Loan Default Prediction: A Comparative Analysis of Machine Learning Models

Introduction:

Precise forecasting of loan default is essential to risk management and decision-making in the financial sector. In this project, a dataset comprising several customer variables is used to build machine learning models for loan default prediction. The effectiveness of several machine learning algorithms was assessed and looked for trends and variables impacting loan default by utilizing predictive modelling approaches.

Throughout the analysis, various questions were attempted to answer:

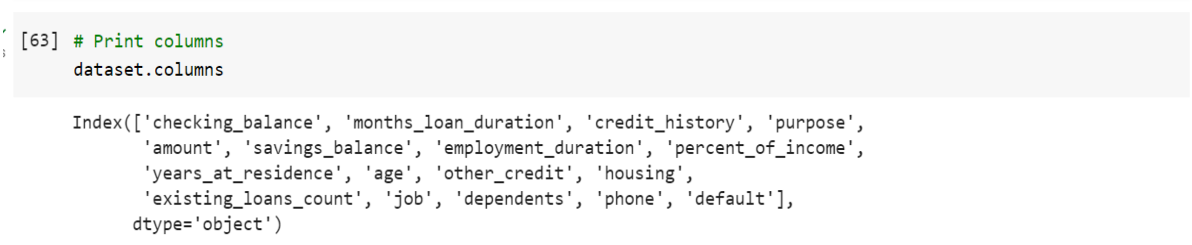
What are the key variables that determine the chance of loan default?
What is the relationship between the likelihood of loan default and various customer factors including age, income, and credit history?
Are there any noteworthy correlations, if any, between loan default and the customer features?
Based on the provided features, which machine learning model best predicts loan defaults?

Methods and Materials:

To acquire a better understanding of the dataset, the research with exploratory data analysis (EDA) was done. The distribution of both numerical and categorical features was represented graphically by box, bar, and histogram plots. statistical tests were also used like ANOVA and the chi-square test to find important indicators of loan default.

1) Data observation:

The dataset contains several features that describe the financial and personal attributes of customers. These include age, income, credit history, loan amount, loan duration, and employment status. The target variable "default," which shows whether or not a consumer defaulted on their loan, is also included in the dataset.



```
[63] # Print columns
     dataset.columns

     Index(['checking_balance', 'months_loan_duration', 'credit_history', 'purpose',
            'amount', 'savings_balance', 'employment_duration', 'percent_of_income',
            'years_at_residence', 'age', 'other_credit', 'housing',
            'existing_loans_count', 'job', 'dependents', 'phone', 'default'],
           dtype='object')
```

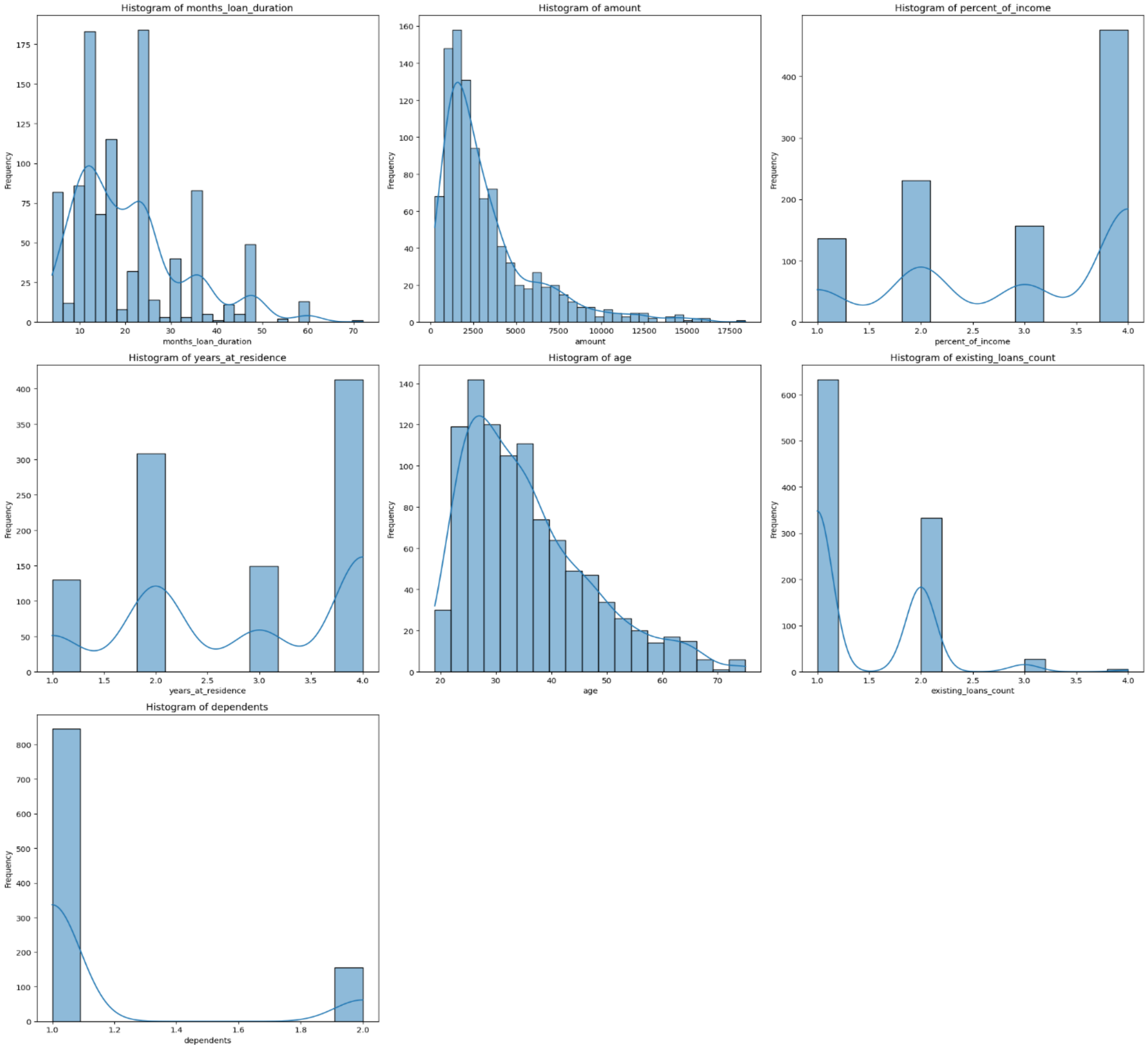
The dataset includes both category and numerical variables, necessitating the use of various modelling and analytic techniques. exploratory data analysis (EDA) will be performed to obtain understanding of the distribution and correlations of these features prior to moving further with model construction.

the dataset's numerical and categorical properties were separated in order to examine it efficiently. While categorical features include features like credit history and employment status, numerical features include age, income, loan amount, and loan duration. Because to this differentiation, proper analytic methods can be applied to each kind of feature.

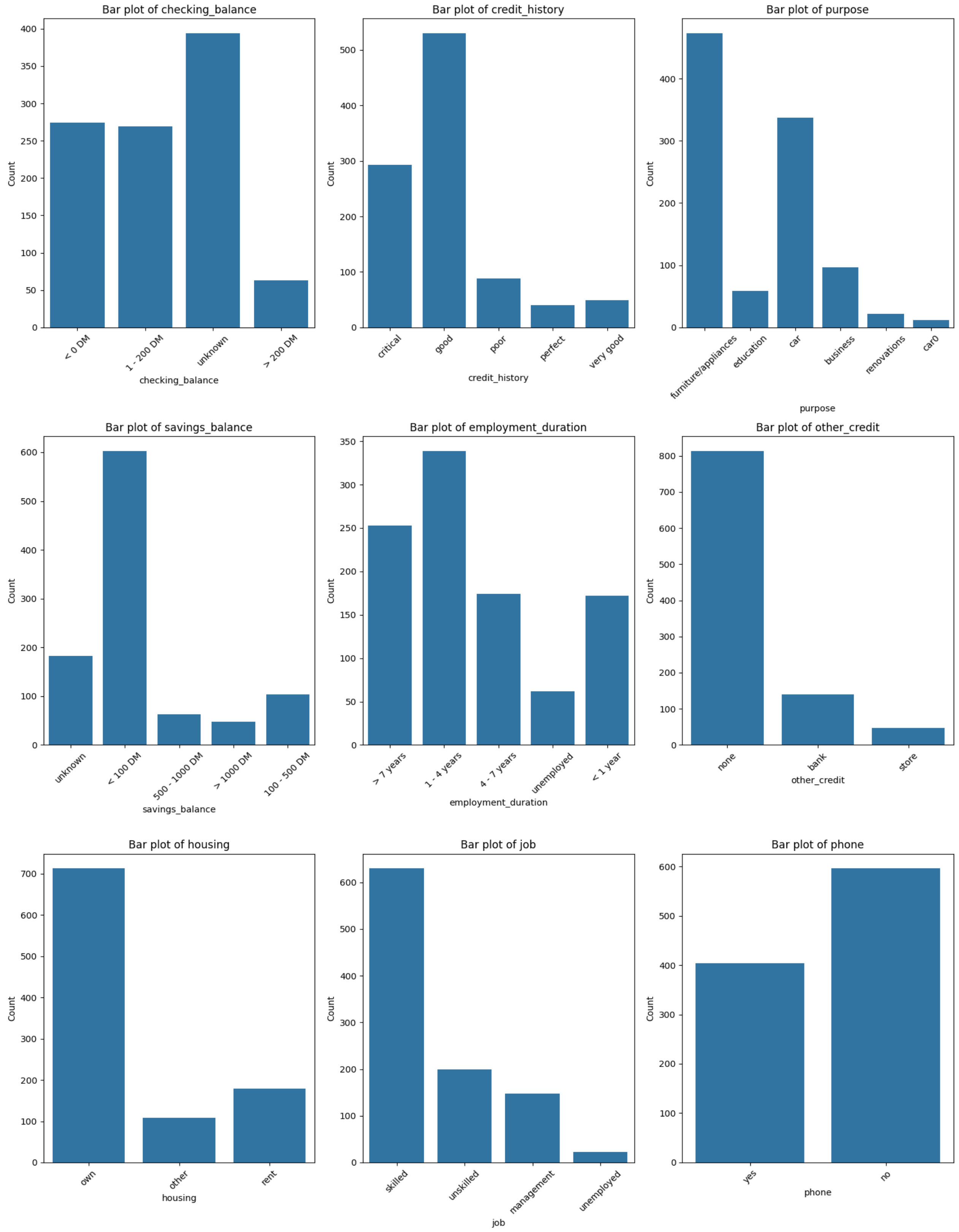
2) Data exploration and visualization

To comprehend the distributions of numerical parameters like age, income, loan amount, and loan duration and spot any outliers, univariate analysis will be done. Histograms, box plots, and density plots will be visualized in order to identify trends and evaluate the data's central tendency and spread.

