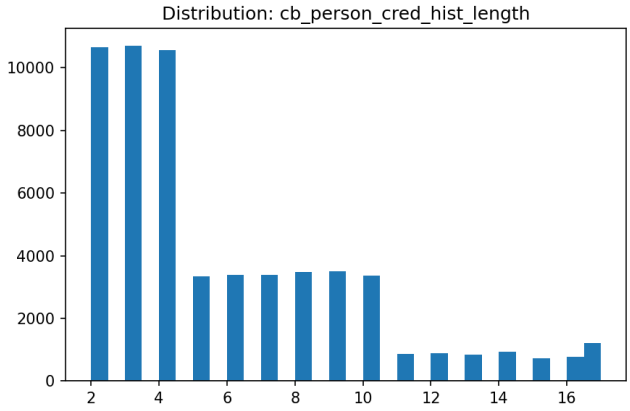

Figure 8. Distribution of credit history length.

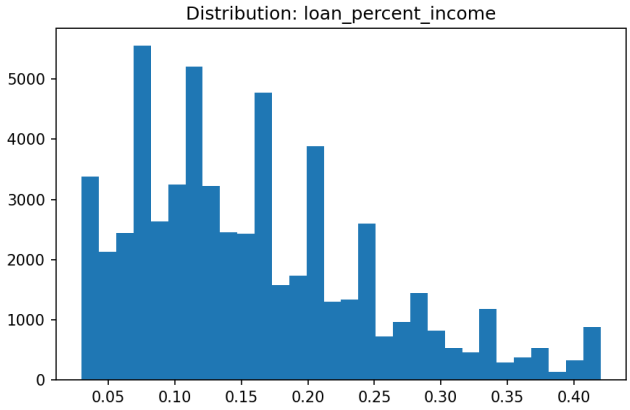

Figure 9. Distribution of loan percent income.

These branches are composed into a single transformer and combined with a classifier in a scikit-learn `Pipeline` with two steps: `"preprocess"` for the feature transformations and a `"clf"` step, which holds the final classifier model (e.g., logistic regression, Random Forest, or XGBoost).

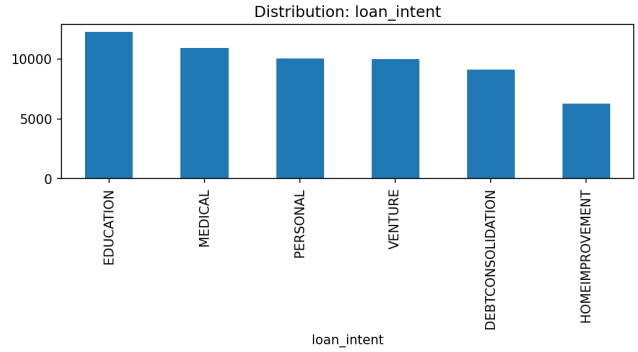

Figure 10. Distribution of loan intent categories.

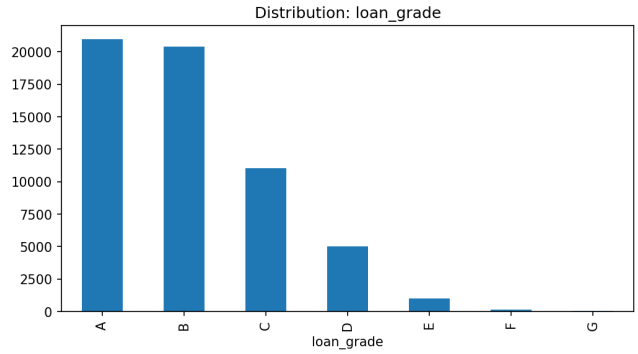

Figure 11. Distribution of loan grade categories.

