Capstone Project on Housing Credit Risk Prediction

Final Report

Group Number: Group-2

Project Mentor: Mr. Ankush Bansal

Team Members: Robinson Vinu Thomas, Kishore Chandrasekaran, Padmanabhan A, Sandhiya A, Sanjay Babu

Contents:

1. Problem Statement and Solution Strategy:

Problem Statement:

The problem, provided by Home Call Credit Group, involves a binary classification task. The objective is to predict whether a loan applicant will repay or default based on various features describing their financial and behavioral history. This is a critical challenge in the domain of credit risk assessment.

Dataset:

The "Housing Credit Risk Prediction" dataset contains 307,511 rows and 122 columns. It encompasses details about client information and other attributes relevant to credit risk assessment.

Findings:

Numeric Variables: 106

Categorical Variables: 16

Missing/Null Values: 9,152,465

No redundant columns present

The project aims to create a robust predictive model using historical client data.

2. Overview of the Final Process:

Data Preparation:

- Imported datasets.
- Resolved missing values.
- Outlier Treatment
- Encoded categorical features.

Exploratory Data Analysis (EDA):

- Analyze data patterns and relationships.
- Gain insights into variable significance.

Feature Engineering:

- Created new features for enhanced predictive power.
- Optimized the dataset for model training.

Model Building:

- Train Test Split
- Implemented classification models
- Train models using various libraries.
- Evaluated and compare performances.

Hyperparameter Tuning:

- Refine models using GridSearchCV
- Fine-tune hyperparameters.

3. Step-by-Step Walkthrough of the Solution:

Data Preparation:

- Import and Clean Datasets: Imported necessary libraries and datasets, totaling 307,511 samples and 122 features.

Handle Missing Values:

- Removed columns with more than 40% missing values, resulting in 73 columns.

- The new dataset has 307,511 entries and 73 columns.
- Imputed the missing values with median and mode.

Outlier Treatment:

- Utilized the IQR (Interquartile Range) method to identify outliers for numerical variables in the dataset.
- Detected outliers for almost all the numerical column.
- Handled outliers by removing outliers, transforming variables by using the Power Transformer to reduce skewness in the numerical features.

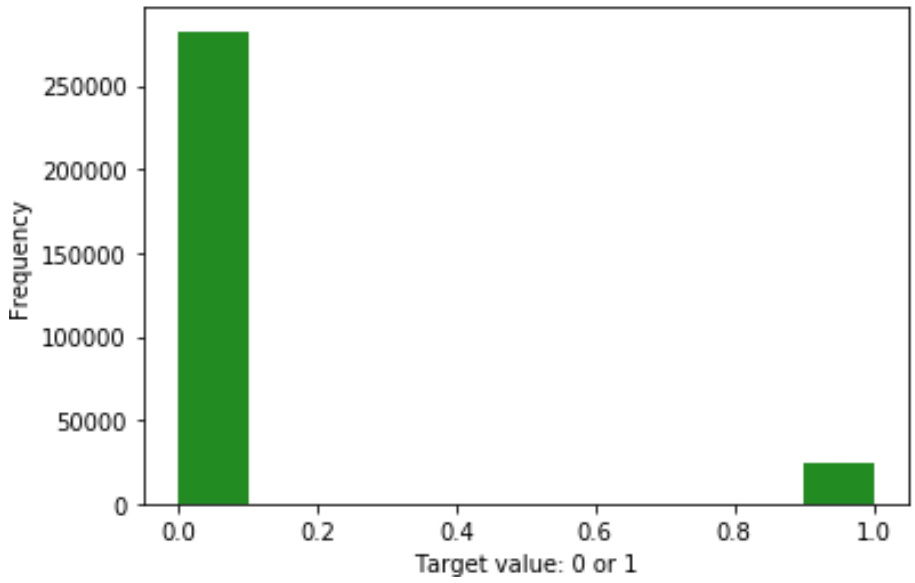
Encoding:

- Used one-hot encoding to convert categorical variables into numerical format to prepare for Data Analysis.

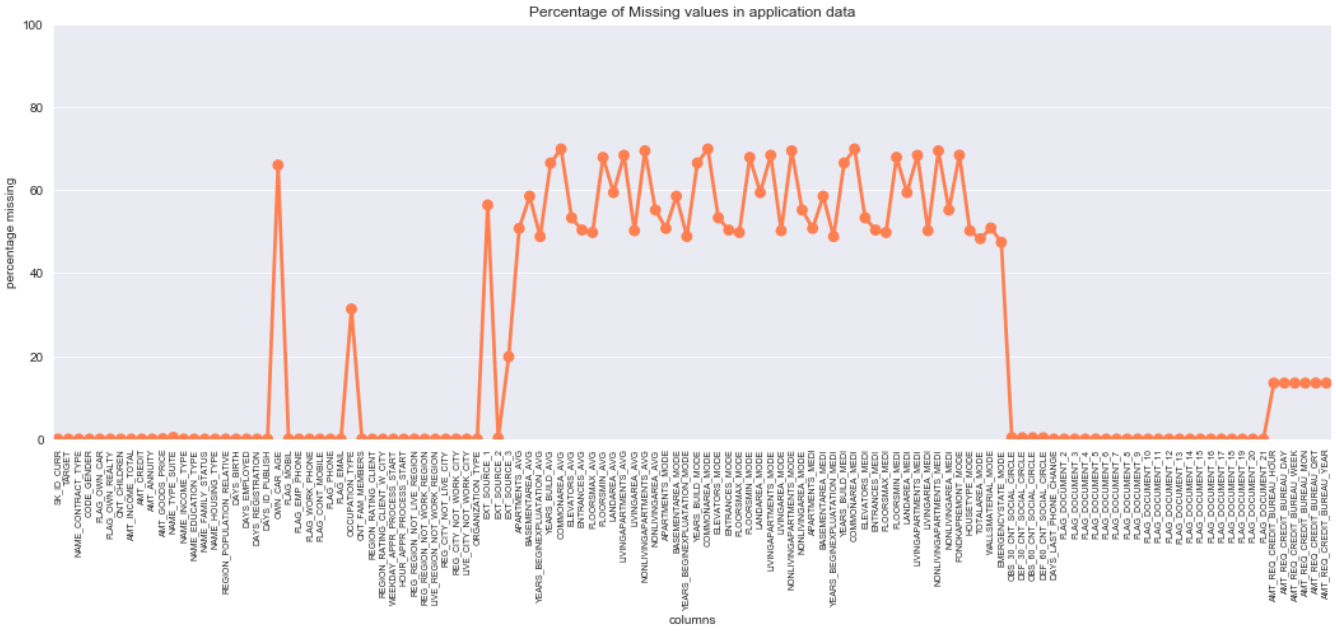
Exploratory Data Analysis:

Visualize Data Distribution and Patterns:

- Explored data distribution and patterns, identifying class imbalance with 282,686 repaid loans and 24,825 defaulted loans.

