German Bank Loan
Project Report

FINAL PROJECT
INTRODUCTION TO MACHINE LEARNING

May 6, 2024

Clinton KJ
Department of Information Science
University of Arizona

1 Introduction

The Banking sector continuously facing challenges in figuring out which customers are at risk of defaulting on their loans. This poses significant financial risks for banks . In this project, we aim to use Machine learning techniques and concepts to build a predictive model that can accurately classify customers based on the given data.

The German bank dataset provided includes borrower details such as account balance, loan duration, credit history, loan purpose, savings amount, employment tenure, income percentage, age and more. Additionally, the dataset features a target variable "default" that indicates whether each customer has defaulted on their loan.

Throughout our project journey, we will analyze the dataset specifics preprocess the data to address any anomalies or missing values and evaluate the performance of machine learning models.As part of this project, we aim to explore several interesting questions related to loan default prediction using machine learning techniques some of them are given below:

- How do various demographic factors such as age, existing loan count, and loan duration affect the default rate?
- Can we identify any patterns or trends in the purposes for which loans are taken, and how do these purposes correlate with loan default?
- Which performance metrics should we give more weightage while tuning the model?
- How does the performance of various machine learning algorithms compare in terms of predicting loan default?

2 Methods and Materials

2.1 Data Preprocessing

Data cleaning involves handling missing values and duplicates. In our project, we found no missing values but did encounter some typos, such as "car" being typed as "car0," as shown below in Figure 1:

```
checking_balance  :  ['< 0 DM' '1 - 200 DM' 'unknown' '> 200 DM']
credit_history  :  ['critical' 'good' 'poor' 'perfect' 'very good']
purpose  :  ['furniture/appliances' 'education' 'car' 'business' 'renovations' 'car0']
savings_balance  :  ['unknown' '< 100 DM' '500 - 1000 DM' '> 1000 DM' '100 - 500 DM']
employment_duration  :  ['> 7 years' '1 - 4 years' '4 - 7 years' 'unemployed' '< 1 year']
other_credit  :  ['none' 'bank' 'store']
housing  :  ['own' 'other' 'rent']
job  :  ['skilled' 'unskilled' 'management' 'unemployed']
phone  :  ['yes' 'no']
default  :  ['no' 'yes']
```

Figure 1: Removing Typos.

2.2 Exploratory Data Analysis (EDA)

2.2.1 Data Exploration

Data exploration involves presenting descriptive statistics to understand the distribution and characteristics of the data. In our project, we produced two tables showing statistics for both categorical and numerical features. The tables are shown below:

	count	unique	top	freq
checking_balance	1000	4	unknown	394
credit_history	1000	5	good	530
purpose	1000	5	furniture/appliances	473
savings_balance	1000	5	< 100 DM	603
employment_duration	1000	5	1 - 4 years	339
other_credit	1000	3	none	814
housing	1000	3	own	713
job	1000	4	skilled	630
phone	1000	2	no	596
default	1000	2	no	700

Figure 2: Categorical data descriptive statistics

For numerical features, it includes counts, means, standard deviations, minimums, 25th percentiles, medians, 75th percentiles, and maximums. For categorical features, it includes counts, unique values, top values, and frequencies. This analysis aims to give a comprehensive overview of the dataset's characteristics and distributions shown in Figure 3. Some notable insights from the descriptive statistics are given below

- On average people usually allocate, about 2.97% of their earnings for repaying loans with a standard deviation of around 1.12%.The range of percentages typically falls, between 1% and 4%.
- The most common category for checking balance is "unknown," with 394 occurrences Figure 2.
- "Furniture/appliances" is the most common purpose for taking a loan, with 473 occurrences.
- "Skilled" jobs are the most common among customers, with 630 occurrences.