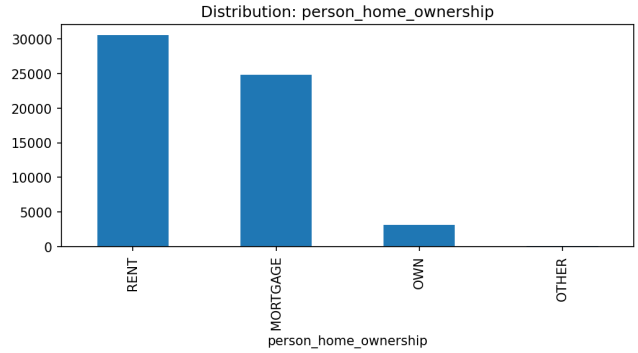

Figure 12. Distribution of home ownership categories.

This design ensures that all preprocessing steps are fitted only on the training folds during cross validation, preventing information leakage from validation data.

4.3 Predictive Modeling and Hyperparameter Tuning

We investigate a range of classifiers, all wrapped in the same preprocessing pipeline:

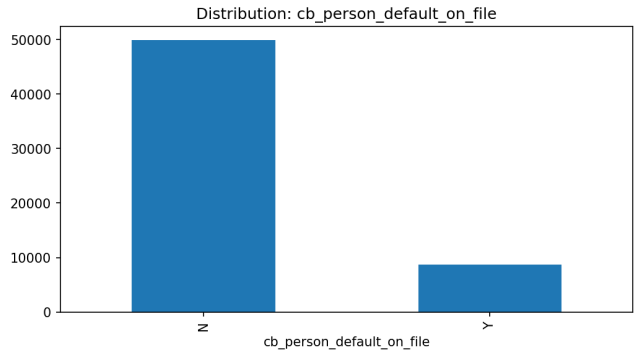

Figure 13. Distribution of prior default indicator.

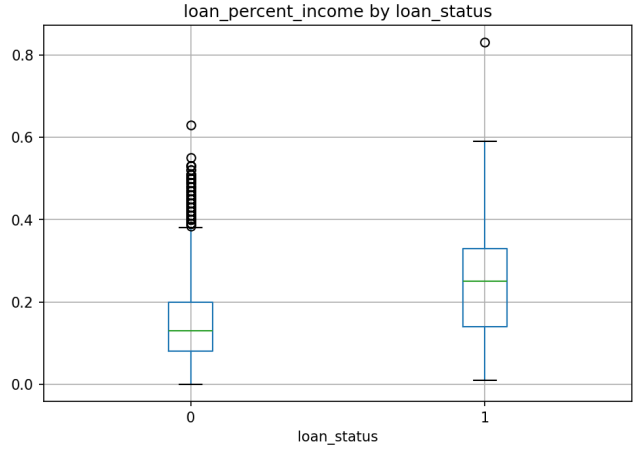

Figure 14. Loan percent income by approval status.

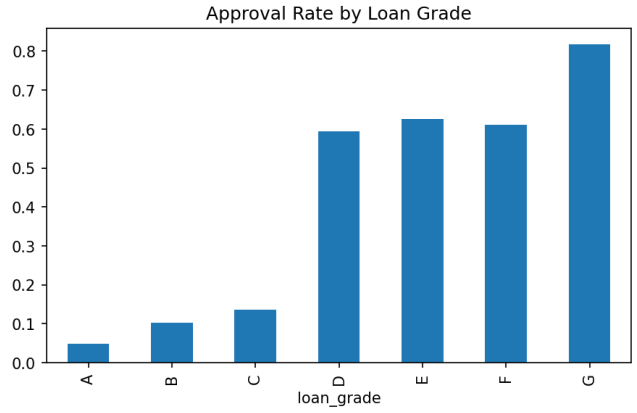

Figure 15. Approval rate by loan grade.

- **Logistic Regression (LR).** Linear baseline with ℓ_2 regularization and `class_weight="balanced"`. This model is interpretable and serves as a benchmark.