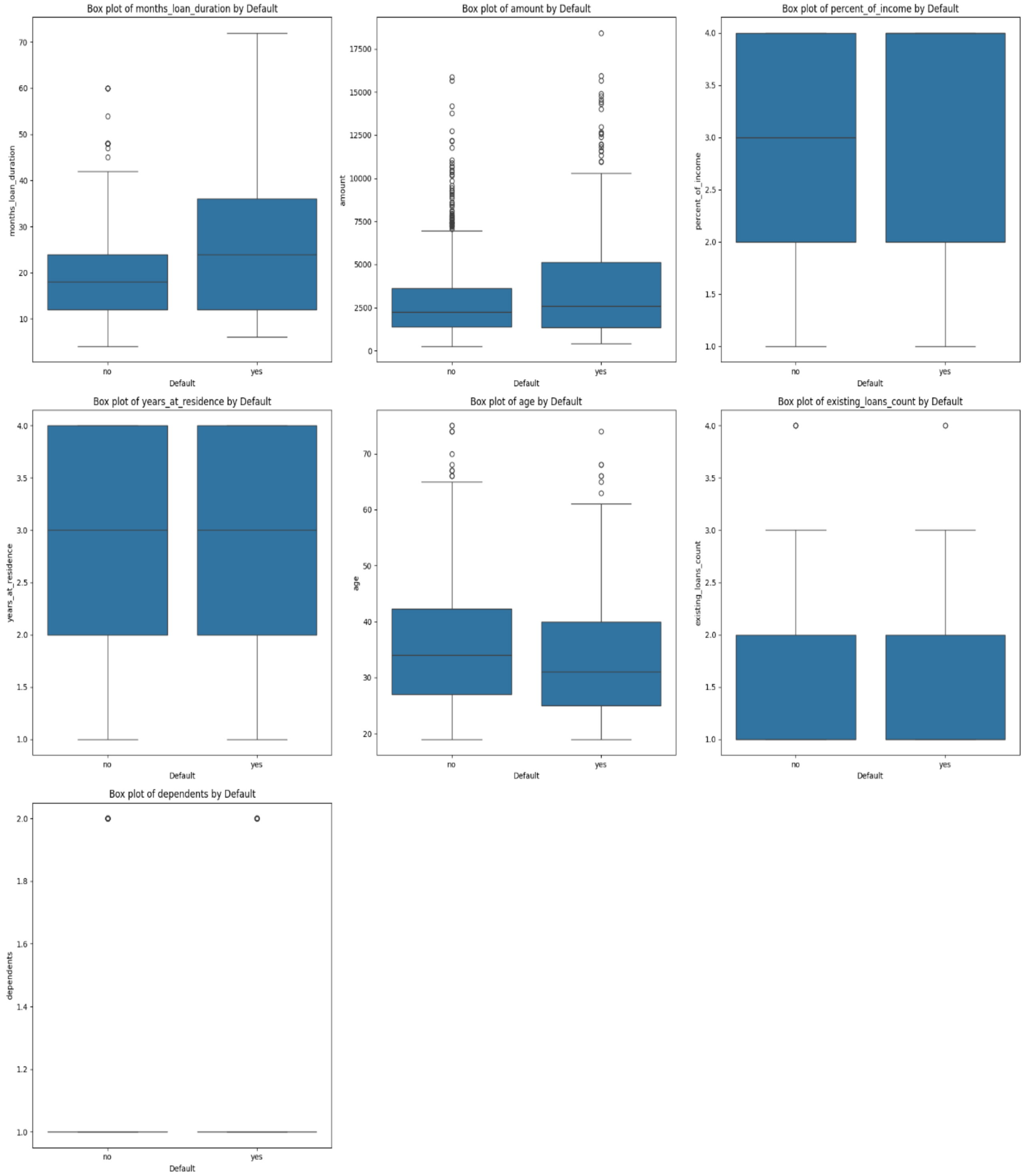
Below is the visualization of univariate analysis of numerical features



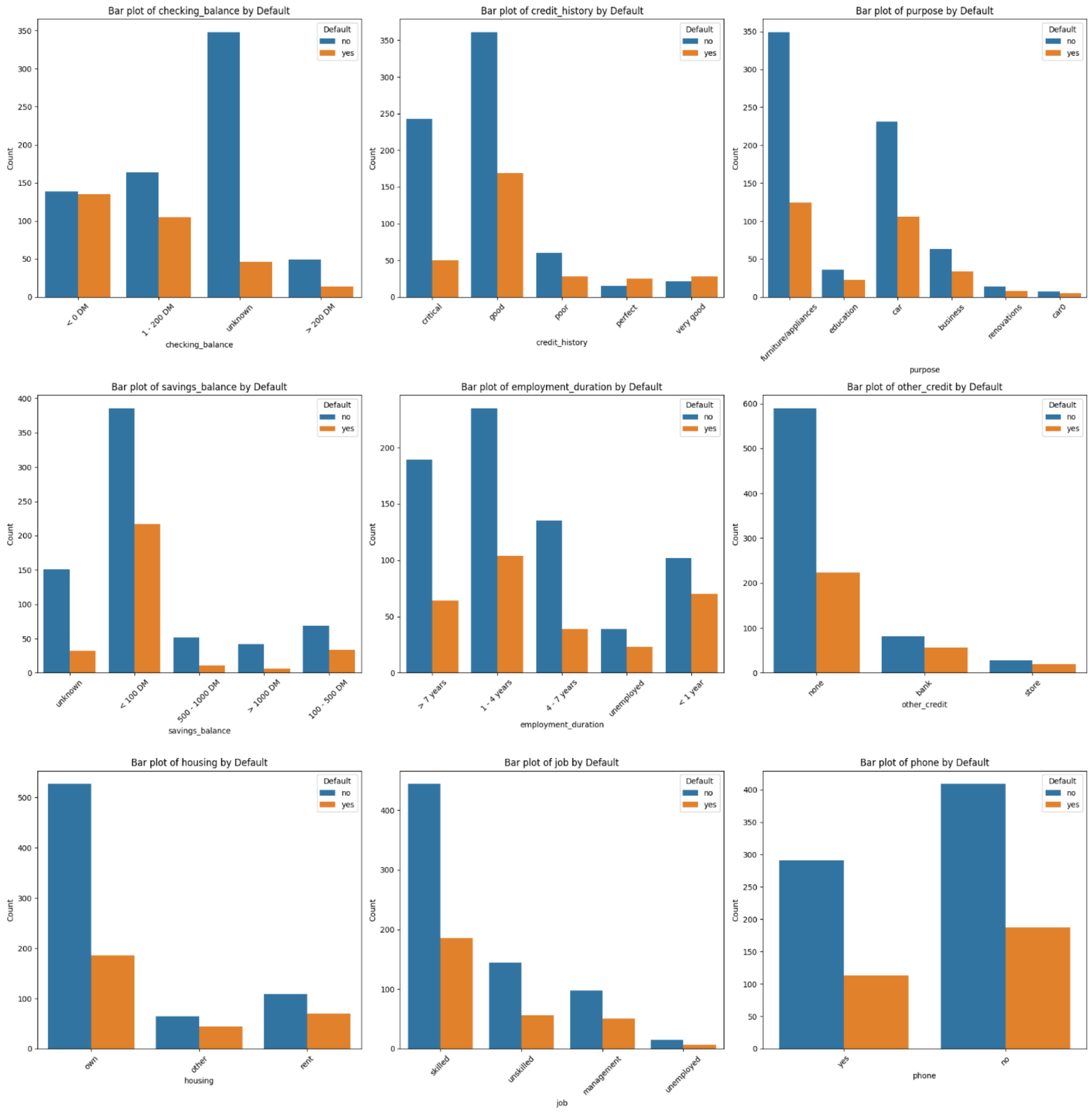
Below is the visualization of univariate analysis of categorical features



Below is the visualization of bivariate analysis of numerical features vs. target variable



Below is the visualization of bivariate analysis for categorical features vs. target variable



3) Identification of predictor variables using statistical tests

Correlation analysis for numerical features:

Predictor variables from correlation analysis are 'months loan duration', 'amount', and 'age'.

