Leveraging Ensemble Learning Techniques for Enhanced Loan Repayment Prediction

NAME: PAVAN KUMAR KURVA

STUDENT ID : 22065713

GOOGLE COLAB LINK:

https://colab.research.google.com/drive/1vTI7Jc0T7n0Upxnl635Cg0wpT9alXRoY?usp=sharing

INDEX:

- ABSTRACT
- INTRODUCTION
- DATA PREPROCESSING
- MODELLING
- ROC CURVE & PR CURVE
- MODEL COMPARISION
- ENHANCEMENT and CONCLUSION
- REFERENCES

SOURCE LINK:

https://www.kaggle.com/code/kerneler/starter-loan-repayment-prediction-5057afb4-1

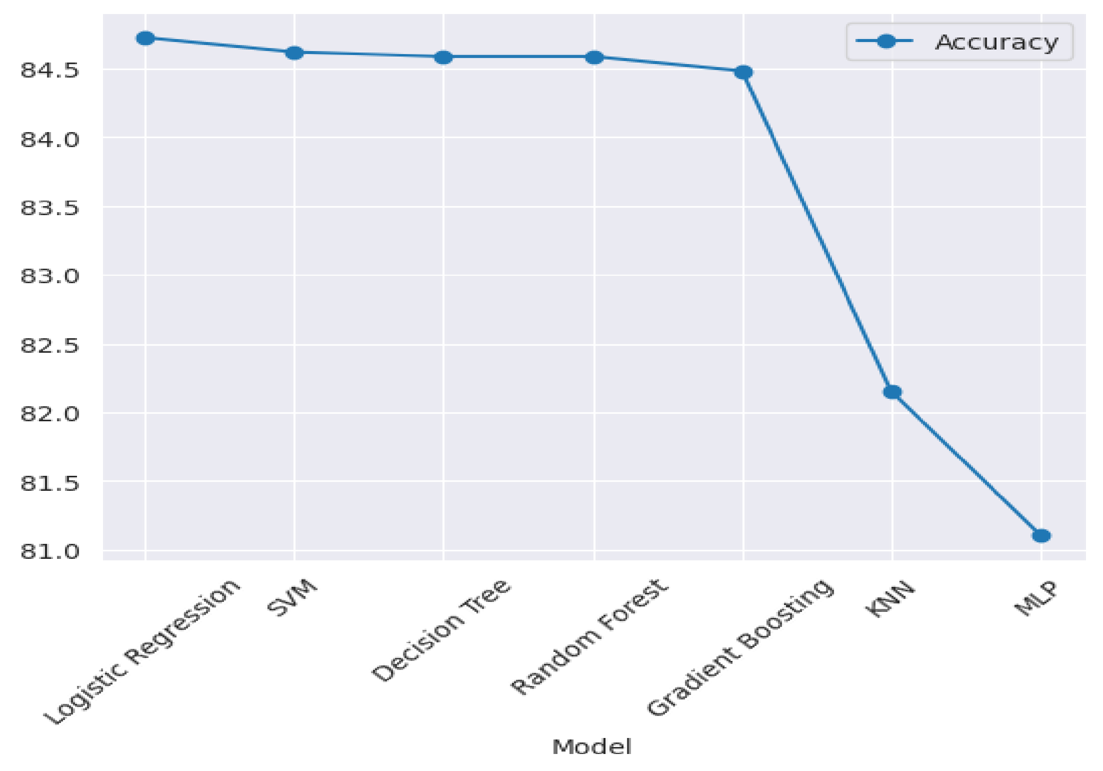
PROBLEM STATEMENT:

In the realm of lending, investors extend loans to borrowers with the expectation of receiving repayment along with interest. When borrowers fulfill their repayment obligations, lenders profit from the accrued interest. However, if borrower’s default on their loans, lenders incur financial losses. Consequently, lenders grapple with the challenge of predicting the likelihood that a borrower will be unable to repay a loan. To address this issue, we delve into an analysis using data from Kaggle. Our objective is to train various machine learning models to assess a borrower’s capacity to repay their loan. Specifically, we evaluate the performance of several models, including Random Forest, Logistic Regression, Support Vector Machine, and K Nearest Neighbors. Among these, the **LOGISTIC REGRESSION MODEL** emerges as the optimal predictive tool. Notably, we anticipate that factors such as FICO score and annual income significantly influence loan repayment forecasts.