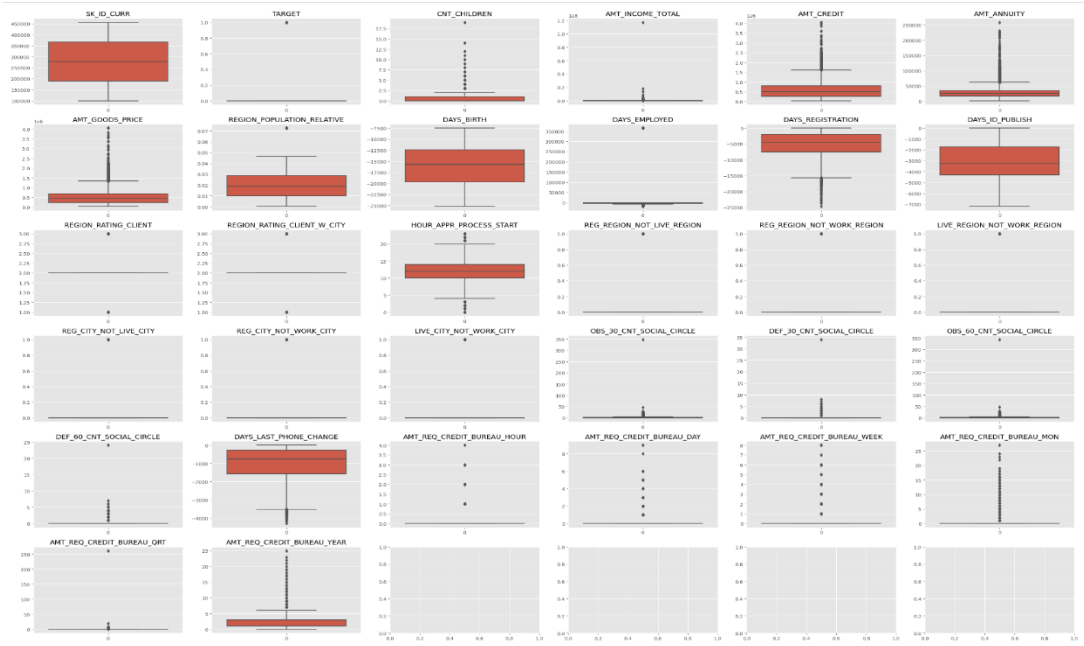


- Identify Key Features and Potential Correlations: Analyzed features and correlations, observing missing data in the main dataset and identifying features with high loan default rates.
- On checking for missing data, it is seen in Figure 2 that a number of features in the main dataset (application) have missing values almost 50%. The features with high fractions of missing data need to be discarded, and those with some missing values need to be imputed before training any model. Similar checks are performed on the other datasets



Univariate Analysis:

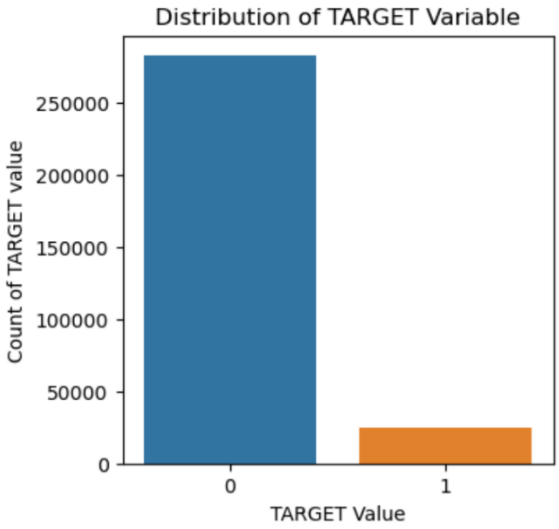
- Boxplots were used to visualize the distribution of numeric variables.
- Almost all variables showed outliers, indicating the need for outlier treatment



Bivariate Analysis:

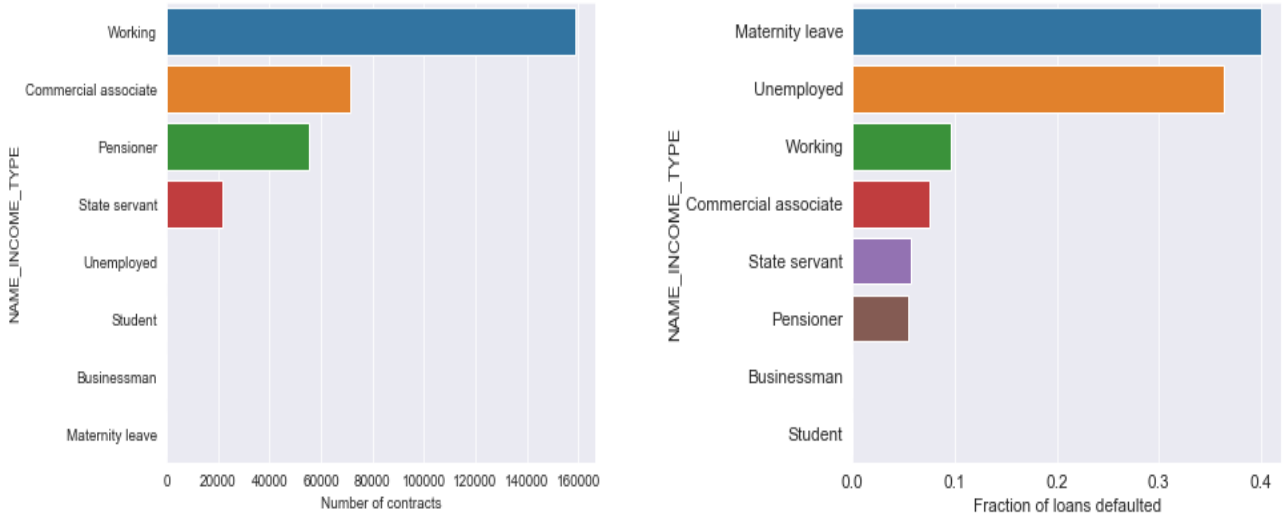
Distribution of TARGET Variable:

- A count plot was created to show the distribution of the TARGET variable.
- The dataset is highly imbalanced, with a ratio of approximately 11.4 non-defaulters (TARGET=0) for every defaulter (TARGET=1).



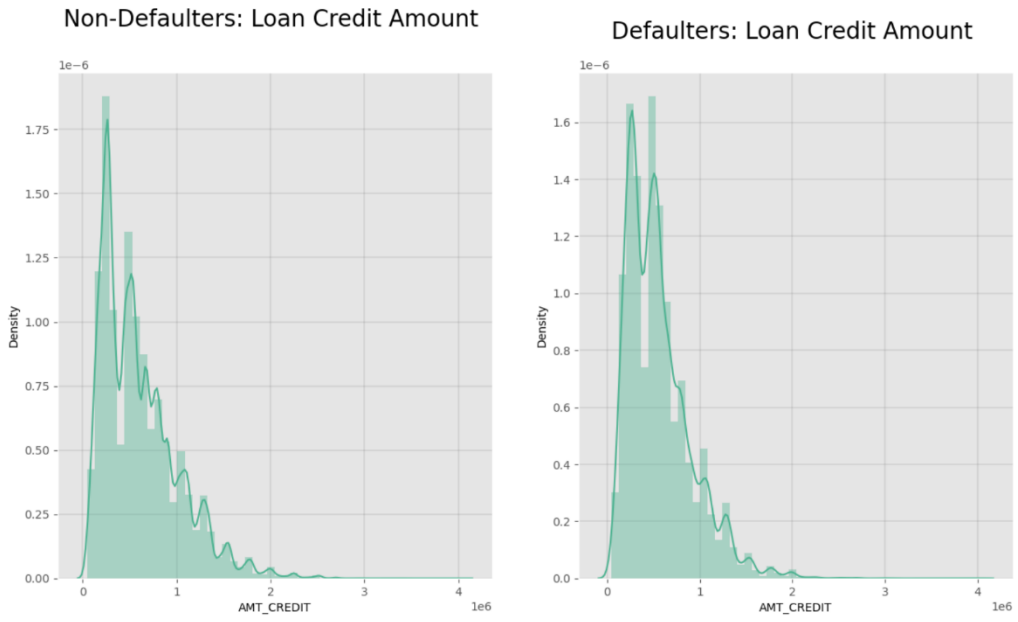
The next step entailed continuing the exploratory data analysis to see if any response features have significant differences for cases when loans are repaid

as opposed to when loans are defaulted. A number of figures were created for categorical features with bar plots for each type, with each figure containing, a) the total number of categories for each response feature, b) the fraction of each category with loan defaulted. For example, for the feature income type (NAME_INCOME_TYPE) in Figure below, we see that there are more loans taken by those who are working and they are more defaulted by those who are on maternity leave or unemployed. A comprehensive list of bar plots is available in the Jupyter notebook.



Distribution of loan credit amount:

- This visualization helps compare the distribution of loan credit amounts between non-defaulters and defaulters.