ANOVA F-test for numerical features:

among all the predictor variables resulted from Anova test 'percent of income' is having a greater p-value i.e. 0.022. Also 'percent of income' is not resulted from correlation test as a predictor. Taking these observations into account a finalised predictor list from numerical features would be numerical predictors are 'months loan duration', 'amount', and ‘age'

Chi-Square test for categorical features:

From Chi-Square test categorical predictors are 'checking balance', 'credit history', 'savings balance', 'employment duration', 'other credit', and 'housing'.

Therefore, the final predictors list includes 'age', 'amount', 'months loan duration', 'checking balance', 'credit history', 'savings balance', 'employment duration', 'other credit', and 'housing'. Whereas, ‘default’ is the response variable.

4) Data pre-processing

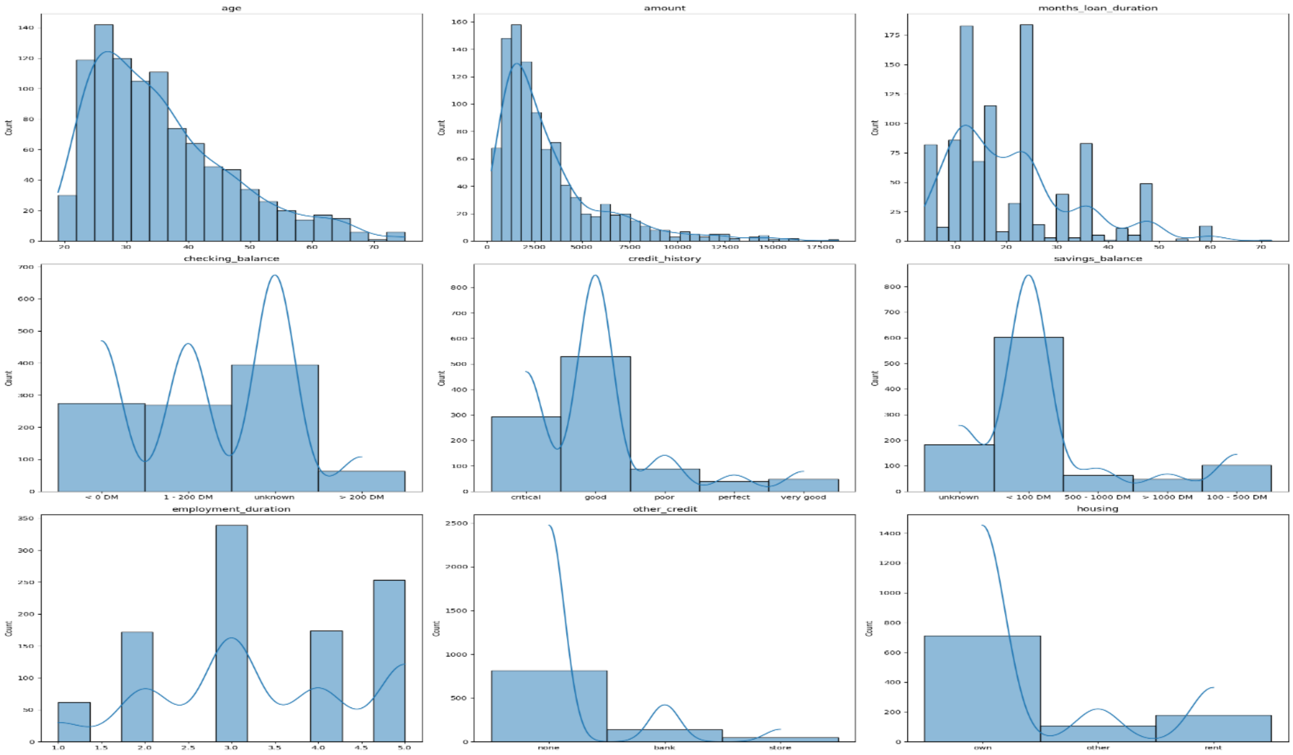
missing values were handled, scaled numerical features, and used one-hot encoding to transform categorical data into numerical format as part of the preprocessing step. Next, a number of machine learning models were trained and assessed, such as the Gradient Boosting Classifier, Random Forest, Support Vector Machine, and Logistic Regression models.

5) Data Preparation: splitting into training and testing sets

Standardization or Normalization? what to choose?

Predictors and target variables are all set. But before proceeding with splitting the data into training and testing sets it's best to standardize or normalize the numerical data depending on the nature of data to ensure that all features have the same scale, making the algorithm work effectively. After this step data is split into training and testing sets in 80:20 ratio.

Let’s observe whether the data is Gaussian distributed



A wave pattern is seen in the histograms, it indicates that data does not follow a Gaussian distribution. Given this observation, normalization (MinMaxScaler) would be more appropriate for this dataset, as it does not assume any specific distribution of the data.

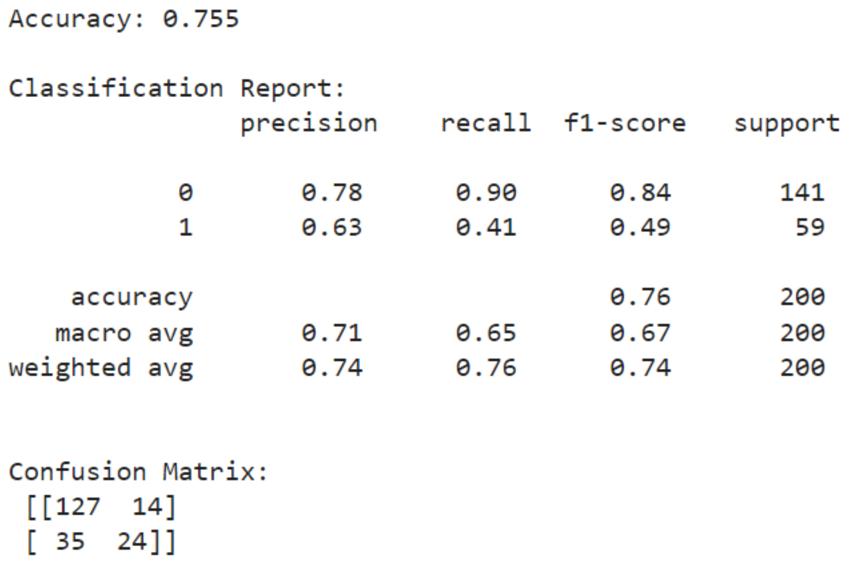
6) Model training and evaluation

Depending on the nature of the data, size of the dataset and the characteristics of data, a model should be chosen. Predicting whether a customer will default or not" is a classification problem. For classification problems like these, Following models can be used.

* Logistic Regression
* Decision Trees
* Random Forest
* Support Vector Machines (SVM)
* Gradient Boosting Machines (e.g., XGBoost, LightGBM, CatBoost)

A) Logistic Regression:

Below are results generated from Logistic Regression model



```
Accuracy: 0.755

Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.90      0.84       141
           1       0.63      0.41      0.49        59

    accuracy                           0.76       200
   macro avg       0.71      0.65      0.67       200
weighted avg       0.74      0.76      0.74       200


Confusion Matrix:
 [[127  14]
 [ 35  24]]
```

Observations:

Accuracy:

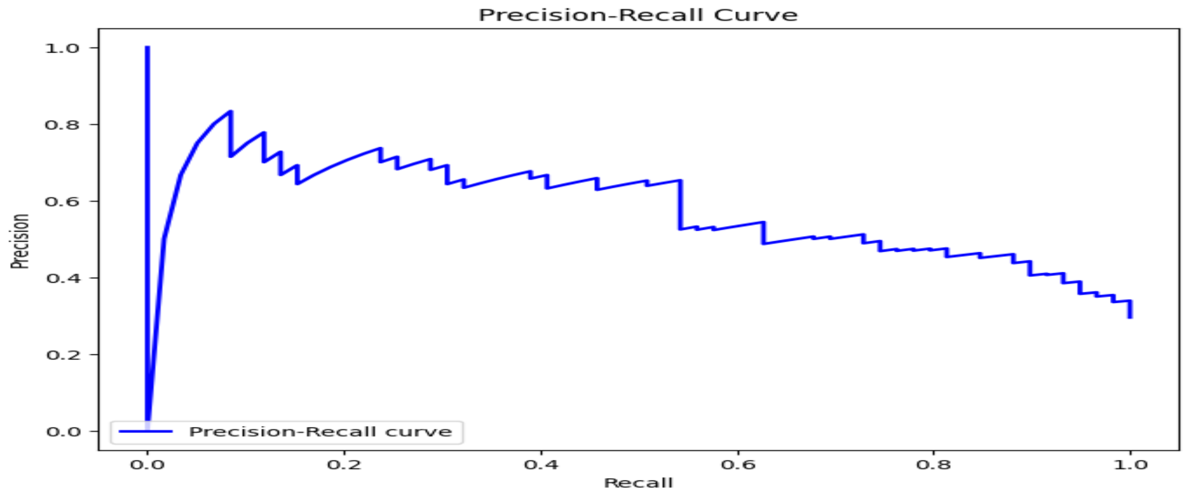
The Logistic Regression model has an accuracy of about 0.755, which means it properly predicts customers' default statuses 75.5% of the time.