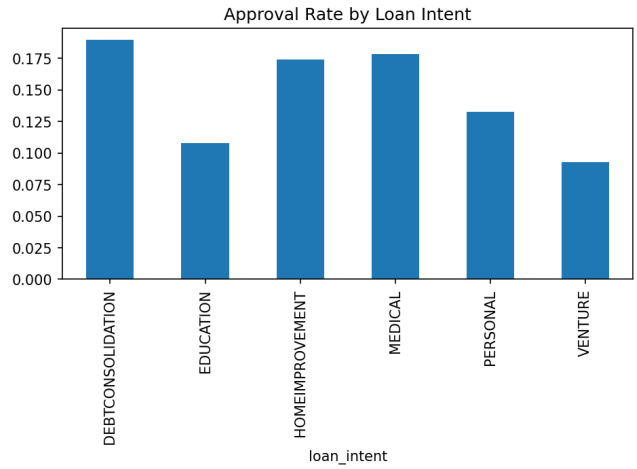

Figure 16. Approval rate by loan intent.

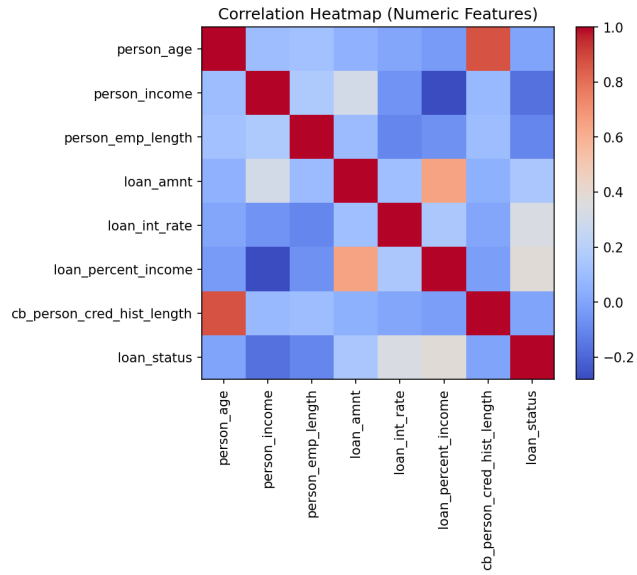

Figure 17. Correlation heatmap of numeric features and `loan_status`.

- **Gaussian Naive Bayes (GNB).** Assumes conditional independence and Gaussian class conditional distributions.
- **k Nearest Neighbors (kNN).** Uses Euclidean distance in the transformed feature space.
- **Support Vector Machine (SVM).** Kernel SVM with an RBF kernel and class weighting.
- **Decision Tree (DT).** Nonlinear model with interpretable paths, regularized by depth and leaf constraints.
- **Random Forest (RF).** Bagging ensemble of decision trees with class weighting.

- **AdaBoost (AB).** Boosting ensemble of shallow trees that focuses on difficult examples.
- **Multilayer Perceptron (MLP).** Small feed forward neural network trained with cross entropy loss and early stopping.
- **XGBoost (XGB).** Gradient boosted tree ensemble with regularization, subsampling, and fast training for tabular data.

For each model, we fit a pipeline on the training split and evaluate on the validation split in terms of ROC AUC and classification metrics. For Random Forest and XGBoost, we perform grid search with 3 fold stratified cross validation using ROC AUC as the scoring metric. Example grids include:

Random Forest grid. `n_estimators` in $\{100, 200\}$, `max_depth` in $\{10, 20\}$, `min_samples_leaf` in $\{2, 4\}$.

XGBoost grid. `n_estimators` in $\{100, 200\}$, `max_depth` in $\{3, 5\}$, `learning_rate` in $\{0.05, 0.1\}$, and `scale_pos_weight` set based on class imbalance.

The best hyperparameters for RF and XGB are retained for subsequent ensemble models.

4.4 Ensemble Frameworks and Novel Components

We design three ensemble layers.

Soft Voting Ensemble. We combine tuned XGBoost, tuned Random Forest, and logistic regression in a `VotingClassifier` with soft voting and weights favoring RF and XGB. This ensemble targets improved robustness and balanced precision recall.

