University Of San Diego

Shiley Marcos School of Engineering

ADS 505 Final Project

Group 3:

Tarane Javaherpour

Mauricio Espinoza Acevedo

"Loan Approval Prediction Using Machine Learning"

Loan Approval Prediction

Problem Statement

The objective of this project is to develop a predictive model that determines whether a loan applicant should be approved or denied. The dataset consists of various applicant features such as income, loan amount, credit history, employment status, and other financial indicators. The goal is to predict a binary outcome: loan approval or loan rejection, based on an applicant's profile. Loan approval decisions are critical for financial institutions as they tr5rt. An accurate and reliable model can help streamline the approval process, reduce the risk of defaulting on a loan, and improve decision-making efficiency. Through machine learning models, banks and lending institutions can improve their ability to assess the risk associated with each applicant and make more informed lending decisions.

We will explore and compare multiple machine learning models, including logistic regression, random forest, and XGBoost. Each model will be tuned and evaluated based on several performance metrics, such as accuracy, precision, recall, F1-score, and confusion matrices. These models will be evaluated and compared based on their ability to predict loan approval with high accuracy while minimizing the risk of misclassification.

Data Exploration

In this step, we examine the relationships between the response variable (customer purchase behavior) and the predictor variables (customer features). Graphical representations such as histograms and boxplots were used to assess the distribution of continuous features as well as for identifying and discarding outliers (figure 1). For example, features such as loan amount might exhibit skewed distributions or contain outliers which should be handled accordingly in the pre-processing phase (figure 2). Visualizing these distributions helped in identifying potential data transformations, such as scaling for logistic regression. Categorical variables were analyzed using bar charts to understand the frequency distribution of different categories and their relationship with the response variable. Additionally, correlation heatmaps

were created to identify relationships between variables. This is especially useful for identifying multicollinearity, which can negatively impact linear models like logistic regression.

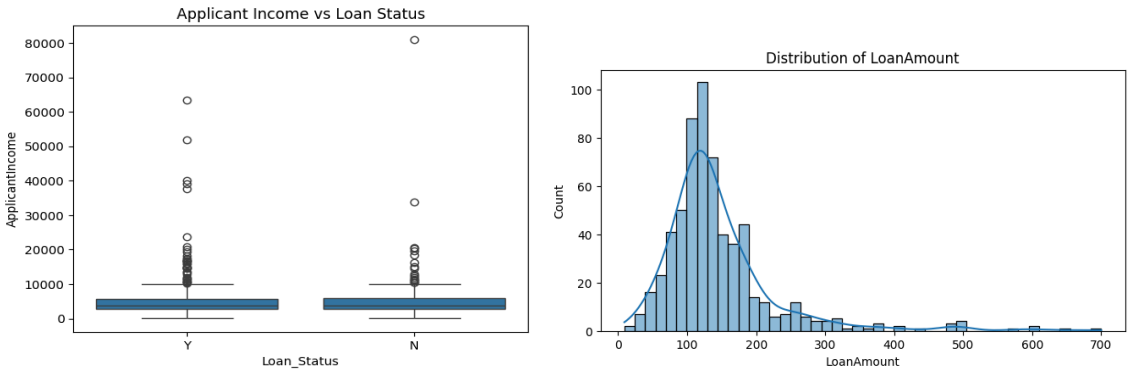

Figure 1&2. Examples of outlier identification and observation of data distribution in EDA.

Data Wrangling and Pre-processing

In any machine learning task, data preprocessing is a crucial step. Missing values, outliers, and scaling were handled systematically to ensure the data was in a suitable format for modeling. Missing values were identified and mode imputation was used to hold the missing values. Mode imputation was used for categorical variables while median imputation was used for continuous and skewed variables in order to preserve the shape and distribution of the data. Additionally, the number of missing variables in each column was small enough for mode imputation to adequately deal with missing values without unnecessarily complicating things. Furthermore, outliers can also distort model performance, particularly for models like logistic regression. Outliers were detected using boxplots or statistical measures such as the interquartile range (IQR), once identified, outliers were removed accordingly.

Logistic regression requires feature scaling to ensure that all variables contribute equally to the model. Continuous variables were standardized to have a mean of 0 and a standard deviation of 1. This ensures that features on different scales (e.g., age vs. income) are treated equally by the model. However, tree-based models like random forest and XGBoost do not

require feature scaling, as they are not sensitive to the magnitude of the input features. If the dataset contained a large number of categories for a variable, dummy encoding might have been used to prevent multicollinearity issues, however, those issues did not arise with our dataset.