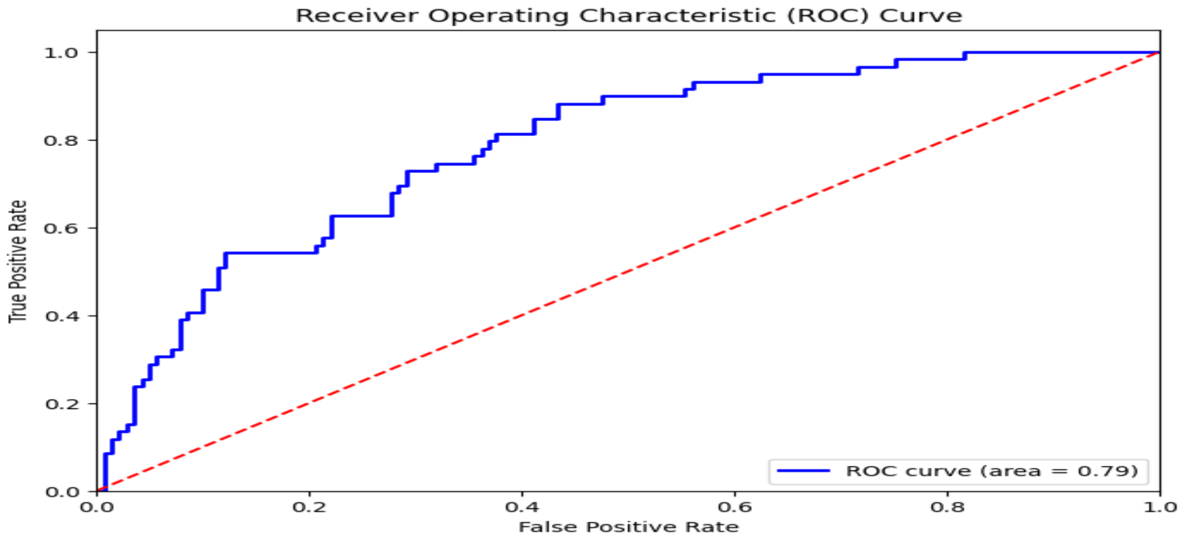
Precision:
Precision for class 0 (non-defaulters) is 0.78, indicating that 78% of customers predicted as non-defaulters are indeed non-defaulters. The precision for class 1 (defaulters) is 0.63, indicating that 63% of the customers projected as defaulters are in fact defaulters. Class 0 has a recall of 0.90, indicating that 90% of non-defaulters are properly anticipated.

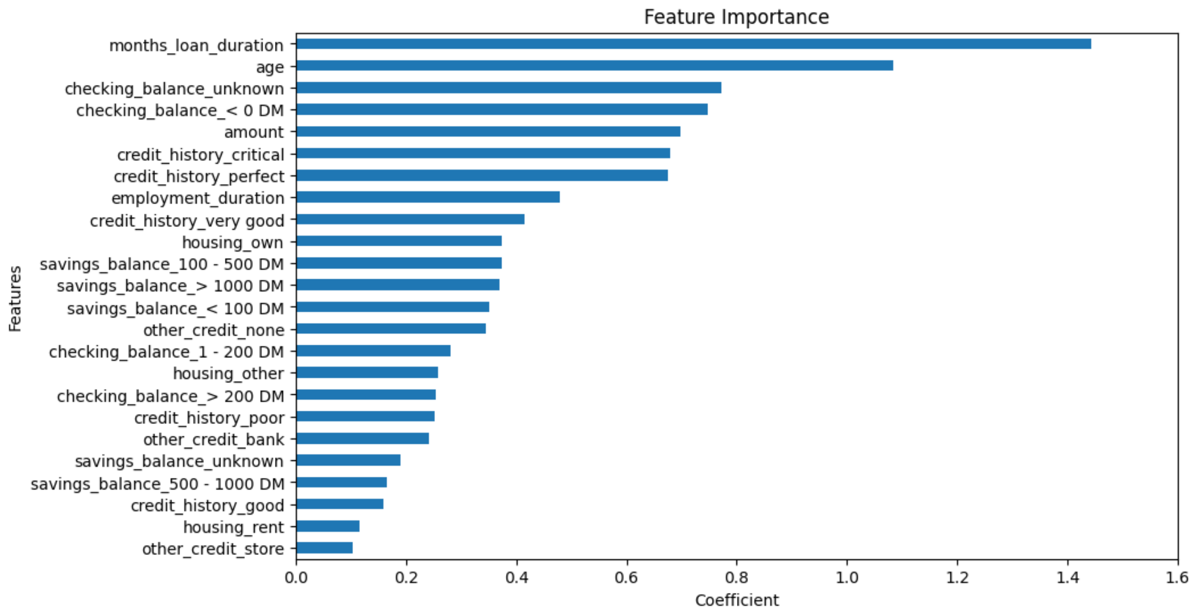
Recall:
Class 0 has a recall of 0.90, indicating that 90% of non-defaulters are properly anticipated. The recall for class 1 is 0.41, which means that 41% of the actual defaulters were properly predicted as defaulters.

F1-score:
The F1-score represents the harmonic mean of precision and recall. It strikes a balance between precision and recall.

Overall, the model appears to perform pretty well. However, given the class imbalance (more non-defaulters than defaulters), other evaluation measures like the ROC-AUC score or the precision-recall curve must be looked at.





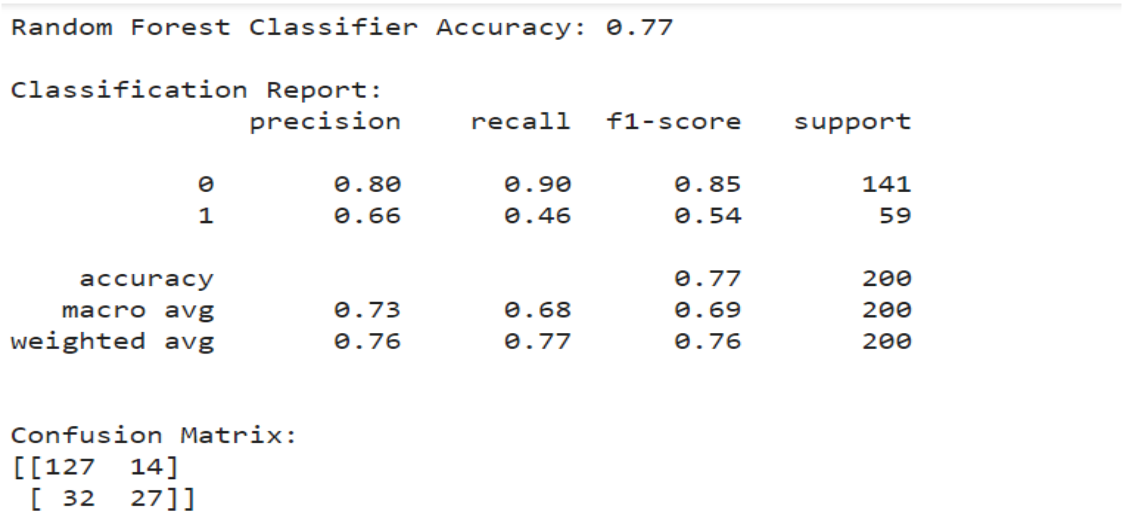


Observations

1. A rough ROC curve suggests that the model's performance may not be optimal. A curve that is mainly above the diagonal line (the line that passes through the origin) indicates that the model outperforms random, but it may not distinguish well between the positive and negative categories. To improve the ROC curve, different algorithms can be given a try.
2. precision-recall curve begins at (0, 1) and declines to (0, 0) before growing and becoming rough indicates that the model initially predicts all instances as positive, resulting in perfect recall but not so good accuracy. This behavior occurs when the model is too optimistic and forecasts everything as positive.

B) Random Forest classifier

Below are the results generated by Random forest classifier



```
Random Forest Classifier Accuracy: 0.77

Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.90      0.85       141
           1       0.66      0.46      0.54        59

    accuracy                           0.77       200
   macro avg       0.73      0.68      0.69       200
weighted avg       0.76      0.77      0.76       200


Confusion Matrix:
[[127  14]
 [ 32  27]]
```

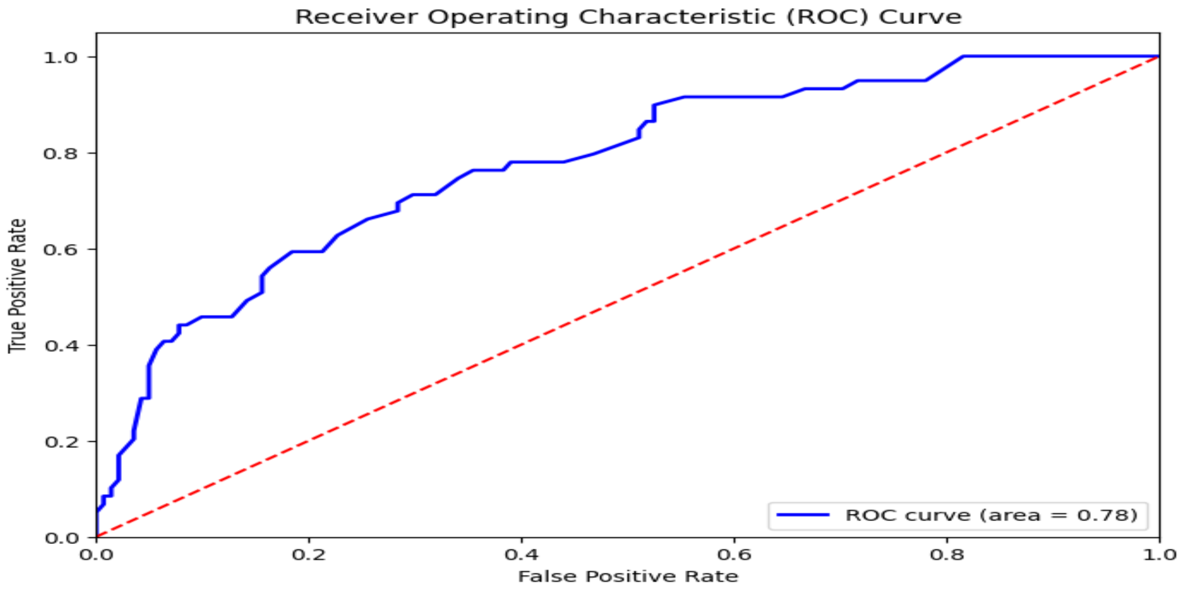
The Random Forest Classifier outperformed the Logistic Regression model. However, there is still room for improvement in the model, particularly in terms of recall for positive classes.