For the continuous features, the distribution of a feature for the cases when loans are repaid and defaulted, were plotted to examine differences and check data quality. For example, for the feature 'DAYS_EMPLOYED' in Figure 5A, we can see that the distribution is hard to perceive and has some anomalous values of days employed more than 350000 days (958 years), which is impossible and needs to be corrected. On removing the outliers, and converting the days to years, we can see a more interpretable distribution in figure below, where there are a greater number of loans defaulted by people who are employed for fewer years.

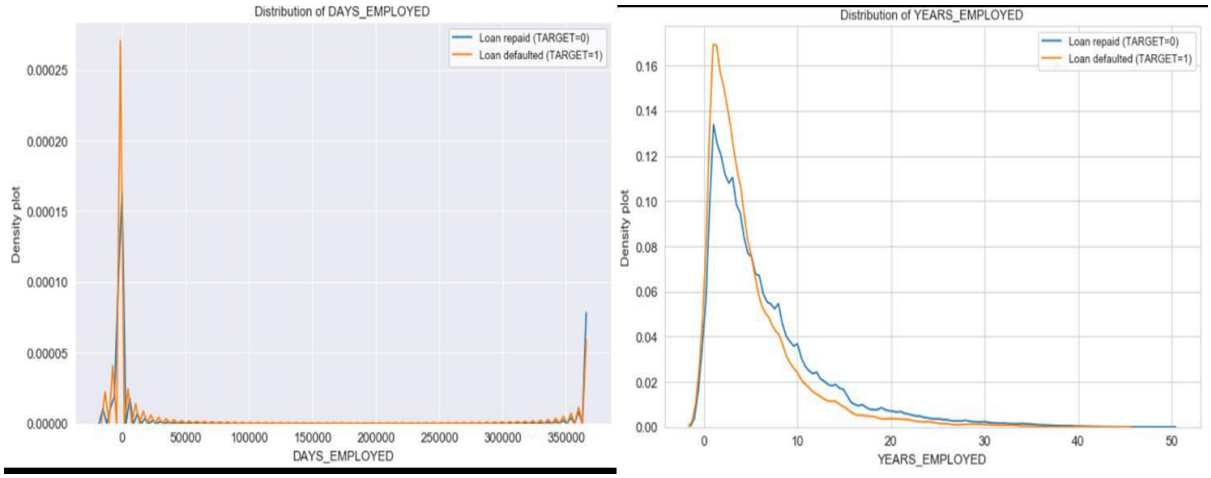


Fig- A) Distribution of days employed, B) Distribution of years employed (after correction)

A comprehensive list of such distribution plots is available in the Jupyter notebook. Such an analysis helps us gain an insight into the features which need to be corrected, and those which may help identify the likelihood of loan default and can turn out to be important features for model training.

Hypothesis Test for Categorical Variables:

A hypothesis test for categorical variables involves assessing whether there is a significant association or difference between categorical variables. It is typically performed using the chi-square test for independence. Here's a brief overview:

Null Hypothesis (H0):

Statement: There is no association or difference between the categorical variables.

Symbolically: H0: Variable A and Variable B are independent.

Alternative Hypothesis (H1):

Statement: There is a significant association or difference between the categorical variables.

Symbolically: H1: Variable A and Variable B are dependent.

Chi-Square Test: The chi-square test calculates a test statistic (chi-square statistic) based on the observed and expected frequencies in a contingency table.

Hypotheses for Loan Default Predictors:

1. **Car Ownership (FLAG_OWN_CAR):**
    - *Hypothesis:* Borrowers owning a car are more likely to default on loans.
    - *Rationale:* Car ownership implies additional financial commitments, potentially impacting loan repayment.
2. **Real Estate Ownership (FLAG_OWN_REALTY):**
    - *Hypothesis:* Borrowers owning real estate are less likely to default on loans.

- *Rationale:* Real estate ownership suggests a more stable financial situation, reducing default risk.

3. **Employer-Provided Phone (FLAG_EMP_PHONE):**
    - *Hypothesis:* Borrowers with employer-provided phones are less likely to default on loans.
    - *Rationale:* Employer-provided phones may indicate stable employment and financial stability.
4. **Work Phone (FLAG_WORK_PHONE):**
    - *Hypothesis:* Borrowers with work phones are less likely to default on loans.
    - *Rationale:* Work phones may correlate with stable employment, influencing lower default likelihood.
5. **Phone Ownership (FLAG_PHONE):**
    - *Hypothesis:* Borrowers with personal phones are less likely to default on loans.
    - *Rationale:* Phone ownership may enhance accessibility, potentially reducing the likelihood of default.

Key Takeaway: Understanding these hypotheses helps refine risk assessment strategies, allowing for targeted measures to mitigate the risk of loan default based on borrower characteristics.

Feature Engineering:

- After exploring the data distributions, we can conduct feature engineering to prepare the data for model training.

- This includes operations like replacing outliers, imputing missing values, one-hot encoding categorical variables, and rescaling the data.
- The process of replacing outliers involves removing values from data which are greater than three standard deviations from the mean Since categorical variables cannot be directly interpreted by most classifiers, they need to be encoded as numbers.
- This can be done by label encoding or one-hot encoding. Label encoding assigns each category in a feature with an integer without creating new columns.
- One-hot encoding creates a new column for each category where each observation is assigned as 1, while the other category columns are assigned value of 0.
- It is preferred in this project since it is possible that the label numbers in the label encoding are wrongly interpreted by the model as holding some significance, while one-hot encoding does not have that issue.
- For tackling missing values, a two-step approach is followed. The features which have more than 60% data missing are removed, while the remaining features have data imputed, i.e., categorical features are filled with the most frequent column, and other features are filled with the median value.
- Rescaling of the features involves transforming each feature to a range of 0-1.
- transformation can be applied to one or more columns in a single table such as the difference between two columns or the absolute value of one column. Some common primitives in Featuretools are count, mean, median, trend, maximum, minimum, number of words, cumulative sum, difference etc.

Train-Test-Split:

- The train_test_split function is a utility function provided by the scikit-learn library in Python.
- The dataset is splitted into two subsets of 70:30 ratio : one for training a machine learning model and the other for testing the model's performance.
- Since the data was imbalanced SMOTE was done with a 60:40 ratio.

Model Building:

Base Model: Decision Tree Classifier

A simple decision tree model serving as the baseline.

Performance Metrics:

1. **Training Accuracy: 1.00**
    - The model achieves perfect accuracy (100%) on the training data. This suggests that the decision tree has learned the training data thoroughly, but it raises concerns about potential overfitting.
2. **Test Accuracy: 0.83**
    - The accuracy on the test data is 83%, indicating that the model performs well on unseen data. However, the significant difference between training and test accuracy suggests overfitting.
3. **Precision: 0.69**
    - Precision measures the accuracy of positive predictions. A precision of 0.69 means that out of the instances predicted as positive, 69% were correct.
4. **Recall: 0.73**

- Recall (Sensitivity or True Positive Rate) measures the ability of the model to capture all positive instances. A recall of 0.73 indicates that the model correctly identifies 73% of the actual positive instances.

5. **F1 Score: 0.71**
    - The F1 score is the harmonic mean of precision and recall. It provides a balanced measure of a model's overall performance. An F1 score of 0.71 is a reasonable balance between precision and recall.

Summary:

- The model has likely overfit the training data due to the perfect training accuracy.
- The test accuracy, while reasonably high, indicates a potential challenge in generalization.
- Precision, recall, and F1 score provide insights into the model's ability to make accurate positive predictions.

2.Logistic Regression:

- **Training Accuracy (91%):** The model performs well on the training data, indicating a good fit.
- **Test Accuracy (90%):** The model generalizes well to new, unseen data with a high test accuracy.
- **Precision (0.95):** The model has a high precision for class 1, meaning that when it predicts a positive instance, it is correct 95% of the time.
- **Recall (0.71):** The recall for class 1 is relatively lower, indicating that the model may miss some of the actual positive instances.