Stacked Cost Sensitive Ensemble. We define a StackingClassifier whose base estimators include logistic regression, tuned Random Forest, tuned XGBoost, and MLP. The meta learner is logistic regression trained on predicted probabilities. On top of this stacked model, we introduce a cost sensitive threshold sweep. We specify asymmetric costs: approve good loan (true positive): gain +1.0, approve bad loan (false positive): loss -5.0, reject good loan (false negative): loss -0.5, reject bad loan (true negative): gain 0.0. We sweep the decision threshold from 0.01 to 0.99 and compute expected profit for each threshold using the validation set. The resulting profit curve is shown in Figure 23.

Segment Specific Expert Ensemble. We design a segment specific expert ensemble that splits borrowers based on the median `loan_percent_income` into low ratio and high ratio segments. For each segment, we fit a specialized pipeline, using tuned Random Forest for the low ratio segment and tuned XGBoost for the high ratio segment. At prediction time, each applicant is routed to the appropriate expert and receives a segment specific probability. We then search over two thresholds, one for the low ratio segment and one for the high ratio segment, using the same cost sensitive profit function. The profit surface over pairs of thresholds is visualized in Figure 24.

These ensemble layers constitute the main technical novelty of the project. Rather than relying solely on off the shelf models, we design a cost aware and segment aware ensemble framework tailored to the loan approval task.

Table 1. Dataset summary statistics.

Quantity	Value	Notes
Training instances	58,645	Full training split
Test instances	39,098	Kaggle competition test set
Positive class rate	0.142	8,350 / 58,645 approved
Numeric features	7	After feature engineering
Categorical features	4	Before one-hot encoding

5 Experimental Evaluation

5.1 Dataset Description

Table 1 summarizes key dataset properties such as total observations, class balance, and feature types. Approved loans constitute about 14% of the training data (8,350 approved vs. 50,295 rejected), so the positive class is a clear minority (Figure 1), motivating the use of stratification and class weighting. The missing-value analysis (Figure 2) confirms that there are no missing values in the provided training data, so imputation is not strictly required but is retained in the pipeline as a safeguard.

5.2 Evaluation Metrics

We use the following metrics:

- **Accuracy:** overall fraction of correctly classified instances.
- **Precision (positive class):** fraction of predicted approvals that are actually approved.
- **Recall (positive class):** fraction of truly approved loans that are correctly predicted as approved.
- **F1 score:** harmonic mean of precision and recall for the positive class.
- **ROC AUC:** area under the ROC curve, measuring how well the model ranks approved loans ahead of rejected loans.
- **Expected profit:** profit obtained under the business inspired cost model, as a function of decision thresholds.

Given the business context, both ROC AUC and F1 score are important: false positives approving risky loans and false negatives rejecting good borrowers have asymmetric costs. The profit based evaluation overlays a more application specific perspective on top of these generic metrics.

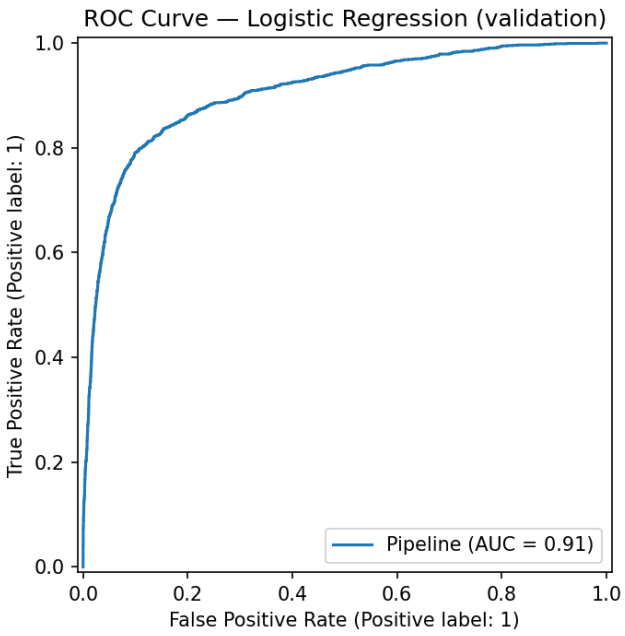

Figure 18. ROC curve for logistic regression on the validation split.