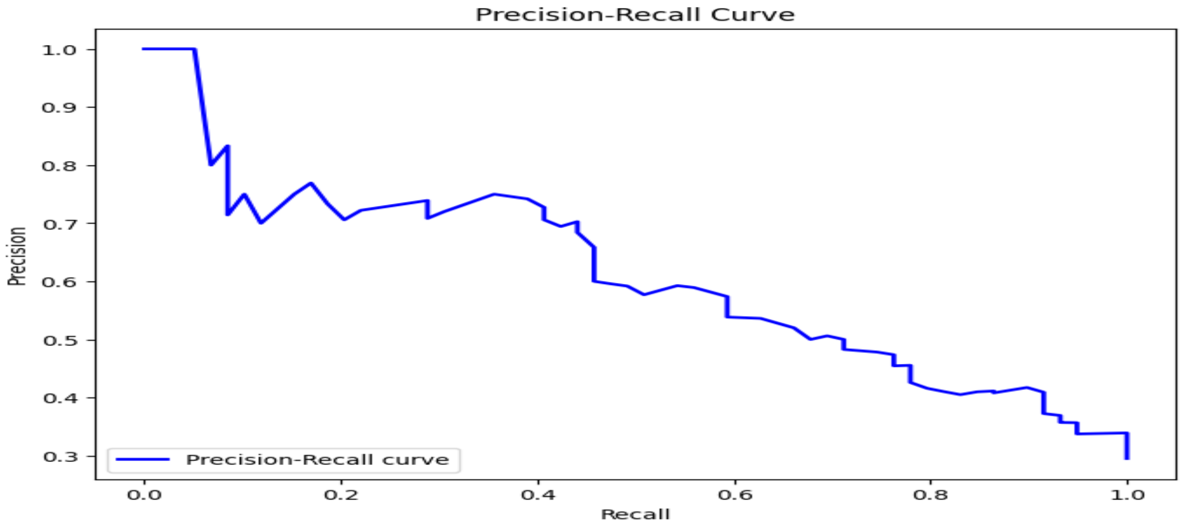
Accuracy:
The Random Forest Classifier has an accuracy of 78.5%, which is somewhat higher than logistic regression.

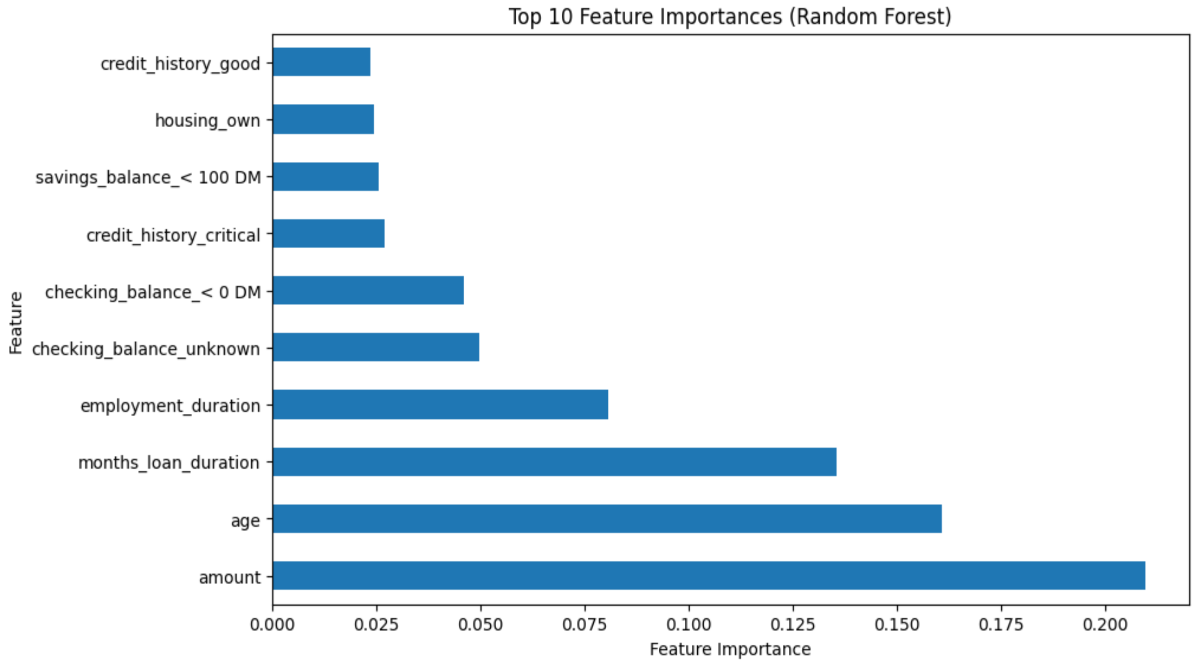
Precision-Recall Tradeoff:
The precision for the positive class is 0.71, while the recall is 0.46. Although the precision is satisfactory, the recall may be enhanced.

Confusion Matrix:
The confusion matrix reveals that while the model accurately predicts a substantial amount of the negative class (141 out of 141), it does not perform as well in the positive class (27 out of 59).

let's observe some plotting to provide insights into feature importances and the model's performance Thereby making informed decisions about threshold selection.







Observations

1) A smooth ROC curve with minimal rigidity that extends far above the line passing through the origin is typically indicative of high model performance. It signifies that the model may attain high True Positive Rates (Sensitivity) while retaining low False Positive Rates (1 - Specificity) across a range of threshold values.

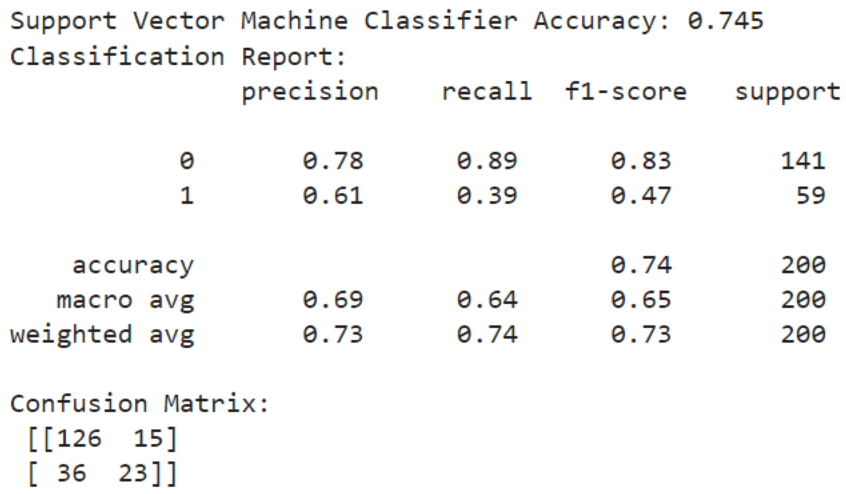
2) However, the ROC curve goes too much above the diagonal line, it might indicate overfitting. overfitting should be checked by evaluating the model on the test set and comparing the ROC curve of the train and test sets.

3) In precision-recall curve, at recall 0.6, the precision is 0.5, and at recall 1, the precision drops to 0.3. This indicates that as the recall increases, the precision of the model drops. It means that the model is making more false positive predictions as it tries to capture more positive cases. This can also be confirmed from FP value of 11 from confusion matrix.

4) the features "amount", "age", "months loan duration", "employment duration", and "checking balance unknown" have higher feature importance values. This suggests that these features are more important in predicting the target variable compared to other features.

C) Support Vector machines

Below are the results of support vector machines



```
Support Vector Machine Classifier Accuracy: 0.745
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.89      0.83       141
           1       0.61      0.39      0.47        59

    accuracy                           0.74       200
   macro avg       0.69      0.64      0.65       200
weighted avg       0.73      0.74      0.73       200

Confusion Matrix:
 [[126  15]
 [ 36  23]]
```

Accuracy:
The model's accuracy is 74.5%, indicating that it correctly predicts the class label for 74.5% of the samples.

Precision:
For class 0 (majority class), precision is 78%, meaning that when the model predicts an instance as class 0, it is correct 78% of the time. For class 1 (minority class), precision is 61%, indicating that when the model predicts an instance as class 1, it is correct 61% of the time.