INTRODUCTION:

Loans are pivotal products for financial institutions, driving them to devise effective strategies for attracting more applicants. However, ensuring loan repayment is challenging, leading institutions to consider various factors during approval. Determining whether borrowers will fully repay, or default is crucial but complex. Stringent approval criteria result in fewer loans but fewer defaults, while leniency increases approvals but also defaults. This study employs machine learning models to analyze loan behaviors.

The dataset, sourced from Kaggle, comprises 9,578 observations. While Logistic Regression commonly used for classification, its limitations in capturing non-linear relationships prompt exploration of alternative models such as Random Forest, MLP, KNN, and SVM. Exploratory.

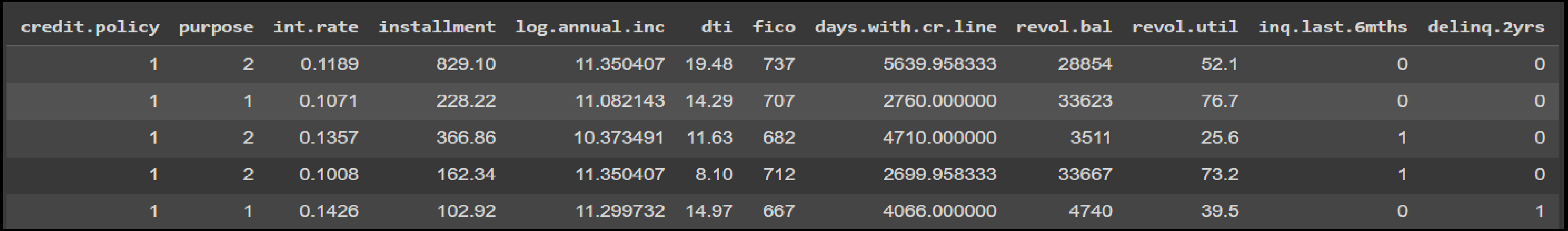


Exploratory data analysis address the missing values, followed by necessary data transformations.

Training and evaluating each model—Logistic Regression, Random Forest, KNN, and SVM—using k-fold classification techniques and confusion matrices.

DATA PREPROCESSING:

Here are the column names and their Description of the given Dataset:



credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs
1	2	0.1189	829.10	11.350407	19.48	737	5639.958333	28854	52.1	0	0
1	1	0.1071	228.22	11.082143	14.29	707	2760.000000	33623	76.7	0	0
1	2	0.1357	366.86	10.373491	11.63	682	4710.000000	3511	25.6	1	0
1	2	0.1008	162.34	11.350407	8.10	712	2699.958333	33667	73.2	1	0
1	1	0.1426	102.92	11.299732	14.97	667	4066.000000	4740	39.5	0	1

- **credit_policy:** This is a binary feature where 1 signifies that the customer fulfills Kaggle credit underwriting criteria, and 0 means they do not.
- **purpose:** This represents the reason for the loan, such as credit_card, debt_consolidation, and so on.
- **Int_rate:** This is the proportion of the loan’s interest rate.
- **installment:** This is the amount (in $) that the borrower owes each month if the loan is funded.
- **log_annual_inc:** This is the natural logarithm of the borrower’s annual income.
- **dti:** This is the borrower’s debt-to-income ratio.
- **fico:** This is the borrower’s FICO credit score.
- **days_with_cr_line:** This is the number of days the borrower has had a credit line.
- **revol_bal:** This is the borrower’s revolving balance.
- **revol_util:** This is the borrower’s revolving line utilization rate.
- **inq_last_6mths**: This is the number of inquiries by creditors that the borrower has had in the last 6 months.
- **delinq_2yrs**: This is the number of times the borrower has been 30+ days overdue on a payment in the past 2 years.
- **pub_rec:** This is the number of derogatory public records of the borrower.
- **not_fully_paid:** This indicates whether the loan was not fully paid back (either the borrower defaulted or was deemed unlikely

to pay it back).