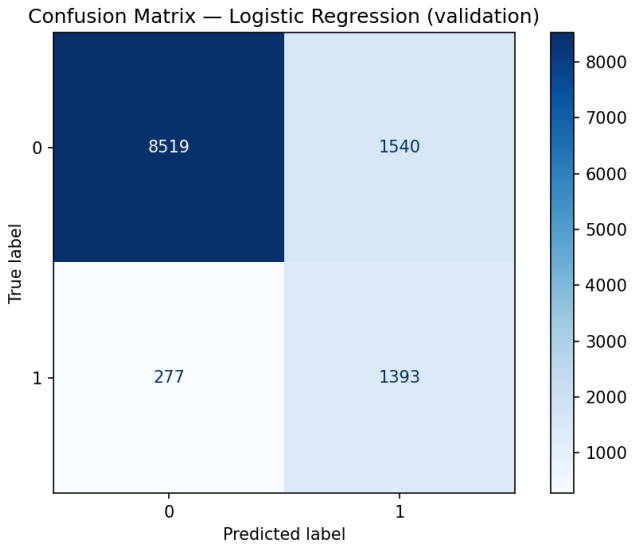

Figure 19. Confusion matrix for logistic regression on the validation split.

5.3 Baseline Methods

We designate logistic regression as the main baseline model, reflecting common practice in credit scoring. In addition, we treat other standard classifiers as candidate baselines for comparison. A comprehensive baseline model comparison is saved in a CSV file containing ROC AUC, F1, precision, recall, and accuracy for each model on the validation split.

5.4 Results and Analysis

Baseline logistic regression. The logistic regression model, using all engineered and processed features, achieves a ROC AUC around the low 0.90s and a respectable recall for approved loans. Its ROC curve is shown in Figure 18, and the confusion matrix is shown in Figure 19. Coefficient analysis, summarized in Figures 20 and 21, confirms that lower loan percent income, higher income, and absence of prior default are associated with higher approval odds.

Tree based and boosting models. Random Forest, AdaBoost, and XGBoost all outperform logistic regression in ROC AUC and F1 score. XGBoost often achieves the highest ROC AUC among baseline models, while Random Forest attains a very competitive F1 score. These results indicate that nonlinear ensembles capture important interactions between features, such as combinations of income, loan grade, and prior defaults.

Hyperparameter tuning and soft voting ensemble. Grid search for Random Forest and XGBoost yields tuned configurations that further improve ROC AUC and F1 score relative

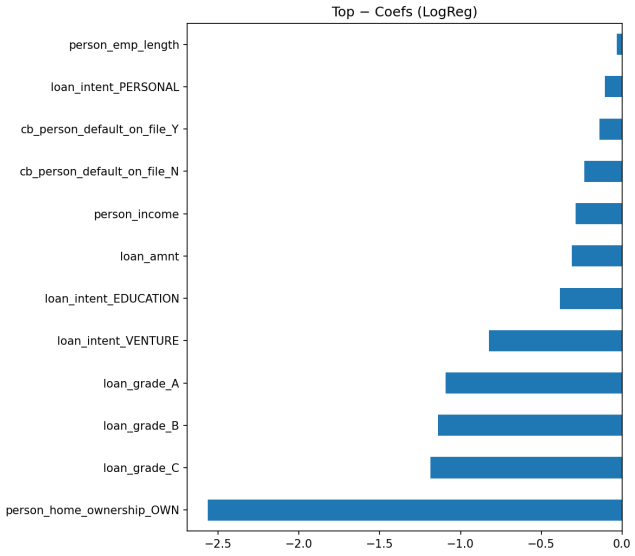

Figure 20. Most negative logistic regression coefficients.

to default settings. We then construct a soft voting ensemble of tuned RF, tuned XGB, and LR. A consolidated ROC curve comparison (Figure 22) shows the logistic regression curve as a strong baseline, tuned XGBoost lying above LR across most of the ROC space, tuned Random Forest performing particularly well at moderate false positive rates, and the soft voting ensemble achieving a ROC AUC close to tuned XGBoost while slightly improving F1 and offering a robust trade-off across metrics.