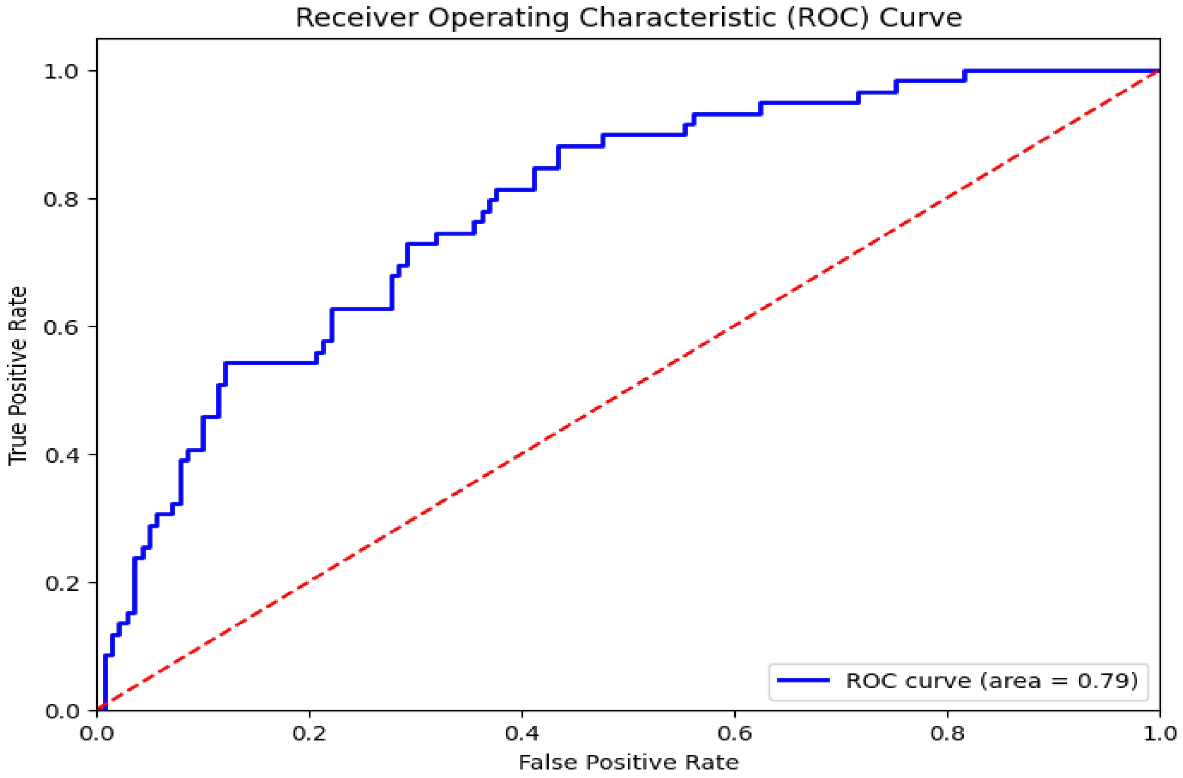
Recall:
For class 0, the recall is 89%, implying that the model correctly identifies 89% of the actual class 0 instances. For class 1, the recall is 39%, meaning that the model only identifies 39% of the actual class 1 instances.

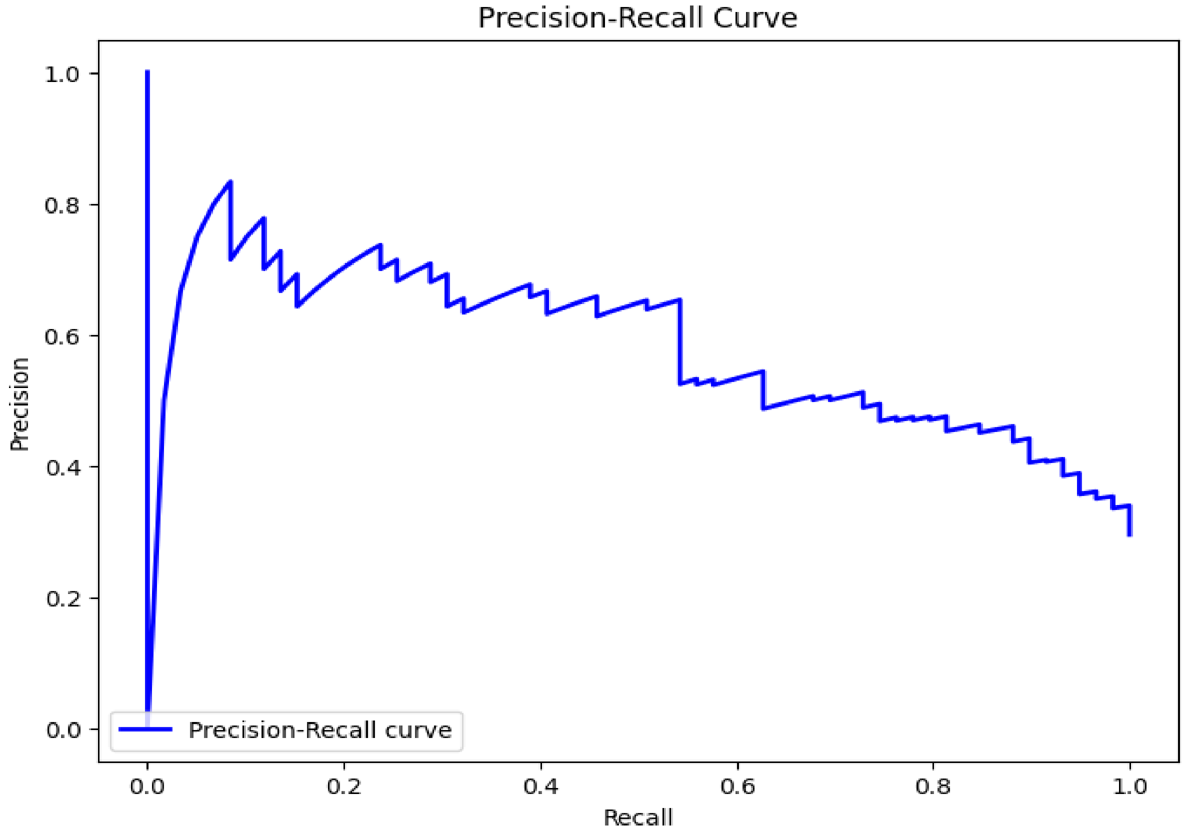
F1-score:
For class 0, the F1-score is 83%, which is the harmonic mean of precision and recall for class 0. For class 1, the F1-score is 47%, which is the harmonic mean of precision and recall for class 1.

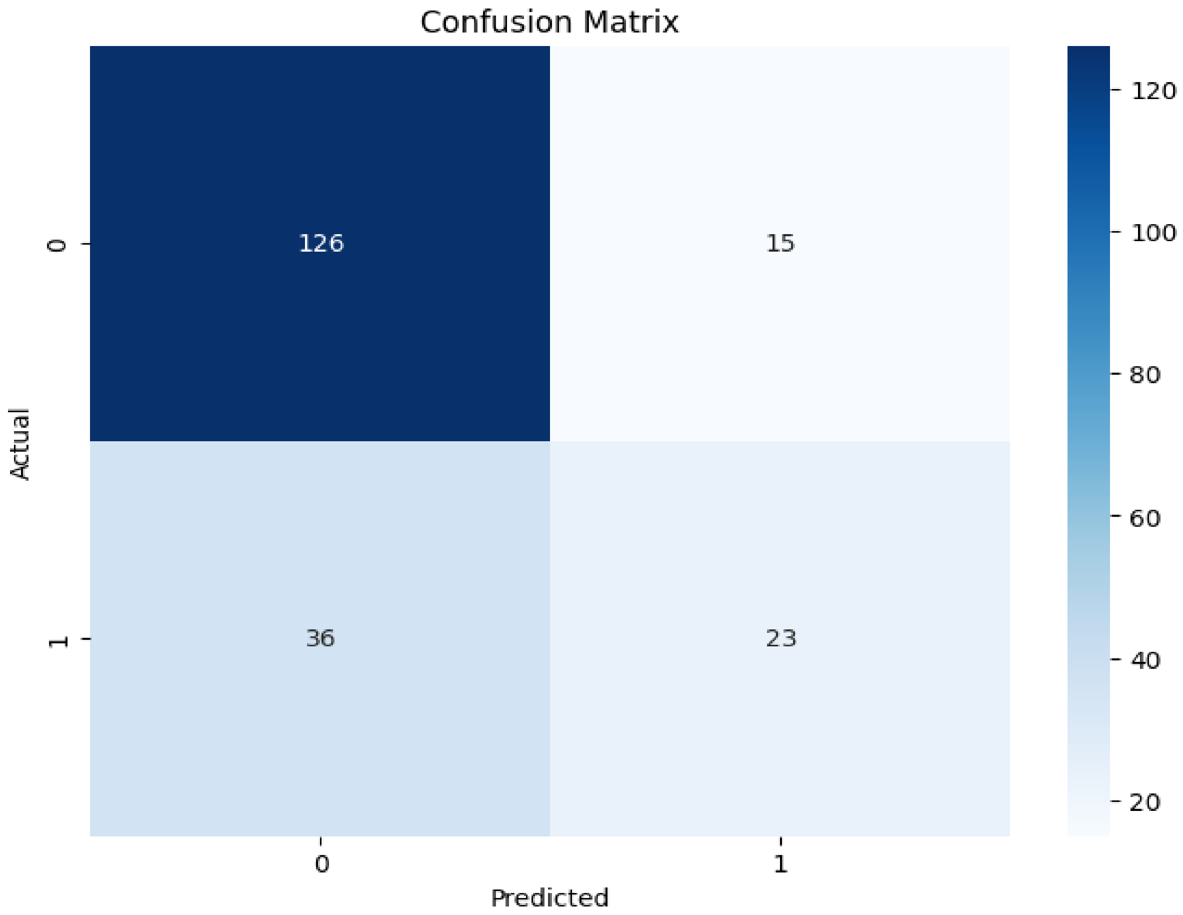
Confusion Matrix:
The model predicted class 0 more accurately than class 1. There are more instances of class 1 being misclassified as class 0 (false negatives) than instances of class 0 being misclassified as class 1 (false positives).

Overall, while the model performs reasonably well in predicting non-default instances, it struggles with predicting default instances, as indicated by the lower precision, recall, and F1-score for class 1.