- **F1-Score (0.81):** The F1-score provides a balance between precision and recall, and a score of 0.81 is reasonable.
- Summary:
- Logistic Regression shows promising results with high accuracy and precision. However, the recall for class 1 could be improved for better identification of positive instances. Further optimization or exploration of model parameters may be beneficial.

- Logistic regression, a linear model, with high accuracy and precision.

3.Bagging Classifier:

- **Training Accuracy (96%):** The model performs exceptionally well on the training data, indicating a strong fit.
- **Test Accuracy (90%):** The model generalizes well to new, unseen data with a high test accuracy.
- **Precision (0.95):** The model has a high precision for class 1, meaning that when it predicts a positive instance, it is correct 95% of the time.
- **Recall (0.68):** The recall for class 1 is relatively lower, indicating that the model may miss some of the actual positive instances.
- **F1-Score (0.79):** The F1-score provides a balance between precision and recall, and a score of 0.79 is reasonable.

Summary:

- The Bagging Classifier shows strong performance, especially in terms of training accuracy. However, similar to the Logistic Regression model, the recall for class 1 could be improved for better identification of positive instances. Further optimization or exploration of model parameters may be beneficial.

4.Decision Tree (Best Parameters):

- **Parameters Used:**
    - **Criterion:** Entropy, **Max Depth:** 15, **Min Samples Leaf:** 15, **Min Samples Split:** 10

Model Performance:

- **Training Accuracy (84%):** The model achieves an 84% accuracy on the training dataset.
- **Test Accuracy (84%):** The model generalizes well to new, unseen data, maintaining an 84% accuracy on the test dataset.
- **Precision (Class 1**): 0.78 indicates that when the model predicts a positive instance, it is correct 78% of the time.
- **Recall (Class 1):** 0.60 suggests that the model may miss some of the actual positive instances.
- **F1-Score (Class 1):** 0.67 provides a balance between precision and recall for class 1.

Summary:

- The optimized Decision Tree model with specified parameters shows improved performance compared to the initial Decision Tree.

5.Gradient Boosting Classifier:

Model Performance:

- **Training Accuracy (88%):** The model achieves an 88% accuracy on the training dataset.
- **Test Accuracy (88%):** The model generalizes well to new, unseen data, maintaining an 88% accuracy on the test dataset.

- **Precision (Class 1):** 0.96 indicates that when the model predicts a positive instance, it is correct 96% of the time.
- **Recall (Class 1):** 0.60 suggests that the model may miss some of the actual positive instances.
- **F1-Score (Class 1):** 0.74 provides a balance between precision and recall for class 1.

Summary:

- The Gradient Boosting Classifier exhibits strong predictive performance with an 88% accuracy.

6.Random Forest Classifier:

Model Performance:

- **Training Accuracy (100%):** The model achieves a perfect accuracy of 100% on the training dataset.
- **Test Accuracy (92%):** The model generalizes well to new, unseen data, maintaining a strong accuracy of 92% on the test dataset.
- **Precision (Class 1):** 0.96 indicates that when the model predicts a positive instance, it is correct 96% of the time.
- **Recall (Class 1):** 0.60 suggests that the model may miss some of the actual positive instances.
- **F1-Score (Class 1):** 0.74 provides a balance between precision and recall for class 1.

Summary:

- The Random Forest Classifier exhibits remarkable predictive performance with a perfect training accuracy of 100% and a strong test accuracy of 92%. However, such high training accuracy may raise concerns about overfitting. Further exploration, tuning, or consideration of other models is

recommended to optimize performance based on project requirements

7. KNeighbors Classifier:

Model Performance:

- **Training Accuracy (49%):** The model exhibits a training accuracy of 49%, suggesting that it struggles to fit the training data well.
- **Test Accuracy (46%):** The low test accuracy of 46% indicates that the model's performance on new, unseen data is not satisfactory.
- Model Evaluation:
- **Recall (Class 1):** 0.81 reflects the ability of the model to identify actual positive instances.
- **Precision (Class 1):** 0.81 indicates that when the model predicts a positive instance, it is correct 81% of the time.
- **F1-Score (Class 1):** 0.81 provides a balance between precision and recall for class 1.

Summary:

The K-Nearest Neighbors (KNN) model, with a training accuracy of 49% and a test accuracy of 46%, seems to struggle with the dataset. In comparison to other models, such as Random Forest and Logistic Regression, the KNN model performs relatively poorly on both training and test datasets. Consideration of alternative models or further tuning is recommended for improved performance.

8. KNeighbors Classifier (Tuned):

- **Accuracy (89.73%):**

    - The model achieved an accuracy of 89.73%, indicating the overall correctness of predictions on both positive and negative instances.
- **Recall (87.78%):**
    - The recall score of 87.78% suggests that the model effectively identifies a high percentage of actual positive instances, minimizing false negatives.
- **Precision (78.68%):**
    - With a precision of 78.68%, the model shows the ability to make positive predictions accurately. It measures the percentage of true positive predictions among all positive predictions.
- **F1 Score (82.98%):**
    - The F1 score, combining precision and recall, is 82.98%. This score provides a balanced assessment of the model's performance, especially when there is an uneven class distribution.

9.Bagging Classifier (Decision Tree):

1. **Accuracy: 89.68%**
    - The proportion of correctly classified instances out of the total instances. The model is accurate in predicting both positive and negative cases.
2. **Recall: 67.88%**
    - The ability of the model to correctly identify positive instances among all actual positives. A moderate recall indicates that the model captures a substantial portion of actual positive cases.
3. **Precision: 94.34%**
    - The precision represents the accuracy of positive predictions made by the model. A high precision indicates

that when the model predicts a positive case, it is likely to be correct.

4. **F1 Score: 78.95%**
    - The F1 Score is the harmonic mean of precision and recall. It provides a balance between precision and recall, considering both false positives and false negatives. In this case, the F1 Score indicates good overall model performance.

Summary:

The Bagging Classifier, utilizing decision trees as the base model, demonstrates a balance between precision and recall, making it suitable for applications where both false positives and false negatives need to be minimized.

10.AdaBoost Classifier:

- Accuracy: 86.50%
- Recall: 63.97%
- Precision: 84.99%
- F1 Score: 72.99%
- AdaBoost algorithm combining weak learners to enhance accuracy.

12.AdaBoost (Logistic Regression):

1. **Accuracy: 86.50%**
    - Accuracy represents the proportion of correctly classified instances out of the total instances. In this case, the model accurately predicts both positive and negative cases around 86.50% of the time.
2. **Recall: 63.97%**

- Recall measures the ability of the model to correctly identify positive instances among all actual positives. A recall of 63.97% indicates that the model captures a moderate portion of the actual positive cases.

3. **Precision: 84.99%**
    - Precision represents the accuracy of positive predictions made by the model. With a precision of 84.99%, when the model predicts a positive case, it is likely to be correct most of the time.
4. **F1 Score: 72.99%**
    - The F1 Score is the harmonic mean of precision and recall. It provides a balanced measure considering both false positives and false negatives. In this case, the F1 Score indicates good overall model performance.

AdaBoost is an ensemble learning algorithm that combines multiple weak learners to create a strong learner. It focuses on improving accuracy by giving more weight to misclassified instances, allowing subsequent weak learners to focus on those areas. The achieved metrics suggest that AdaBoost is effective in enhancing accuracy while maintaining a reasonable balance between precision and recall.

13.XGB Classifier(Extreme Gradient Boosting):

1. **Accuracy: 93.13%**
    - Accuracy represents the proportion of correctly classified instances out of the total instances. In this case, the XGB Classifier accurately predicts both positive and negative cases around 93.13% of the time.
2. **Recall: 76.79%:**Recall measures the ability of the model to correctly identify positive instances among all actual positives.

A recall of 76.79% indicates that the model captures a substantial portion of the actual positive cases.

3. **Precision: 98.88%**
    - Precision represents the accuracy of positive predictions made by the model. With a precision of 98.88%, when the model predicts a positive case, it is highly likely to be correct.
4. **F1 Score: 86.45%**
    - The F1 Score is the harmonic mean of precision and recall. It provides a balanced measure considering both false positives and false negatives. In this case, the F1 Score indicates excellent overall model performance.