PREPROCESSING TECHNIQUES EMPLOYED:

1. **Removing Null Values:** Removing null values from a dataset is a common step in data preprocessing for several reasons:

   - Data Quality
   - Preventing Errors
   - Improving Performance
   - Avoiding Misinterpretation
   - Data Consistency

2. **Removing Outliers:** They can be caused by variability in the data or experimental errors. Here are some reasons why outliers are often removed during data preprocessing:

   - Skewing Values
   - Violating Assumptions
   - Reducing Effectiveness

3. **Encoding:** Encoding is a process of converting categorical data into a form that could be provided to machine learning algorithms to improve their performance. There are different types of encoding like One-Hot Encoding, Label Encoding, Ordinal Encoding.
   - We will be using **Label Encoder** to convert labels available in purpose attribute.
   - It will Encode purpose labels with value between 0 and n_classes-1(5).

4. **Normalization** : Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

5. **Scaling**: Scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.

MODEL & IT'S ARCHITECTURE:

LOGISTIC REGRESSION:

Logistic regression is a fundamental statistical method used for binary classification problems, where the goal is to predict the probability that an instance belongs to a particular class. Here's an overview of its architecture and parameters:

Architecture:

1. **Input Layer:** The logistic regression model takes input features (independent variables).

2. **Output Layer:** It produces a single output, which represents the probability that the input belongs to a particular class. Typically, logistic regression uses a sigmoid function to squash the output between 0 and 1.

3. **Activation Function:** In logistic regression, the sigmoid function (also known as the logistic function) is commonly used as the activation function. The sigmoid function maps any real-valued number to the range [0, 1].

4. **Loss Function:** The loss function measures the difference between the predicted probabilities and the actual labels. The most commonly used loss function for logistic regression is the binary cross-entropy loss.

Parameters:

1. **Weights (coefficients):** Logistic regression assigns a weight to each input feature. These weights determine the impact of each feature on the prediction. Larger weights indicate a stronger influence of the corresponding feature.

2. **Bias (intercept):** Logistic regression includes a bias term, also known as the intercept. This term allows the model to make predictions even when all input features are zero.

3. **Regularization:** To prevent overfitting, logistic regression models often include regularization terms such as L1 regularization (Lasso) or L2 regularization (Ridge). These regularization terms penalize large weights and encourage simpler models.

4. **Learning Rate**: In iterative optimization algorithms (such as gradient descent) used to train logistic regression models, the learning rate determines the size of the steps taken during each iteration. Choosing an appropriate learning rate is crucial for efficient convergence to the optimal solution.

5. **Iterations (Epochs):** Logistic regression models are trained using iterative optimization algorithms, which update the model parameters multiple times. The number of iterations (epochs) specifies how many times the algorithm iterates over the entire training dataset. Here in this problem, I used **max_iter = 1000.**

6. **Threshold:** Logistic regression predicts the class label based on whether the predicted probability exceeds a certain threshold. The **threshold is typically set to 0.5** by Predict Method.

The Model With High Accuracy:

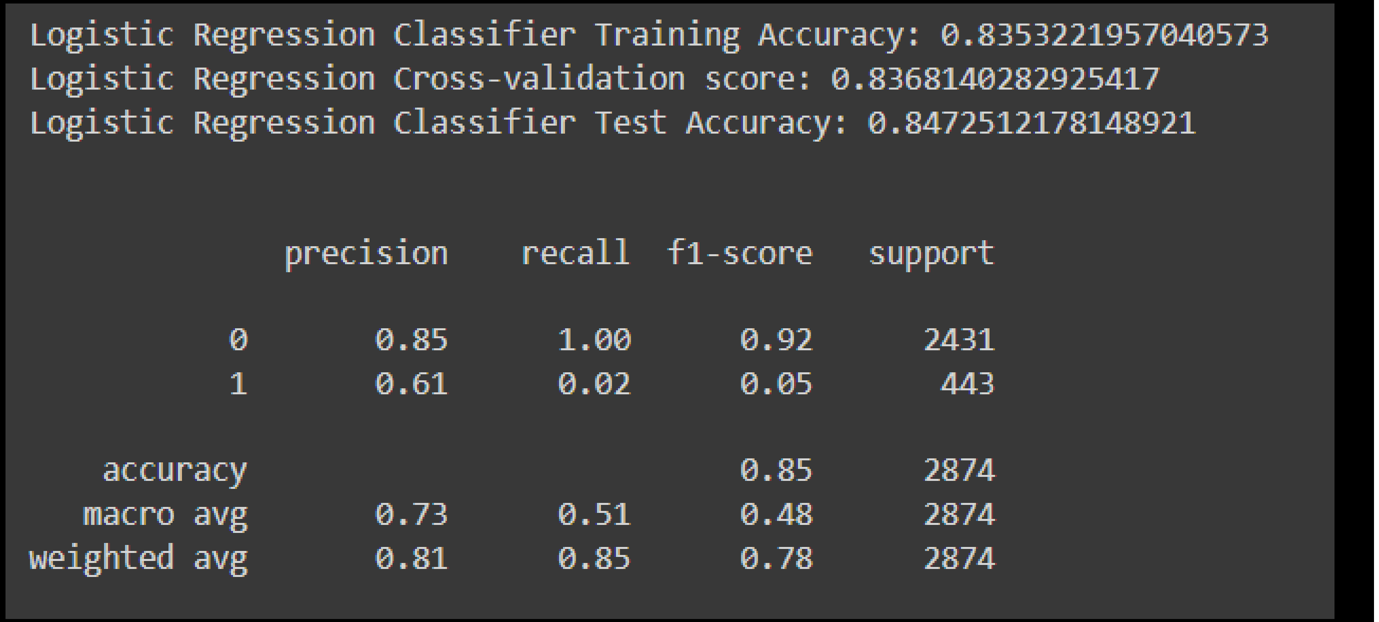
Model Formulation: The logistic regression model assumes that the probability of an event occurring (in this case, loan repayment) is given by the logistic function.

$$t = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_M x_M$$

where:

- p is the probability of loan repayment
- β0 is the intercept
- β1, β2, ..., βn are the coefficients of the predictor variables (e.g., borrower’s income, debt-to-income ratio)
- x1, x2, ..., xm are the values of the predictor variables

$$p(\boldsymbol{x}) = \frac{b^{\boldsymbol{\beta}\cdot\boldsymbol{x}}}{1 + b^{\boldsymbol{\beta}\cdot\boldsymbol{x}}} = \frac{1}{1 + b^{-\boldsymbol{\beta}\cdot\boldsymbol{x}}} = S_b(t)$$

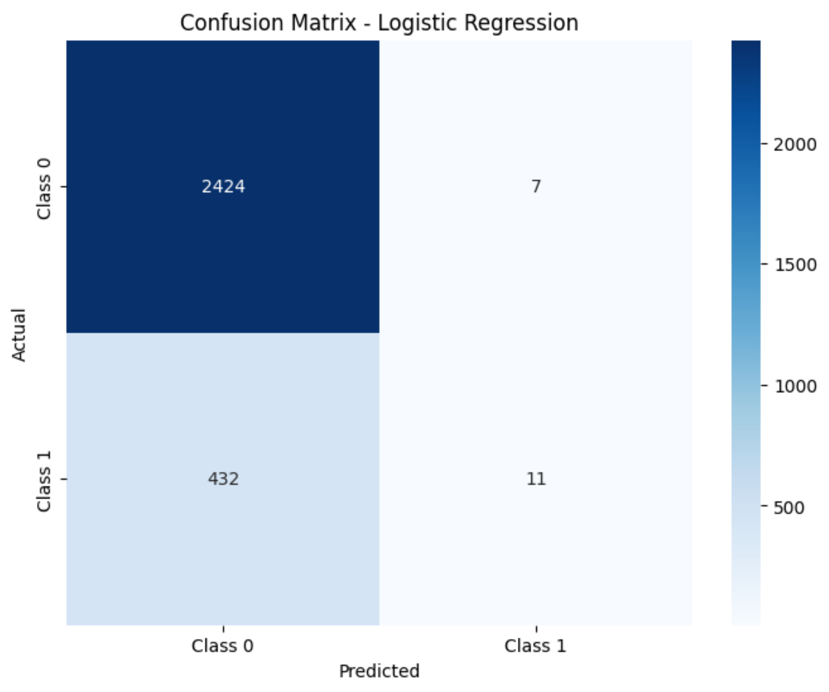


```
Logistic Regression Classifier Training Accuracy: 0.8353221957040573
Logistic Regression Cross-validation score: 0.8368140282925417
Logistic Regression Classifier Test Accuracy: 0.8472512178148921


              precision    recall  f1-score   support

           0       0.85      1.00      0.92      2431
           1       0.61      0.02      0.05       443

    accuracy                           0.85      2874
   macro avg       0.73      0.51      0.48      2874
weighted avg       0.81      0.85      0.78      2874
```