Let's do the visual analysis for the same


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.79)


Precision-Recall Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Precision-Recall curve



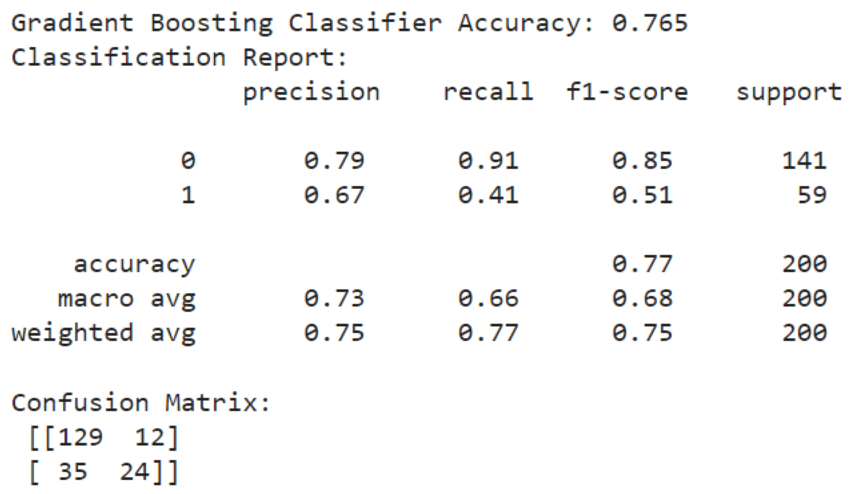
Observations

1) A rigid ROC curve that extends far above the line passing through the origin indicates that the model strikes a good balance between true positive rate (sensitivity) and false positive rate (1-specificity). In other words, the model is effective in distinguishing between positive and negative classes.

2) A Precision-Recall curve dropping from 0.8 to 0.4 as the recall value increases along the x-axis indicates that the model has high precision for lower recall values but suffers a significant drop in precision as recall increases. This behavior suggests that the model may have difficulty correctly identifying all positive instances, leading to a lower overall precision.

D) Gradient Boosting Classifier

Below are the results of Gradient Boosting Classifier model which includes accuracy, precision-recall and confusion matrix



```
Gradient Boosting Classifier Accuracy: 0.765
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.91      0.85       141
           1       0.67      0.41      0.51        59

    accuracy                           0.77       200
   macro avg       0.73      0.66      0.68       200
weighted avg       0.75      0.77      0.75       200

Confusion Matrix:
 [[129  12]
 [ 35  24]]
```

Accuracy:
The accuracy of the model is 0.765, indicating that it correctly predicts the target variable around 76.5% of the time.

Precision and Recall:
Precision for class 0 (non-default) is 0.79, and for class 1 (default) is 0.67. This means that when the model predicts a non-default, it is correct 79% of the time, and when it predicts a default, it is correct 67% of the time.

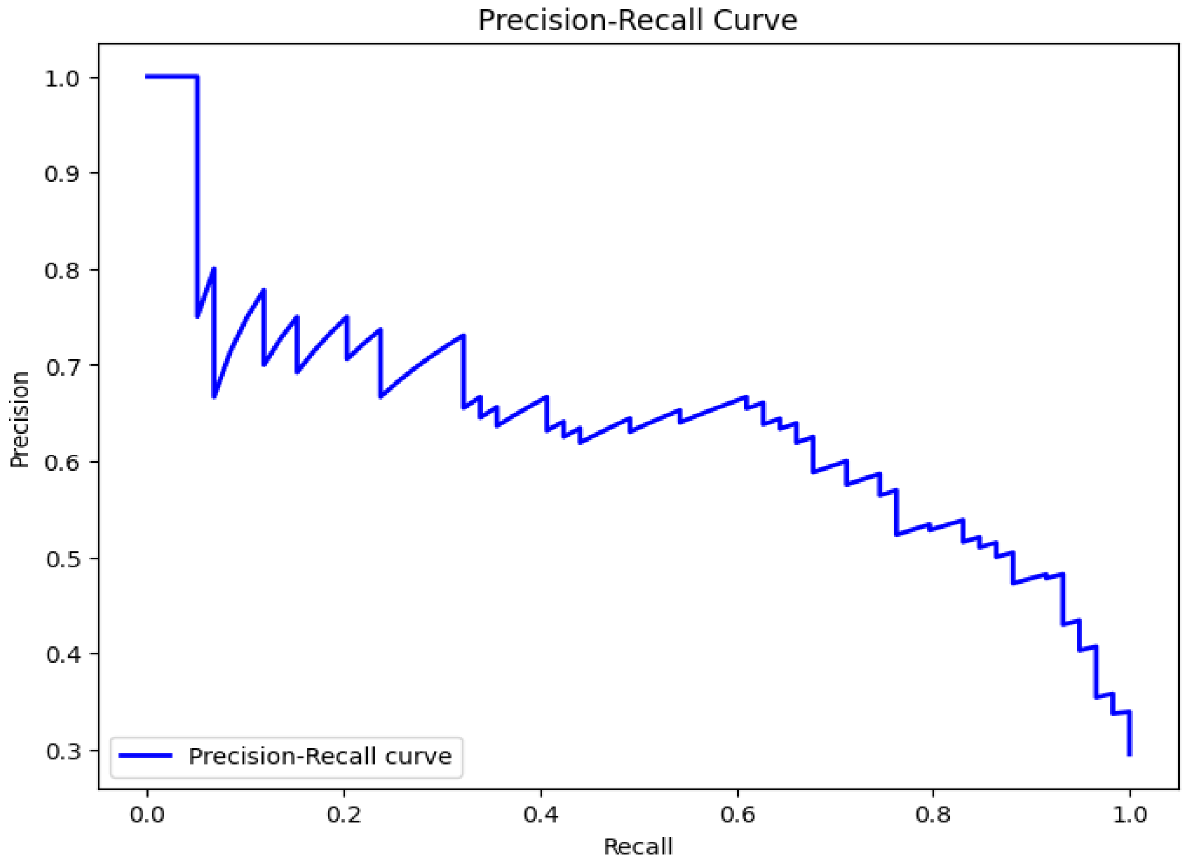
Recall for class 0 is 0.91, and for class 1 is 0.41. This indicates that the model correctly identifies 91% of the non-default cases but only 41% of the default cases.

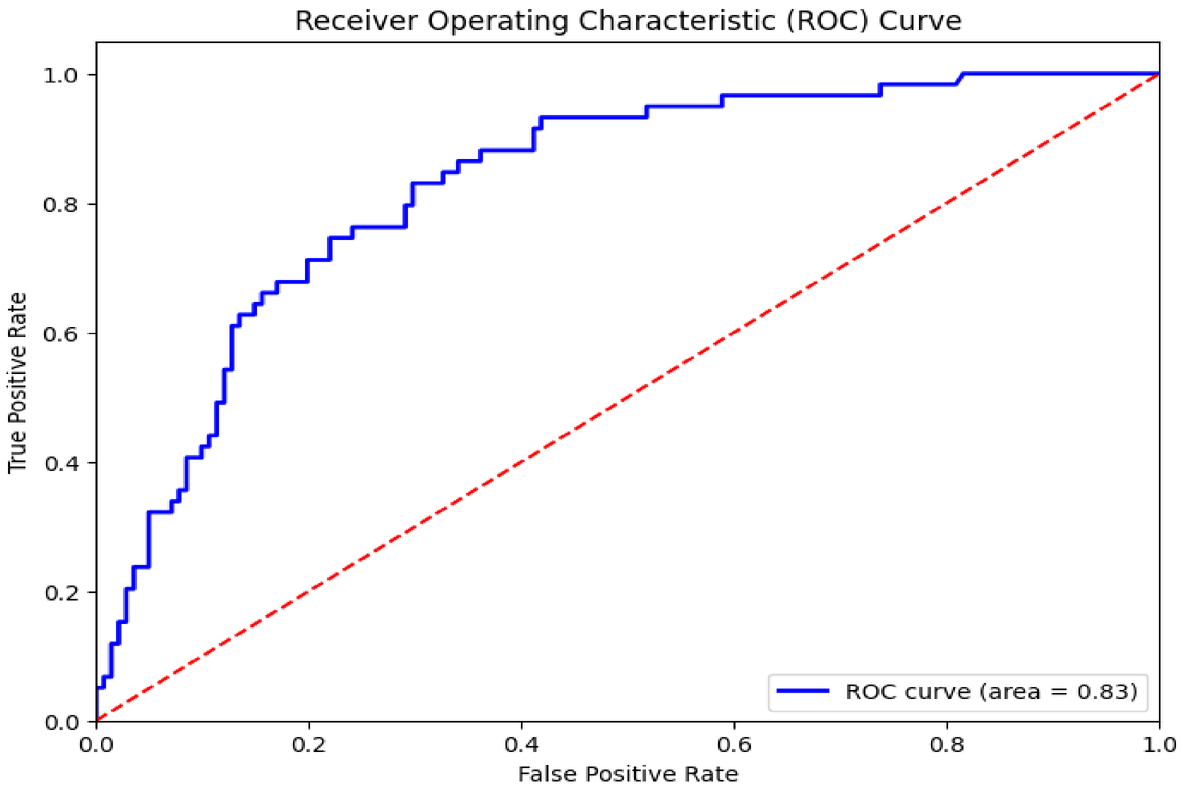
F1-score:
The F1-score for class 0 is 0.85, and for class 1 is 0.51. The F1-score is the harmonic mean of precision and recall. It is a measure of a test's accuracy and is used to balance precision and recall.