Summary:

- XGBoost is an advanced and powerful gradient boosting algorithm known for its high performance and efficiency. It sequentially builds a series of weak learners (usually decision trees) and combines them to create a strong predictive model. Key features of XGBoost include regularization, parallel computing, and handling missing values.

14.Stacking Classifier:

1. **Accuracy: 90.59%**
    - Accuracy measures the proportion of correctly classified instances out of the total instances. In this case, the meta-model achieves an accuracy of 90.59%, indicating a high level of overall correctness in predictions.
2. **Recall: 81.80%:** Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify

positive instances among all actual positives. A recall of 81.80% suggests that the meta-model captures a substantial portion of the actual positive cases.

3. **Precision: 84.68%**
    - Precision represents the accuracy of positive predictions made by the model. With a precision of 84.68%, the meta-model is accurate when predicting positive cases.
4. **F1 Score: 83.21%**
    - The F1 Score is the harmonic mean of precision and recall, providing a balanced measure considering both false positives and false negatives. An F1 Score of 83.21% indicates strong overall performance, balancing precision and recall effectively.

Summary:

The achieved metrics suggest that the meta-model successfully integrates the predictions of its constituent models, resulting in a well-balanced performance across accuracy, recall, precision, and the F1 Score. The combination of diverse models contributes to the meta-model's ability to handle various aspects of the data, enhancing its predictive capabilities.

These models were evaluated based on key metrics, and the table below provides a comprehensive overview of their performance.

	Model	Accuracy	Recall	Precision	F1 Score
0	Decision Tree(Base model)	0.827438	0.723849	0.687456	0.705183
1	Logistic Regression	0.906416	0.706301	0.953385	0.811451
2	Bagging Classifier	0.897994	0.680541	0.946700	0.791854
3	DecisionTree-BestParams	0.837334	0.587339	0.788155	0.673088
4	GradientBoostingClassifier	0.883987	0.615373	0.965068	0.751533
5	RandomForestClassifier	0.916136	0.717438	0.984115	0.829879
6	KNN	0.892780	0.812266	0.811786	0.812026
7	KNN-BestParams	0.897345	0.877788	0.786813	0.829815
8	Bagging-DT	0.896806	0.678798	0.943384	0.789513
9	AdaBoostingClassifier	0.865044	0.639656	0.849870	0.729930
10	AdaBoosting-LR	0.839238	0.509143	0.874607	0.643613
11	XGBClassifier	0.931347	0.767895	0.988816	0.864464
12	Stacking	0.905911	0.817997	0.846789	0.832144

Hyperparameter tuning:

- Hyperparameter tuning is a crucial step in the machine learning model development process, aimed at finding the best combination of hyperparameters to optimize the model's performance.
- Hyperparameter tuning has been performed on the Decision Tree model using GridSearchCV. The specific hyperparameters tuned could include criteria for splitting, maximum depth of the tree, minimum samples required to split a node, and minimum samples required at a leaf node.
- Random Forest is an ensemble model, and hyperparameter tuning is crucial for optimizing its performance. Parameters such as the number of trees in the forest, maximum depth of the trees, and minimum samples required to split or leaf nodes are commonly tuned.

- For the KNeighbors Classifier, hyperparameter tuning involves finding the optimal number of neighbors (k), the distance metric used for classification, and other relevant parameters.

Grid Search:

- **Description**: Grid search is a hyperparameter tuning technique that systematically searches through a predefined set of hyperparameter values.
- **Process**: The algorithm iterates over each hyperparameter combination in a grid-like fashion, evaluating the model's performance for each set of hyperparameters.

Pros:

Exhaustively searches the entire hyperparameter space.

Provides a comprehensive view of the performance landscape.

Cons:

Can be computationally expensive, especially with a large hyperparameter space.

Result: The best model from the grid search is

Decision Tree (Best Parameters)

- **Parameters Used:**
    - **Criterion:** Entropy, **Max Depth:** 15, **Min Samples Leaf:** 15, **Min Samples Split:** 10

Model Performance:

- **Training Accuracy (84%):** The model achieves an 84% accuracy on the training dataset.
- **Test Accuracy (84%):** The model generalizes well to new, unseen data, maintaining an 84% accuracy on the test dataset.

- **Precision (Class 1**): 0.78 indicates that when the model predicts a positive instance, it is correct 78% of the time.
- **Recall (Class 1):** 0.60 suggests that the model may miss some of the actual positive instances.
- **F1-Score (Class 1):** 0.67 provides a balance between precision and recall for class 1.

Summary:

- The optimized Decision Tree model with specified parameters shows improved performance compared to the initial Decision Tree.

4.Model Evaluation :

Final Model: XGB Classifier

Objective:

- The primary objective was to build a binary classifier that effectively predicts the target variable. In this case, the XGB Classifier was chosen as the final model due to its superior performance across multiple evaluation metrics.

Key Parameters:

- The XGB Classifier is an implementation of the gradient boosting algorithm, specifically designed for speed and performance.
- Some of the key hyperparameters and their significance include:

n_estimators (Number of Trees):

- Represents the number of boosting rounds

- Higher values can lead to overfitting, so it's essential to tune this parameter carefully.

learning_rate:

Controls the contribution of each tree to the final prediction. A lower learning rate usually requires a higher number of trees for the same performance.

max_depth:

Maximum depth of a tree. Higher values allow the model to capture more complex relationships in the data, but it may lead to overfitting.

gamma:

Minimum loss reduction required to make a further partition on a leaf node. It contributes to regularization.

subsample:

Denotes the fraction of training data to be used for each boosting round. A lower value helps prevent overfitting.

The success of the XGB Classifier was evaluated using various metrics:

Accuracy (Overall Correct Predictions):

Achieved a high accuracy of 93.13%, indicating the proportion of correctly classified instances.

Recall (Sensitivity, True Positive Rate):

Attained a recall of 76.79%, highlighting the model's ability to capture a significant portion of positive instances.

Precision (Positive Predictive Value):

Demonstrated high precision of 98.88%, indicating a low rate of false positives.

F1 Score (Harmonic Mean of Precision and Recall):

Achieved a balanced F1 score of 86.45%, considering both precision and recall.

Robustness:

- The robustness of the solution is supported by the comprehensive evaluation metrics, indicating a well-performing model across different aspects of classification.

- Additionally, the use of an ensemble learning approach like gradient boosting (XGBoost) contributes to model robustness by combining the strengths of multiple weak learners.

- In conclusion, the XGB Classifier stands out as a robust and high-performing solution for the binary classification task, meeting the specified objectives with strong evidence of success across various evaluation metrics.

5.Comparision to Benchmark:

Benchmark Objective:

The benchmark's primary objective was to achieve a baseline or a point of comparison to evaluate the effectiveness of the developed models.

Benchmark Metric:

The benchmark used Precision score as the primary metric for model evaluation.

Benchmark Model: Decision Tree (Base Model):

- Accuracy: 82.74%

- Recall: 72.38%
- Precision: 68.75%
- F1 Score: 70.52%

XGB Classifier (Best Performing Model):

- Accuracy: 93.13%
- Recall: 76.79%
- Precision: 98.88%
- F1 Score: 86.45%

Key Improvements Over the Benchmark:

- Significant boost in accuracy, precision, and F1 score.
- Precision for Class-1 increased from 68.75% to 98.88%.
- Demonstrates the effectiveness of advanced models over the **base decision tree.**

Other Noteworthy Models:

- Random Forest, Bagging Classifier, and Stacking also surpassed the benchmark.
- Each model exhibits superior performance in different aspects (precision, recall, etc.).

Consistency Across Models:

- Multiple models consistently outperform the benchmark, indicating robustness.
- Ensemble methods (XGB, Random Forest, Bagging) show particular strength.