Data splitting

To ensure a robust evaluation of the models, the dataset was split into training and validation sets. This process is essential for assessing the performance of the model on unseen data, preventing overfitting, and ensuring that the model generalizes well to new loan applicants. The dataset contains both the predictor variables and the target variable (loan status), which indicates whether a loan was approved or denied. To prepare the data for modeling, the features and target variable were first separated. The target variable was removed from the feature set, leaving only the independent variables that will be used to predict loan approval.

After separating the features and target, the “train” dataset (df_train) was split into training and validation sets using an 80-20 split. This means that 80% of the data was used to train the model, while the remaining 20% was reserved for validation. The *train_test_split* function from the *sklearn.model_selection* library was used to perform this split, with the *random_state* parameter set to 42 for reproducibility. This ensures that the same split is produced every time the code is run, enabling consistent results.

The training set (*X_train, y_train*) was used to fit the machine learning models, while the validation set (*X_val, y_val*) was used to evaluate model performance during development. By keeping the validation data separate, we ensured that model tuning and selection were based on how well the models generalize to new data, rather than their ability to fit the training data. This is crucial for preventing overfitting, where a model performs well on training data but poorly

on new, unseen data. The final models were later evaluated on an entirely unseen test (df_test) set to further assess its real-world performance.

Modeling

Model strategies

The primary research question is can a machine learning model predict loan approvals with high accuracy, thereby aiding financial institutions in making efficient and data-driven decisions? To address this, we tested various models to determine which would provide the highest predictive performance while minimizing the risk of misclassification. We explored multiple machine-learning techniques for building the predictive model, specifically logistic regression, random forest, and XGBoost. Logistic regression was used as a baseline to assess the linear separability of the data. It provided insights into the influence of individual features, such as Credit History and Applicant Income, on the likelihood of loan approval. Random Forest was utilized for its ensemble learning capabilities, which help reduce overfitting and enhance generalization. The Random Forest model provided an advantage by averaging multiple decision trees, thereby reducing the risk of high variance. XGBoost was implemented to take advantage of its gradient boosting and hyperparameter tuning capabilities for improved accuracy. XGBoost's ability to handle missing data internally made it particularly useful for our dataset, which contained several missing values.

We used three machine learning models: Random Forest Classifier, Logistic Regression, and XGBoost Classifier. Each model was evaluated to determine its predictive accuracy and reliability.

Validation and testing (model tuning and evaluation)

To ensure optimal model performance, hyperparameter tuning was conducted using cross-validation. Logistic Regression had its regularization strength tuned using grid search, with performance measured using accuracy and recall metrics. The goal was to balance the simplicity of the model with its ability to generalize to unseen data. Random Forest's hyperparameters, such as the number of estimators, maximum depth, and minimum sample split, were optimized using `GridSearchCV`. By limiting the maximum depth and adjusting the minimum samples per split, we aimed to reduce overfitting while maintaining model accuracy. XGBoost was tuned with parameters such as `n_estimators`, `max_depth`, and `learning_rate` using a grid search approach to find the combination that resulted in the best performance on the validation set.

Confusion matrices were generated for each model to visualize the distribution of predictions and identify the number of true positives, false positives, true negatives, and false negatives. This helped in understanding the model's strengths and weaknesses in predicting loan approvals versus rejections. Classification reports including precision, recall, and F1 scores were evaluated for each model to provide insights into how well the models classified approved and rejected loans. Accuracy scores of all models were compared to identify the best-performing model, focusing on minimizing false approvals to mitigate lenders' financial risk. The performance of the training models reveals notable differences in predicting loan approvals and rejections. The Decision Tree model, with an accuracy of 71.54%, had a relatively balanced recall for both approvals ("yes") and rejections ("no"). However, it struggled with precision when predicting rejections, indicating that many predicted rejections were actually approvals. XGBoost, which achieved an accuracy of 76.42%, excelled in predicting approvals, with a high recall of 93% for "yes" decisions. However, its ability to accurately predict rejections was

moderate, with a precision of 77% for "no" decisions. The Random Forest model slightly outperformed the other two with an accuracy of 77.24%. It was particularly strong at predicting rejections, with the highest precision for "no" decisions at 86%. However, its recall for rejections was lower, meaning it missed identifying some rejections. Overall, Random Forest showed the best precision for predicting rejections, while XGBoost offered a more balanced approach with strong performance in predicting approvals.

Hyperparameter Tuning:

XGBoost was tuned with parameters such as `n_estimators`, `max_depth`, and `learning_rate`. This was done using a grid search approach to find the combination that resulted in the best performance on the validation set.

Results and final model selection

The Logistic Regression model achieved an accuracy of 78.86%. The model's simplicity provided good interpretability but could not capture complex relationships in the data. The Random Forest model achieved an accuracy of 80.65% with balanced performance across precision and recall. The model's ensemble nature reduced variance and provided insights into feature importance. The XGBoost model achieved an accuracy of 76.42%. With tuned hyperparameters, it demonstrated the best generalization capability and handled class imbalances effectively.