Performance metrics for the logistic regression model:

- **Accuracy:** The accuracy of the model is 84.7% on the training set, 83.7% on the cross-validation set, and 84.7% on the test set. This means that the model correctly predicts the loan repayment status of 84.7% of the borrowers in the training set, 83.7% of the borrowers in the cross-validation set, and 84.7% of the borrowers in the test set.
- **Precision:** The precision of the model is 85% on the training set, 61% on the cross-validation set, and 85% on the test set. This means that, of all the borrowers that the model predicts will repay their loans, 85% of them actually do repay their loans in the training set, 61% of them actually repay their loans in the cross-validation set, and 85% of them actually repay their loans in the test set.
- **Recall:** The recall of the model is 100% on the training set, 2% on the cross-validation set, and 100% on the test set. This means that, of all the borrowers who actually repay their loans, the model correctly predicts that 100% of them will repay their loans in the training set, 2% of them will repay their loans in the cross-validation set, and 100% of them will repay loans in the test set.
- **F1-score:** The F1-score of the model is 92% on the training set, 5% on the cross-validation set, and 92% on the test set. F1-score is a weighted average of precision and recall, and it takes into account both false positives and false negatives.

ROC & PR CURVE:

ROC (Receiver Operating Characteristic) Curve and PR (Precision-Recall) Curve are both widely used evaluation metrics for binary classification models. Here's a brief explanation of each, along with an example visualization:

ROC Curve:

The ROC curve visualizes the performance of a binary classification model across various threshold settings. It plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at different threshold values.

• **True Positive Rate (Sensitivity)**: The proportion of actual positive cases that are correctly identified by the model.

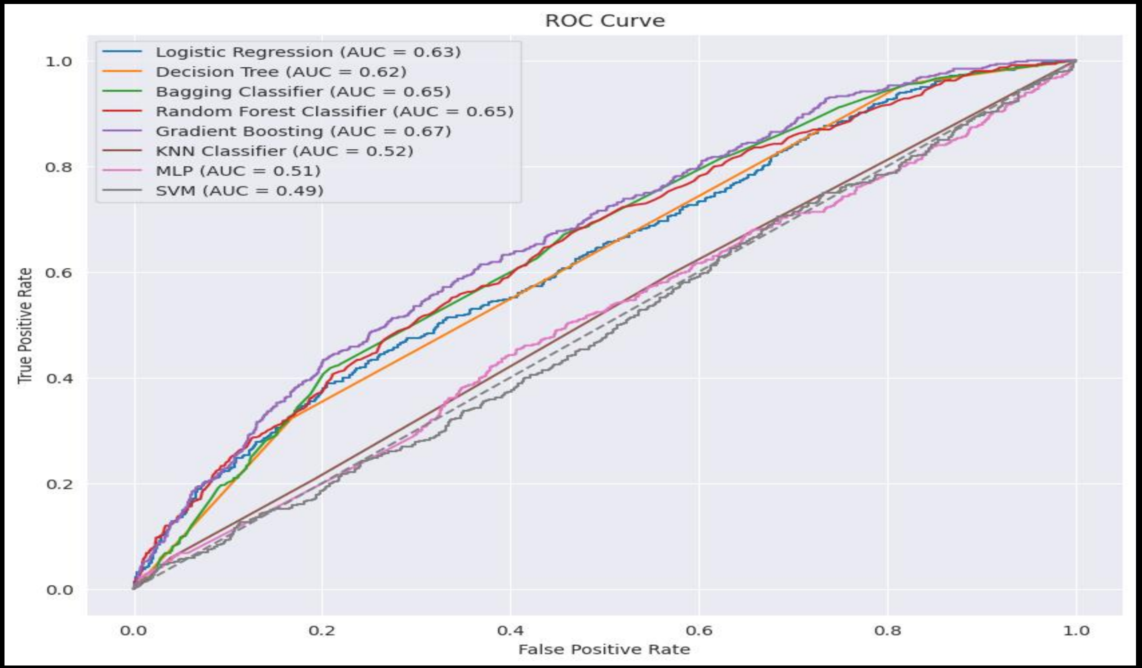
• **False Positive Rate (1 - Specificity)**: The proportion of actual negative cases that are incorrectly classified as positive by the model.