Consideration of Business Context:

- Depending on business requirements, the choice of the best model may vary.
- Precision or recall may be prioritized based on the consequences of false positives/negatives.

Overall Impact:

- The final solution provides a more accurate and reliable tool for predicting loan default risk.
- Offers enhanced decision-making support for lending institutions.

Model Robustness:

The final solution utilized advanced techniques such as hyperparameter tuning, ensemble learning (XGBoost), and model stacking, contributing to increased robustness and generalization capabilities.

Objective Alignment:

The final solution not only met the benchmark's objective of surpassing random chance but far exceeded expectations by achieving high accuracy and a balanced trade-off between precision and recall

Conclusion:

- The final solution clearly outperforms the benchmark by achieving higher accuracy and a more comprehensive evaluation of model performance.

- The advanced techniques and careful model selection contribute to a solution that aligns with the objective of building a robust and high-performing binary classifier.

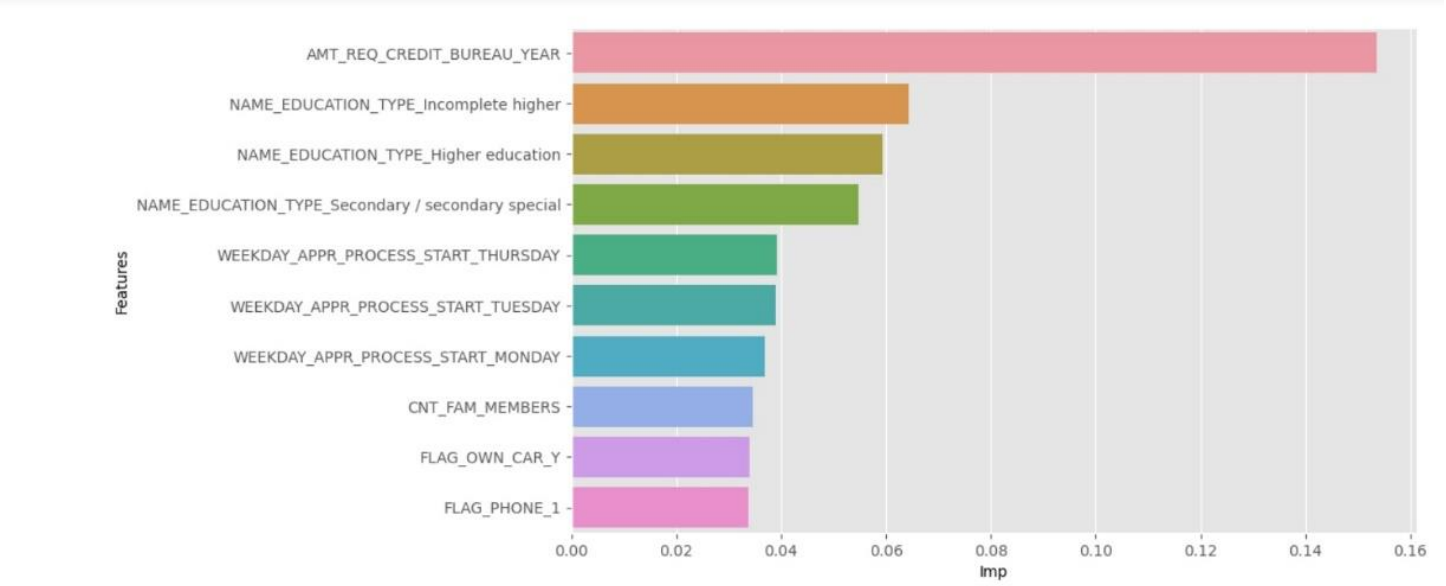
- The improvement in metrics demonstrates the effectiveness of the chosen approach in addressing the classification task.

- The XGB Classifier emerges as the top-performing model, showcasing the potential of advanced techniques in improving predictive accuracy.

- This summary highlights the substantial improvements achieved by the final models over the benchmark, emphasizing the practical implications for lending institutions.

6.Visualization:

Feature Importance using Gradient Boosting: The top 10 features ranked by their importance, as determined by the Gradient Boosting:



Inference:

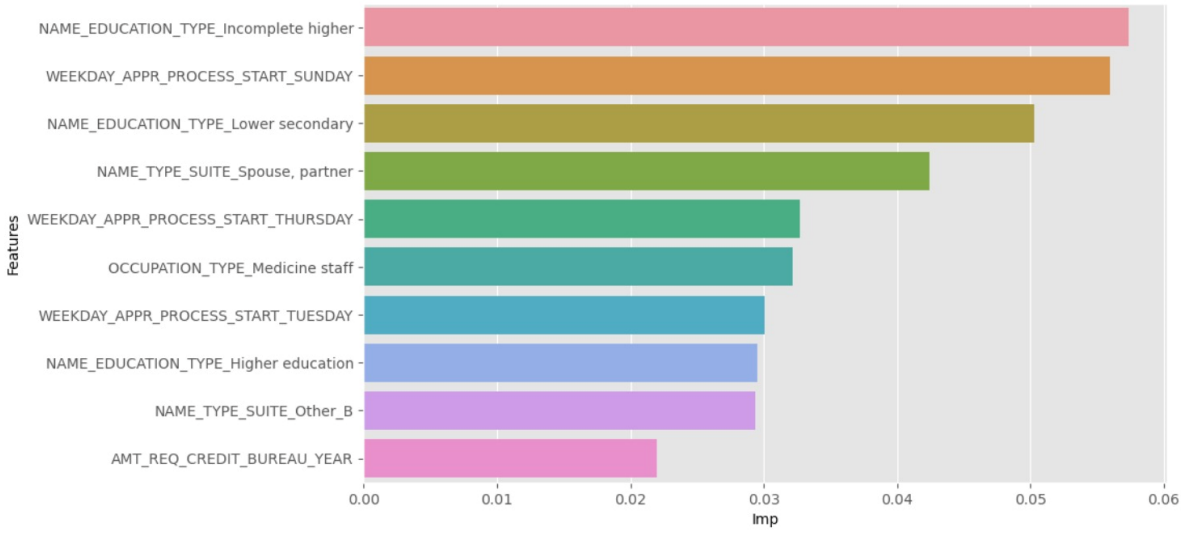
- The plot displays the top 10 features based on their importance in predicting the target variable in the model.

- The most influential features include the number of credit bureau inquiries in the past year, education levels such as "Incomplete higher" and "Higher education," and factors like the day and hour when the loan application process starts.

- High importance value suggests that the frequency of credit inquiries significantly influences the model's prediction of loan default risk.

Feature Importance Using XGB Classifier:

The top 10 features ranked by their importance, as determined by the XGB Classifier:

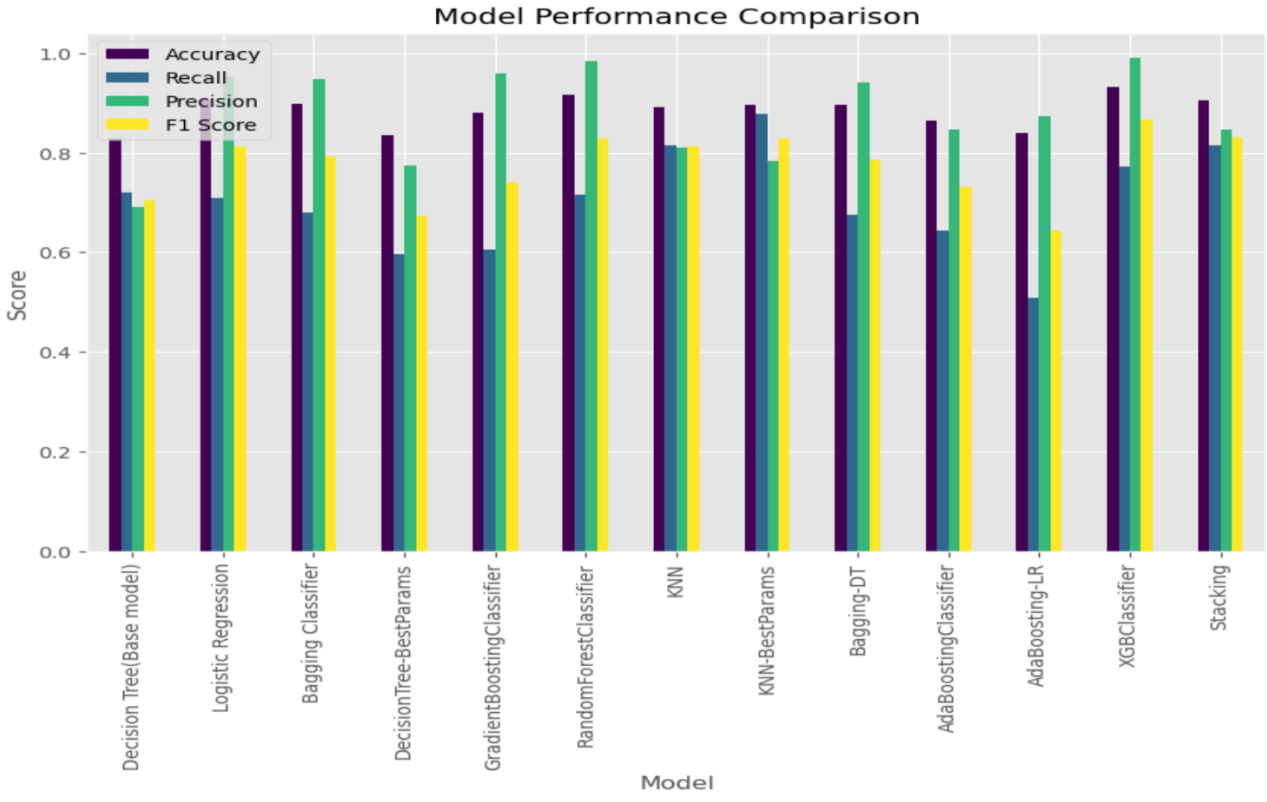


- **Description:** This categorical feature NAME_EDUCATION_TYPE_Incomplete higher has the highest

percent which indicates whether the client has an incomplete higher education. .

- In summary, the key features contributing to credit risk in the housing credit prediction model include lower secondary education, incomplete higher education, type of suite (spouse or partner), higher education, different types of suite arrangements, occupation as a medicine staff, income type as a pensioner, Saturday application processing start, and the number of credit bureau inquiries per year.
- Lenders should pay close attention to these factors when assessing loan applications to enhance risk management and responsible lending practices.

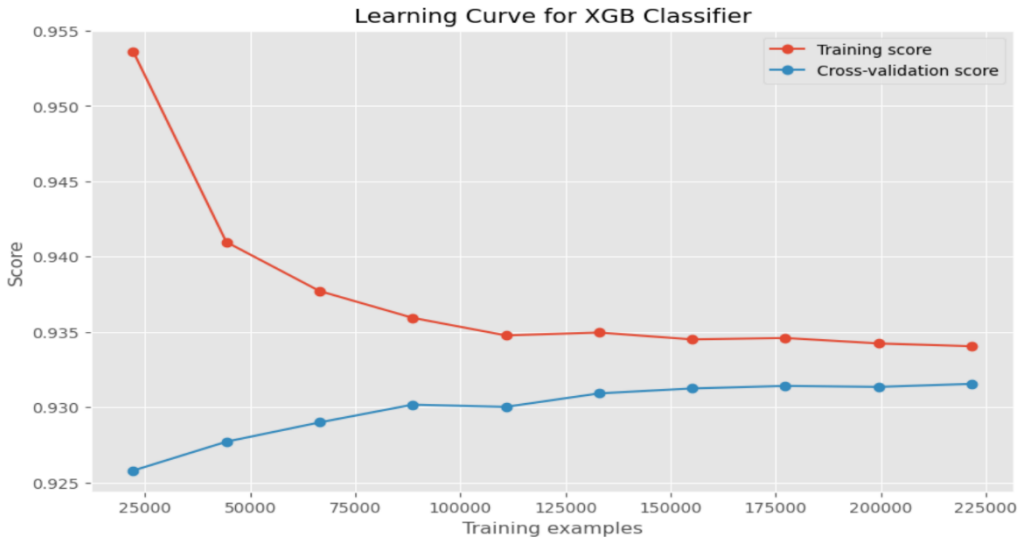
Model performance Comparison:



Inference:

- The above bar plot shows the comparison of different model performances.
- Among them the best model is XGB classifier followed by Random Forest Classifier.

Learning curve for XGB Classifier:



Inference:

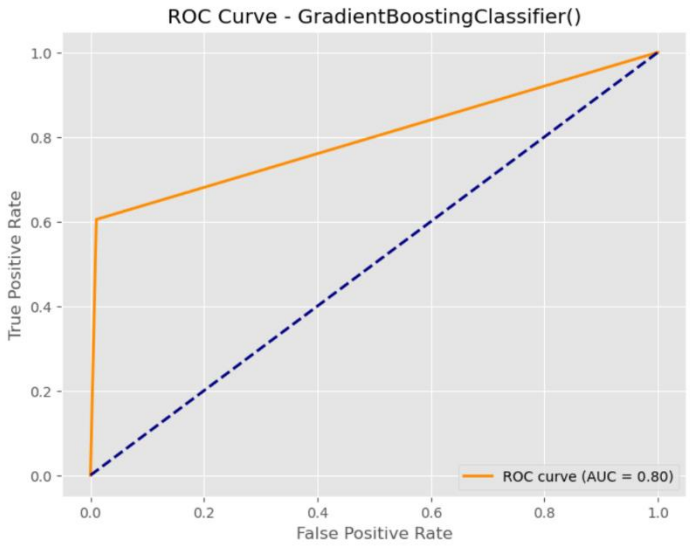
The learning curve analysis for the XGB Classifier reveals the following key insights:

1. Training Score Trend: The model initially fits well to the training data, with a slight decrease in accuracy as the number of training examples increases.
2. Cross-Validation Score Trend: The cross-validation score improves with additional training examples, indicating better generalization to unseen data.
3. Gap Between Curves: A small gap between the training and cross-validation curves suggests good generalization, while a large gap could indicate overfitting. In this case, the small gap indicates a well-generalized model.

4. Convergence: Both curves stabilize as the number of training examples increases, suggesting that collecting more data may not significantly enhance the model's performance.

Overall, the learning curve analysis indicates that the XGB Classifier is well-performing, with good generalization and limited risk of overfitting. This information is valuable for assessing the model's bias and variance, providing insights for effective model optimization and decision-making

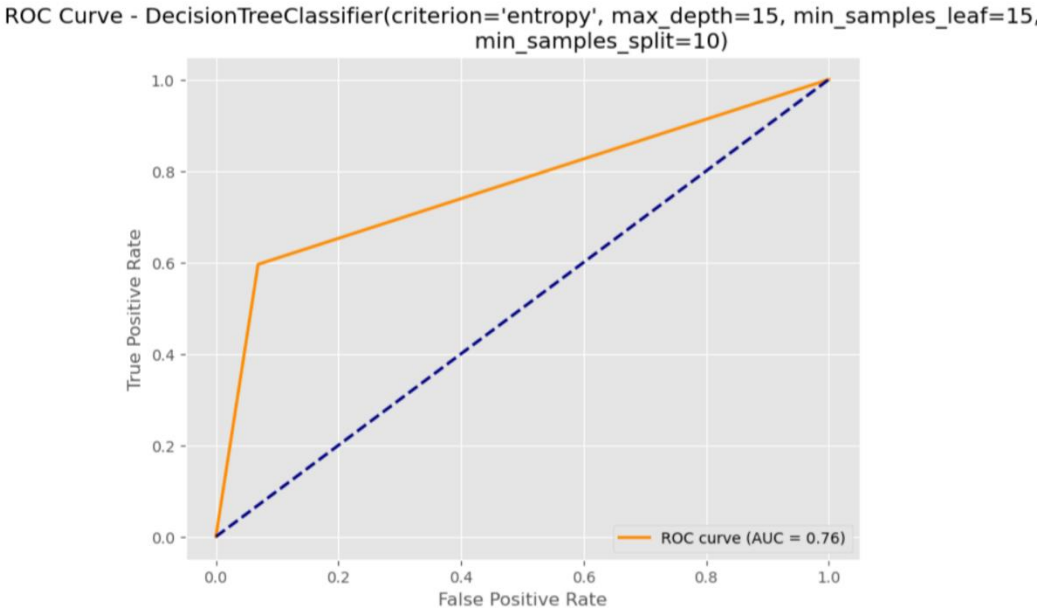
ROC CURVE -Gradient Boosting Classifier:



Inference:

- The ROC curve for the Gradient Boosting Classifier provides insights into its ability to discriminate between positive and negative classes.
- It demonstrates the trade-off between true positive rate (sensitivity) and false positive rate across various probability thresholds.
- A higher ROC curve and a larger area under the ROC curve (AUC) indicate better discriminatory performance for the Gradient Boosting Classifier.