Based on accuracy, recall, and generalization performance, Random Forest was selected as the final model for deployment. Although Logistic Regression and XGBoost offered comparable

performance, Random Forest provided a balance between interpretability and accuracy, with fewer false approvals, making it suitable for deployment in a financial context where interpretability and minimizing risk are critical. Feature importance analysis from Random Forest highlighted Credit History, Applicant Income, and Loan Amount as the most critical predictors of loan approval.

Discussion and conclusions

The comparison of four models—Random Forest, XGBoost, Logistic Regression, and Decision Tree—provides valuable insights into their respective performance in predicting loan approvals. Each model's effectiveness is evaluated using key metrics such as accuracy, precision, recall, and F1-score, which reflect their ability to correctly classify loan-worthy and non-loan-worthy individuals, minimizing both false positives (incorrect loan approvals) and false negatives (incorrect loan rejections).

The Random Forest model, using an adjusted threshold, achieved an overall accuracy of 80.65%. However, its classification performance was highly imbalanced. For class 0 (individuals who should not receive loans), it performed poorly, achieving no correct classifications (precision and recall of 0.00). This indicates that the model was unable to identify any individuals correctly as non-loan-worthy, leading to a high number of false positives. On the other hand, for class 1 (loan-worthy individuals), the model performed much better, achieving a precision of 0.81 and a recall of 1.00, meaning it identified all loan-worthy individuals correctly. Despite its excellent performance on class 1, the failure to handle class 0 significantly detracts from the model's usefulness in real-world lending scenarios. This imbalance in performance between the two classes is further illustrated by the macro-average F1-score of 0.45, indicating that while the model is very good at approving qualified applicants, it is prone to making risky approvals.

The XGBoost model, in contrast, offered a more balanced performance, achieving an overall accuracy of 84%. For class 0, it attained a precision of 0.55 and a recall of 0.83, with an F1-score of 0.66, indicating that it performs moderately well in identifying individuals who should not receive loans, though there remains room for improvement in precision. For class 1, XGBoost delivered a precision of 0.95 and a recall of 0.84, with an F1-score of 0.89, reflecting strong performance in identifying loan-worthy applicants. Compared to Random Forest, XGBoost is more balanced across both classes, and while it sacrifices some recall for class 1, it reduces the likelihood of making risky approvals by improving its performance on class 0.

The Logistic Regression model achieved an accuracy of 81.74%, with an extreme disparity between the two classes. For class 0, it achieved perfect precision of 1.00, but its recall was only 0.06, meaning it correctly identified very few individuals who should not receive loans, resulting in a high number of false positives. The F1-score for class 0 was just 0.11, underscoring the poor performance for this class. In contrast, for class 1, the model achieved a precision of 0.82, a perfect recall of 1.00, and an F1-score of 0.90, indicating that it performed exceptionally well at approving qualified loan applicants while failing to filter out high-risk individuals. The weighted F1-score of 0.75 reflects this stark imbalance. Thus, while Logistic Regression excels at minimizing false negatives, it tends to approve far too many risky individuals, making it unsuitable for scenarios where financial risk is a concern.

The Decision Tree model, with an accuracy of 82.02%, offers a compromise between the extremes seen in the other models. For class 0, it achieved a precision of 0.52 and a recall of 0.92, with an F1-score of 0.66. This high recall suggests that the Decision Tree is adept at correctly identifying individuals who should not receive loans, though the relatively low precision reflects a higher number of false positives. For class 1, the model performed more robustly, with a precision of 0.98, a recall of 0.80, and an F1-score of 0.88. The model is therefore fairly effective at identifying loan-worthy applicants while maintaining a reasonable balance between

precision and recall for non-loan-worthy individuals. Its ability to achieve a good recall for class 0, unlike Logistic Regression and Random Forest, makes it a more balanced option, though the lower precision for class 0 still leaves some room for improvement.

In conclusion, Random Forest performed very well for classifying loan-worthy individuals but struggled with non-loan-worthy applicants, making it risk-prone. XGBoost offered a more balanced performance across both classes, making fewer risky approvals compared to Random Forest. Logistic Regression, while highly effective at identifying loan-worthy applicants, had serious issues with over-approving risky individuals due to its high false positive rate. Finally, Decision Tree provided a middle ground, achieving relatively good performance across both classes, though it still exhibited some weaknesses in precision for identifying risky applicants. Ultimately, XGBoost and Decision Tree appear to offer the best trade-offs between risk minimization and approval of qualified individuals, with Random Forest being more effective in contexts where identifying qualified applicants is of greater concern than risk.