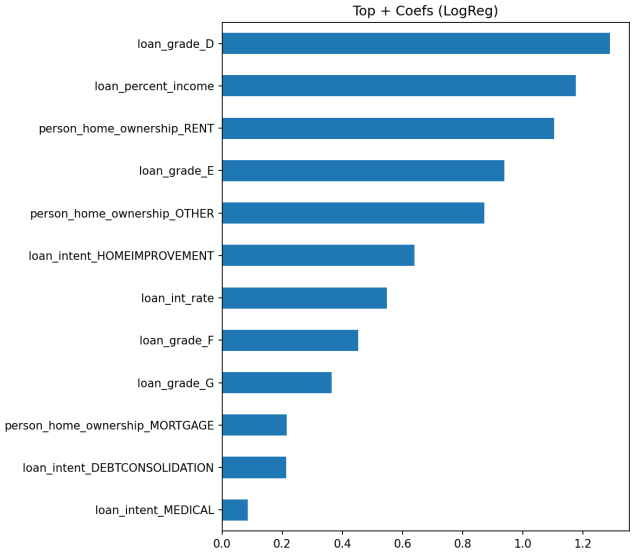

Figure 21. Most positive logistic regression coefficients.

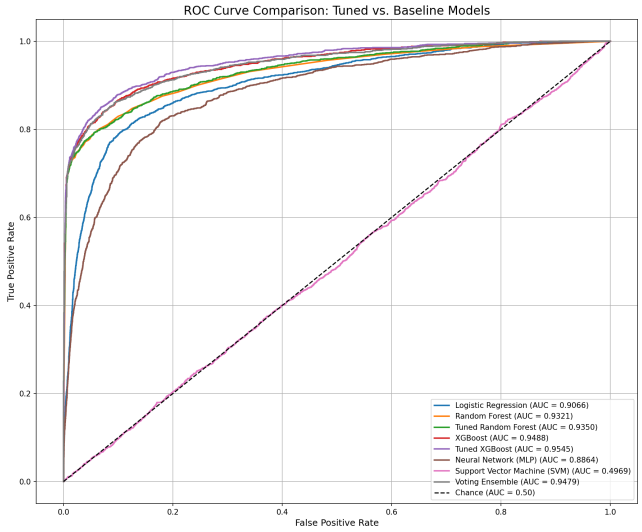

Figure 22. ROC curve comparison for logistic regression, tuned Random Forest, tuned XGBoost, and the soft voting ensemble.

Table 2. Baseline performance of candidate models on the validation split.

Model	ROC AUC	F1 score
Logistic Regression	0.9062	0.6057
Gaussian Naive Bayes	0.8862	0.6043
k Nearest Neighbors	0.5503	0.0556
Support Vector Machine	0.4969	0.2161
Decision Tree	0.8258	0.6986
Random Forest	0.9321	0.8037
AdaBoost	0.9147	0.6990
Multilayer Perceptron	0.7643	0.3466
XGBoost	0.9488	0.7698

Table 3. Tuned models and ensemble performance on the validation split.

Model	ROC AUC	F1 score
Tuned Random Forest	0.9350	0.7961
Tuned XGBoost	0.9545	0.7642
Voting Ensemble	0.9479	0.7784
Logistic Regression	0.9062	0.6057

The tuned Random Forest is an attractive choice when precision and interpretability via feature importances are prioritized, while tuned XGBoost is ideal for maximizing ROC AUC and ranking. The voting ensemble offers a balanced compromise suitable as a default deployment model.

Cross validation stability. We run 5 fold stratified cross validation for logistic regression and XGBoost using ROC AUC as the metric. The mean ROC AUC scores across folds show small standard deviations, indicating that the performance estimates are stable and not overly sensitive to a particular train validation split.