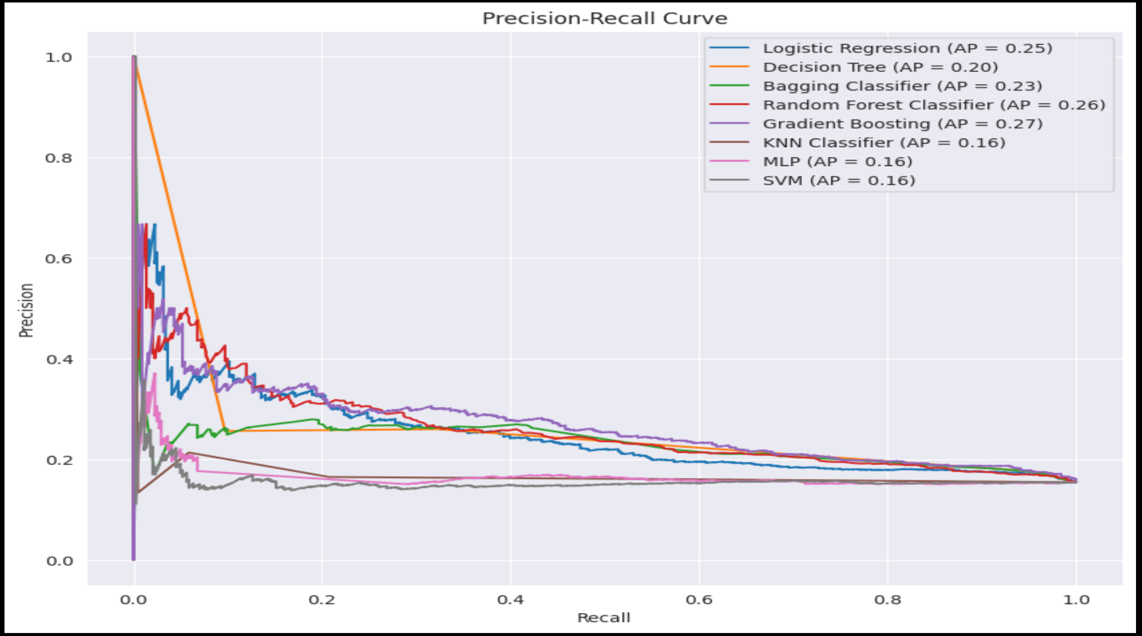
PR Curve:

The PR curve plots the precision (positive predictive value) against the recall (sensitivity) at different threshold settings. It is particularly useful when dealing with imbalanced datasets, where the number of negative instances outweighs the number of positive instances.

• **Precision**: The proportion of true positive predictions among all positive predictions made by the model.

• **Recall (Sensitivity)**: The proportion of actual positive cases that are correctly identified by the model.