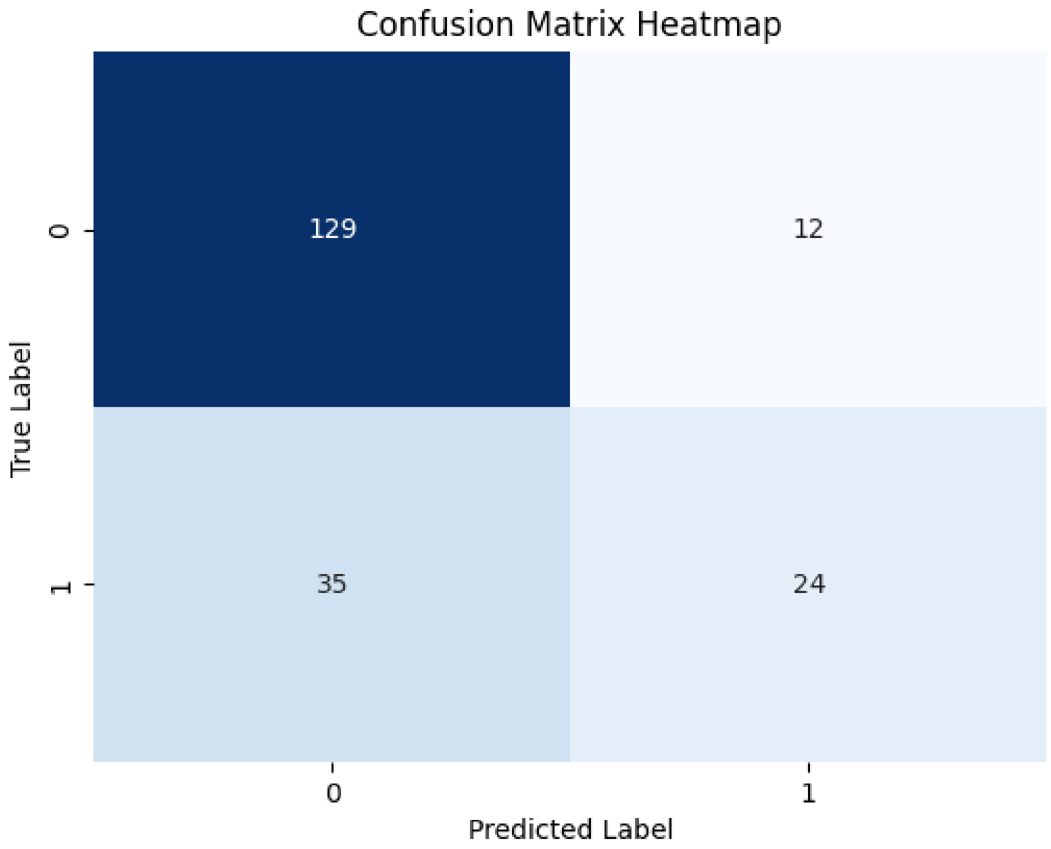
The model shows decent performance, with good accuracy, precision, and recall for the majority class (non-default). However, precision and recall for the minority class (default) are relatively lower.

Confusion matrix:
In the confusion matrix, we see that out of 141 non-defaulters, 129 were correctly classified, while out of 59 defaulters, only 24 were correctly classified.

Let’s observe the visualization of this model


Precision-Recall Curve
Precision
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.0
0.2
0.4
0.6
0.8
1.0
Precision-Recall curve


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.83)



Observations

1) Starting at a high precision value of 1, as recall increases, precision drops gradually but consistently, indicating a trade-off between the two metrics. This decline in precision suggests that as the model identifies more positive cases (higher recall), it becomes less accurate in classifying those cases correctly (lower precision). Overall, this behavior reflects the model's ability to balance between identifying all relevant instances (recall) and ensuring their accuracy (precision).

2) The ROC curve exhibits stiffness as it crosses considerably above the reference line that runs through the origin, showing a good separation between the positive and negative classes. This behaviour indicates that the model has a high true positive rate while maintaining a low false positive rate at various threshold levels. In other words, the model has great discriminatory power and effectively distinguishes between the two classes.

The Gradient Boosting Classifier showed good performance with an accuracy of 76.5%. It demonstrated high precision and recall for the negative class.

Results

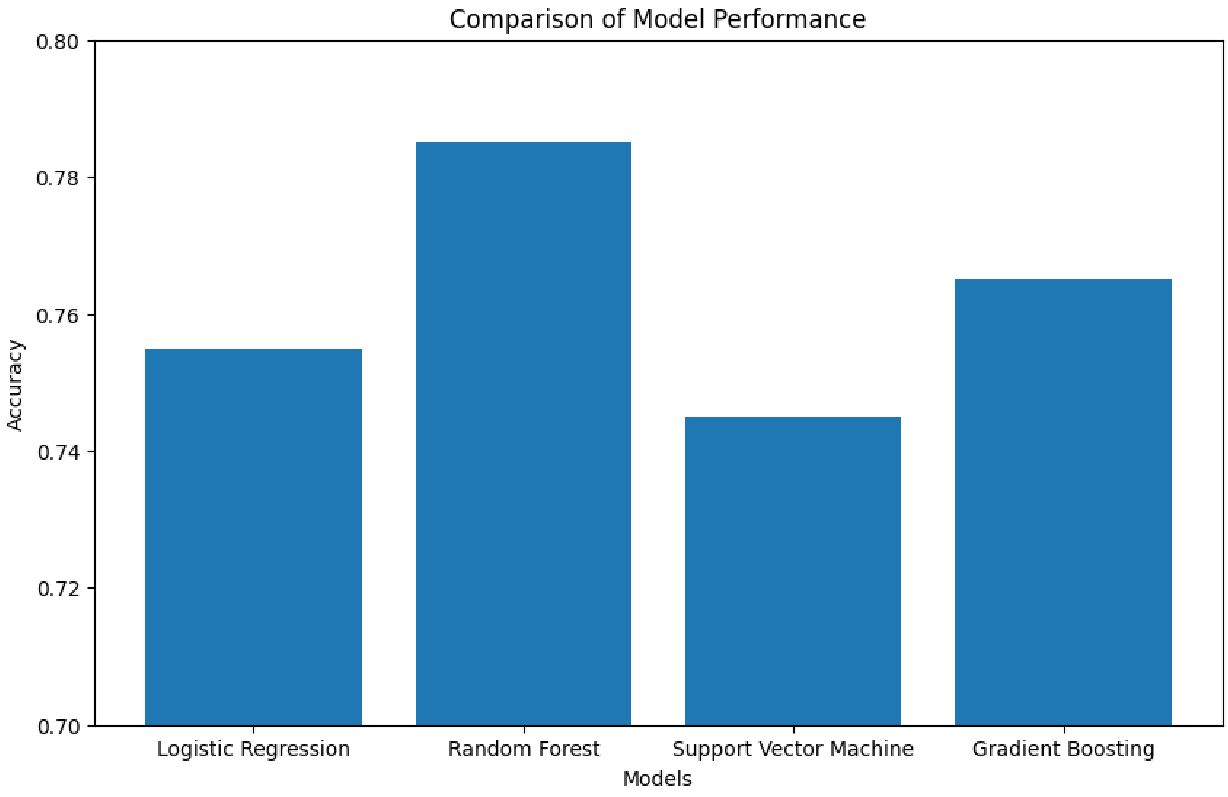
Our analysis revealed several key findings. Firstly, the loan default rate in the dataset was found to be approximately 30%. Secondly, through EDA, we identified that features such as loan amount, applicant's age, duration of the loan, and employment duration were significant predictors of loan default.

Among the models we trained, the Random Forest Classifier achieved the highest accuracy of 78.5%. It outperformed other models such as Logistic Regression, Support Vector Machine, and Gradient Boosting Classifier, which had accuracies of 75.5%, 74.5%, and 76.5% respectively.

Discussion

Comparison of models

The final model is chosen based on a number of factors, including the problem's specific requirements, computational resources, model interpretability, and the trade-off between model complexity and performance. The SVM model performed reasonably well in identifying non-default instances (precision of 78%), its accuracy in predicting default cases was lower (61%). The Random Forest Classifier produced promising results, with an accuracy of 78.5%.



Although Random Forest Classifier obtained somewhat higher accuracy than the Gradient Boosting Classifier, it exhibited lower precision and recall for default cases. Despite its reputation for durability and capacity to handle noisy data, the Random Forest Classifier did not outperform the Gradient Boosting Classifier in our scenario. Furthermore, the Gradient Boosting Classifier improves interpretability, providing for an increased understanding of the factors that contribute to default predictions. As a result, Gradient Boosting Classifier has been selected as the final model.

Reference:

Friedman, Jerome H. "Greedy function approximation: A gradient boosting machine." Annals of statistics (2001): 1189-1232.

https://dl.acm.org/doi/10.1145/3436369.3437405

One limitation of our study is the lack of information on external factors such as economic conditions, which could also influence loan default. Future studies could incorporate additional features to improve the predictive accuracy of the model.

Conclusion

The Random Forest Classifier showed the highest accuracy, while the Gradient Boosting Classifier demonstrated balanced precision and recall for both classes. In conclusion, our study demonstrates the effectiveness of machine learning models in predicting loan defaults. By leveraging features such as loan amount, applicant's age, and employment duration etc, we were able to develop a reliable predictive model. According to our findings, the Gradient Boosting Classifier is the best effective model for predicting loan defaults. The model's accuracy of 0.765 suggests that it can precisely predict if a customer will default on a loan. This approach could be an effective tool for financial organizations to manage loan defaults and mitigate associated risks.