5.5 Cost Sensitive Stacked Ensemble

The stacked ensemble model, composed of LR, RF, XGB, and MLP base learners with a LR meta learner, shows strong ROC AUC comparable to the best individual models. The key contribution is the threshold selection driven by the profit function. The profit threshold curve in Figure 23 reveals a broad plateau where expected profit is near its maximum and a specific threshold that achieves the highest profit under our assumed cost ratios. Compared to the default threshold of 0.5, this optimal threshold typically reduces the number of false positives at the expense of some additional false negatives.

5.6 Segment Specific Expert Ensemble

The segment specific expert ensemble recognizes that borrowers with very different loan to income ratios may warrant different approval rules. By splitting the data into low and high loan percent income segments and training specialized RF and XGB experts, we can assign separate approval thresholds to each group. The profit surface over pairs of thresholds (Figure 24) shows that high risk applicants benefit from stricter thresholds to limit false positives, while low risk applicants can be treated with more lenient thresholds, improving recall among borrowers likely to repay. The best threshold pair yields higher expected profit than any single global threshold applied to the full population.

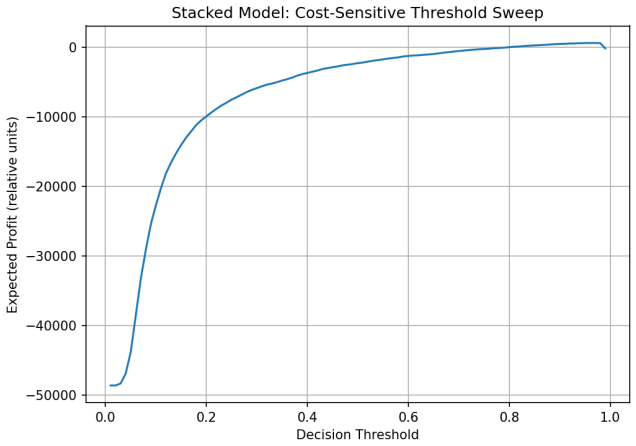

Figure 23. Expected profit as a function of decision threshold for the stacked ensemble.

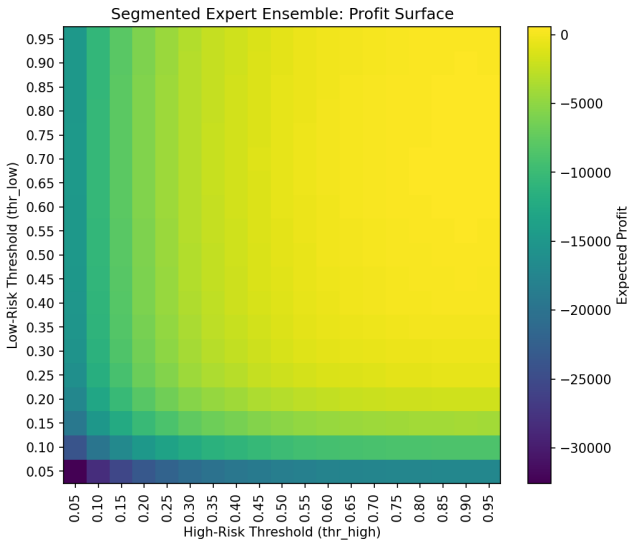

Figure 24. Profit surface over thresholds for low and high loan percent income segments.

5.7 Ablation and Component Wise Evaluation

We perform several component wise comparisons.

- **Model complexity.** Comparing logistic regression with RF and XGB demonstrates that nonlinear ensembles significantly improve F1 score and ROC AUC, underscoring the value of capturing feature interactions.
- **Hyperparameter tuning.** Tuned RF and XGB consistently outperform their untuned counterparts, especially in ROC AUC, highlighting the importance of systematic hyperparameter search.
- **Ensembling.** The soft voting ensemble and stacked ensemble provide more stable performance across metrics and splits than any single model.
- **Feature importance.** Experiments with and without `loan_percent_income` show a degradation in ROC AUC and F1 score when this feature is removed, confirming its central role in discrimination between approved and rejected loans.
- **Cost sensitive versus metric based decisions.** The cost sensitive approaches shift thresholds relative to those that would be chosen solely to maximize F1 or accuracy, emphasizing how business costs can and should influence operational decision rules.