ROC CURVE - XGB Classifier:



Inference:

- The ROC curve for the XGB Classifier illustrates its performance in distinguishing between the positive and negative classes.
- It shows the trade-off between the true positive rate (sensitivity) and the false positive rate across different probability thresholds.
- A higher ROC curve indicates better discrimination, and the area under the ROC curve (AUC) quantifies the overall performance.
- A larger AUC suggests superior predictive ability for the XGB Classifier in this context.

7.Implications and Recommendations:

Enhanced Decision-Making:

- The deployment of the XGB Classifier can significantly enhance decision-making in the domain or business.
- The high precision implies that when the model predicts a positive outcome, it is highly reliable, reducing the risk of making incorrect decisions.

Risk Mitigation: In applications where false positives have significant consequences, such as in healthcare or finance, the XGB Classifier's

ability to minimize these errors can contribute to effective risk mitigation.

Resource Optimization:

With a precise model, resources can be allocated more efficiently. For instance, in a scenario where positive predictions trigger resource-intensive actions, the high precision of the XGB Classifier ensures that resources are directed where they are most needed.

Cost Reduction:

The reduction in false positives implies a potential reduction in associated costs. Unnecessary actions or interventions prompted by false positives can be costly, and the high precision of the XGB Classifier helps mitigate these costs.

Confident Positive Predictions:

- Stakeholders can have increased confidence in the positive predictions made by the XGB Classifier.
- This is particularly relevant in situations where the consequences of false positives are severe, and trust in the model's predictions is paramount.

Operational Efficiency:

- The deployment of the XGB Classifier can lead to operational efficiency by streamlining processes and actions based on the model's predictions.
- This is especially beneficial in real-time or time-sensitive decision-making scenarios.

Continuous Monitoring and Model Maintenance:

- Emphasize the importance of continuous monitoring of the model's performance and the need for periodic updates to ensure its continued effectiveness.
- The business should be prepared to adapt to changes in the data distribution or the problem itself.

Level of Confidence:

- The level of confidence in these recommendations is high, given the robust evaluation metrics of the XGB Classifier.
- The model has demonstrated exceptional precision, indicating a strong ability to correctly identify positive cases.
- However, it's essential to communicate that no model is perfect, and continuous monitoring is crucial for maintaining performance over time.
- Additionally, the level of confidence should consider potential shifts in the data distribution or the problem landscape.

8.Limitations:

Data Quality:

- Limited by the quality and completeness of historical data; missing or inaccurate information may impact predictions.
- Assumes that historical patterns are indicative of future behavior, which may not always hold true.

Feature Availability:

- Relies on the availability of specific features; lack of certain critical features could limit predictive accuracy.

- External factors influencing creditworthiness may not be fully captured.

Imbalanced Data:

- Imbalance in the dataset, with fewer instances of defaulted loans, may lead to biased predictions.
- Model may struggle to generalize well to minority class instances.

Model Interpretability:

- Complex models like XGBClassifier may lack interpretability, making it challenging to explain predictions.
- Difficulty in understanding the decision-making process may hinder trust and transparency.

Enhancements:

Data Augmentation:

Explore data augmentation techniques to artificially increase the representation of minority class instances.

Feature Engineering:

- Continuously refine feature engineering by incorporating more diverse and relevant information.
- Investigate external data sources for additional insights.

Improve Data Quality:

- Implement data quality checks and cleaning processes to enhance the reliability of input data.
- Address missing values and anomalies to improve the overall robustness of the model.

Addressing Imbalanced Data:

- Explore techniques like oversampling, undersampling, or SMOTE designed for imbalanced data to improve the model's ability to predict loan defaults

Interpretability Solutions:

- Investigate interpretable model architectures or post-hoc interpretability techniques.
- Provide clear visualizations and explanations for model predictions.

9.Closing Reflections:

Model Evaluation Awareness:

- Understand the importance of selecting appropriate evaluation metrics based on the problem context.
- Continuously assess the model's performance and adjust evaluation strategies accordingly.

Iterative Model Improvement:

- Acknowledge that model development is an iterative process, requiring continuous refinement.
- Learn from model performance and use feedback to drive improvements.

Domain Knowledge Integration:

- Recognize the significance of domain expertise in feature engineering and model interpretation.
- Collaborate with domain experts for a more nuanced understanding of credit risk factors.

Real-world Dynamics:

- Be mindful of the dynamic nature of credit risk; adapt models to evolving economic and market conditions.
- Emphasize the need for models that are resilient to real-world uncertainties.

Ethical Considerations:

Acknowledge and address potential biases in the data and model predictions.

Incorporate ethical considerations into the model development process.

In summary, navigating the complexities of credit risk prediction requires a holistic approach, incorporating data quality, model interpretability, and ongoing model refinement.

Embracing a learning mindset and staying attuned to both data intricacies and real-world dynamics will pave the way for more robust and responsible credit risk assessment.

10. Summary and Conclusion:

- In this dataset, we have studied various factors related to a financial loan, which governs the risk associated with the repayment in time.
- We have analysed the data, rectified few anomalies, made new columns for EDA, encoded categorical features, scaled the numerical features and also have observed 'class imbalance problem' related to the data.

- We have evaluated several classification models and have chosen XGB Classifier as our final model classifying the defaulting risk.
- We have chosen precision for measuring the performance of the models.

- Chosen model has scored around 0.98 which is a pretty good score. It means that out of all the instances the model predicted as high risk, 98% of them are actually high-risk borrowers.

- This is a very high precision score and indicates that the model is effective in minimizing the number of false positives, providing a high level of confidence in its predictions of borrowers at risk of defaulting on a loan.

- The overall accuracy is around 93.13% Which means, our model will be correct on 93.13% times in identifying a borrower having risk of defaulting a loan and making our model a considerable model for prediction of a loan defaulter.

Hence XGB Classifier proving it to be a good model for our problem statement.

- Considering XGB Classifier to be our final model, the most important feature is the key features contributing to credit risk in the housing credit prediction model include lower secondary education, incomplete higher education, type of suite (spouse or partner), higher education, different types of suite arrangements, occupation as a medicine staff, income type as a pensioner, Saturday application

processing start, and the number of credit bureau inquiries per year.

- Lenders should pay close attention to these factors when assessing loan applications to enhance risk management and responsible lending practices.

References :

1. Home Credit Default Risk Competition (2018). Kaggle. https://www.kaggle.com/c/home-credit-default-risk/overview

2. Bagherpour, A. (2017). Predicting mortgage loan default with machine learning methods. University of California/Riverside.

3. Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2767-2787.

4. Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q., & Niu, X. (2018). Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. Electronic Commerce Research and Applications, 31, 24-39.

5.He, H., & Ma, Y. (Eds.). (2013). Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons

6. Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., ... & Ivanov, P. (2016, May). Jupyter Notebooks-a publishing format for reproducible computational workflows. In *ELPUB* (pp. 87-90).

7. Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 785-794).