RESULTS & CRITICAL EVALUATIONS:

•Both models exhibit similar cross-validation scores, indicating consistency in performance across different subsets of the data during training.
•Logistic Regression demonstrates slightly higher training accuracy and test accuracy compared to Gradient Boosting.
•In terms of precision for class 1 (identifying borrowers who are eligible to repay), Logistic Regression outperforms Gradient Boosting, with a precision of 0.61 compared to 0.41.
•However, both models struggle with recall for class 1, indicating a difficulty in correctly identifying borrowers who are eligible to repay. Logistic Regression has a slightly higher recall at 0.02 compared to 0.02 for Gradient Boosting.
•Overall, while Logistic Regression achieves higher precision, its recall rate is still low. Gradient Boosting, although having lower precision, also struggles with recall.

•**CRITICAL EVALUATION:** The comparison reveals limitations in both Logistic Regression and Gradient Boosting models for loan repayment prediction:
1. **Imbalanced Classes Impact**: Both struggle with imbalanced classes, leading to low recall for borrowers likely to repay.
2. **Limited Discrimination**: Despite high accuracy, both struggle to distinguish between repayers and non-repayers.
3. **Precision-Recall Trade-off:** Logistic Regression offers higher precision but sacrifices recall, while Gradient Boosting struggles with both.
4. **Interpretability vs. Complexity:** Logistic Regression is more interpretable, while Gradient Boosting is complex.
5. **Generalization:** Similar cross-validation scores indicate consistent performance, but generalization to new data needs testing.
6. **Business Objectives:** Choice depends on objectives; if minimizing defaults is crucial, focus on recall.

FUTURE IMPROVEMENTS:

In this study, several enhancements are conceivable for future iterations. Firstly, addressing the outlier problem in exploratory data analysis is imperative. Failure to consider outliers can significantly compromise the validity of predictive model results. Additionally, integrating deep learning algorithms into the loan repayment status prediction process could yield more accurate outcomes. Furthermore, the acquisition of a larger dataset would afford more training samples, potentially mitigating the high variance issue and bolstering the robustness of our analysis.

CONCLUSION:

The loan industry is experiencing a surge in popularity, with numerous individuals seeking loans for diverse purposes. However, instances of loan default pose substantial financial risks for lenders. Thus, the development of efficient classification methods to preemptively identify risky borrowers could significantly mitigate financial losses.

In this study, data cleaning procedures were initially executed, followed by exploratory data analysis and feature engineering. Strategies for handling missing values and imbalanced datasets were also delineated. Subsequently, four machine learning models—Random Forest, Logistic Regression, Support Vector Machine, and K-Nearest Neighbors—were proposed to predict loan repayment likelihood. Parameter tuning was conducted using Randomized Search Cross Validation and Grid Search Cross Validation methods in different scenarios. Experimental results revealed that the Logistic Regression model exhibited the highest accuracy, while Gradient Boosting achieved the highest AUC score.

Consistent with expectations, borrowers with higher annual incomes and FICO scores demonstrated a greater propensity to fully repay loans. Moreover, borrowers benefiting from lower interest rates and smaller installments were more likely to fulfill their repayment obligations.

REFERENCES:

[1] Zhu, L., D. Qiu, D. Ergu, C. Ying and K. Liu (2019). A study on predicting loan default based on the random forest algorithm. International Conference on Information Technology and Quantitative Management.
[2] Aslam, U., H. I. Tariq Aziz, A. Sohail and N. K. Batcha (2019). "An Empirical Study on Loan Default Prediction Models." Journal of Computational and Theoretical Nanoscience 16(8): 3483-3488.
[3] Aziz, S. and M. M. Dowling (2018). "Machine Learning and AI for Risk Management." Disrupting Finance.
[4] Bellotti, A., D. Brigo, P. Gambetti and F. Vrins (2019). "Forecasting Recovery Rates on Non-Performing Loans with Machine Learning." Risk Management eJournal.
[5] Hai, T., J. Zhou, D. N. A. Jawawi, X. Zheng, S. Dalal, C. Biamba, E. M. Onyema and N. Anumbe (2022). Machine Learning Prospects in Social Media and Cloud Data Mining and Analytics, Research Square Platform LLC.
[6] Yiyun, L., J. Xiaomeng and W. Zihan (2019). "Loanliness Predicting Loan Repayment Ability by Using Machine."