6 Conclusion

This project presented a complete data mining pipeline for loan approval prediction using a realistic bank loan dataset. We formulated the task as a binary classification problem, conducted detailed exploratory data analysis, and designed a principled preprocessing and modeling framework that handles mixed feature types and missing data.

Starting from a logistic regression baseline, we evaluated a wide range of models and found that tree based ensembles and gradient boosting methods substantially outperformed the linear baseline in terms of ROC AUC and F1 score. Hyperparameter tuning further improved performance, and a soft voting ensemble of tuned Random Forest, tuned XGBoost, and logistic regression provided a robust and well balanced precision recall trade off.

Beyond these standard techniques, we introduced a stacked cost sensitive ensemble and a segment specific expert ensemble. These frameworks incorporate explicit business cost assumptions and borrower segmentation into the decision rule, moving closer to the way real lending policies are crafted. The cost sensitive threshold selection and segment specific thresholds both yielded improvements in expected profit over naive global thresholding.

For future work, several directions are promising: incorporating resampling methods such as SMOTE or more advanced cost sensitive algorithms, applying interpretability tools such as SHAP values and partial dependence plots, exploring temporal and cohort based evaluation to assess model stability across time and subpopulations, and integrating the trained model into a prototype decision support interface with real time scoring and performance monitoring.

Team Member Responsibilities

Table. Team member responsibilities.

Member	Responsibilities
Mohammad Hamza Choudhry	Defined the project methodology for the loan approval task, including the problem formulation, evaluation criteria, and list of candidate models; led the initial exploratory data analysis and prepared the core EDA visualisations; designed the comparative analysis across nine classifiers by defining the set of algorithms to compare and implemented their initial pipelines; extended the strongest models with soft voting, stacked, and segment specific cost aware ensemble decision rules; led the drafting and editing of the final report.
Ryan Sam Varghese	Developed the preprocessing pipeline based on `ColumnTransformer` (imputation, scaling, and one hot encoding) and applied it across baseline and ensemble models; set up stratified train/validation splits and cross validation routines; developed metric reporting and comparison scripts; and maintained the main experiment code, including debugging and refactoring for a consistent and reproducible workflow.
Sai Gangaraju	Implemented and tuned Random Forest and XGBoost using cross validated grid search; built the soft voting ensemble and contributed to the stacked model and threshold search under the cost model; produced performance tables and ROC curves for checkpoint two and the final report; and assisted in analysing feature importances and model behaviour.
Aswath Sridhar	Executed and monitored model training runs; collected and organised result files, including baseline summaries and hyperparameter tuning logs; created visualisations for model performance and decision analysis, such as ROC curves, confusion matrices, profit curves, and approval rate plots; and helped prepare figures for the presentations and final report, as well as updating comments and documentation in the codebase.

Code Repository

The code used in this project is available at:
https://drive.google.com/file/d/1jxDnBMV_cwqt4FkMKAltuZJkz2FS6Lnb/view?usp=sharing

Acknowledgments

This project was completed as part of CSE 572 Data Mining at Arizona State University.

References

[1] Leo Breiman. Random forests. *Machine Learning* 45(1):5–32, 2001.
[2] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29(5):1189–1232, 2001.
[3] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
[4] Fabian Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830, 2011.
[5] Kaggle. Bank Loan Approval Prediction Dataset. Online competition and dataset description, accessed 2025.
[6] D. J. Hand and W. E. Henley. Statistical classification methods in consumer credit scoring. *Journal of the Royal Statistical Society, Series A*, 160(3):523–541, 1997.
[7] S. Lessmann, B. Baesens, H. Seow, and L. C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring. *European Journal of Operational Research*, 247(1):124–136, 2015.
[8] I. Brown and C. Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3):3446–3453, 2012.
[9] T. Verbraken, C. Bravo, R. Weber, and B. Baesens. Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research*, 238(2):505–513, 2014.
[10] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017.