	count	mean	std	min	25%	50%	75%	max
months_loan_duration	1000.0	20.903	12.058814	4.0	12.0	18.0	24.00	72.0
amount	1000.0	3271.258	2822.736876	250.0	1365.5	2319.5	3972.25	18424.0
percent_of_income	1000.0	2.973	1.118715	1.0	2.0	3.0	4.00	4.0
years_at_residence	1000.0	2.845	1.103718	1.0	2.0	3.0	4.00	4.0
age	1000.0	35.546	11.375469	19.0	27.0	33.0	42.00	75.0
existing_loans_count	1000.0	1.407	0.577654	1.0	1.0	1.0	2.00	4.0
dependents	1000.0	1.155	0.362086	1.0	1.0	1.0	1.00	2.0

Figure 3: Numerical data descriptive statistics

2.2.2 Data Visualization

This part of EDA includes creating visual representations of the data using plots, charts, and graphs to gain insights and identify patterns. We have used histogram, boxplots and countplots. Using these plots we were able to find insights about data. Furthermore we have analyzed correlation using heatmap to explore how different variables relate to each other.
From Figure 4,it is noticed that by the increase in the loan duration may increase loan

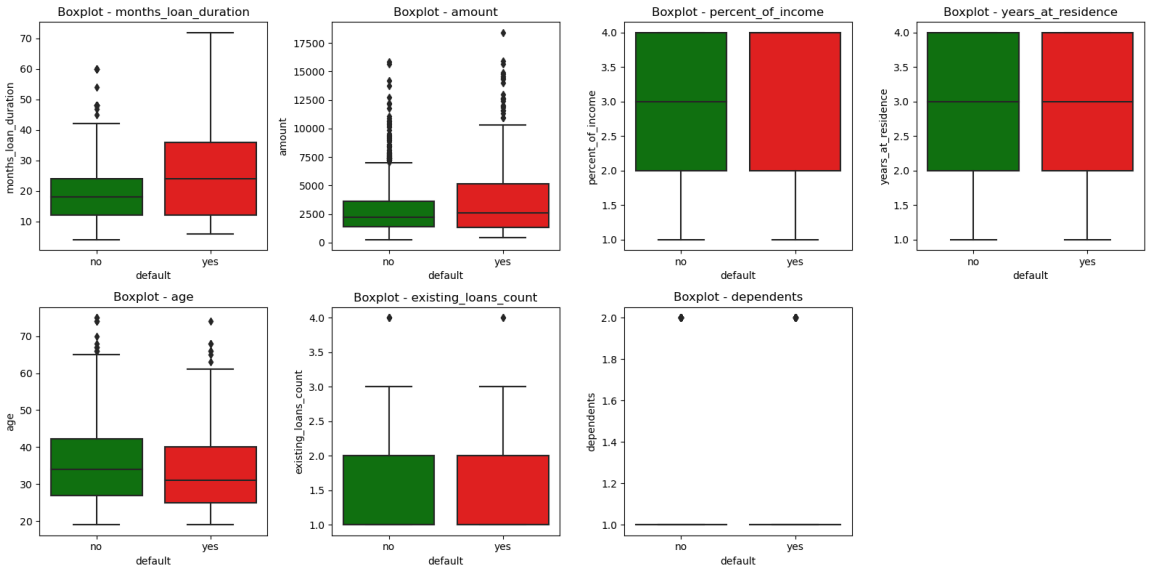


Figure 4: Numerical features Boxplot

default rates. This trend likely stems from higher interest burdens over extended loan periods, straining borrower's financial capacity. Consequently, longer loan terms may heighten default risks.

Figure 5 depicts the histogram plot for numerical data it is evident from the histogram that there exist positive skewness for the month loan duration,amount and age columns.

During the analysis of Figure 6, two intriguing patterns emerged. Firstly, in the plot illustrating checking balance, a significant trend becomes apparent: as checking balance decreases, the number of loan defaulters increases. This observation suggests a potential association between lower checking balances and higher default rates, indicating the importance of financial stability in loan repayment.

Secondly, examining the plot for credit history, it became evident that defaulters are more prevalent among individuals with "perfect" and "very good" credit histories. This finding raises the possibility that the higher default rates in these categories may stem from limited data availability, leading to a skewed representation of credit history in the dataset.

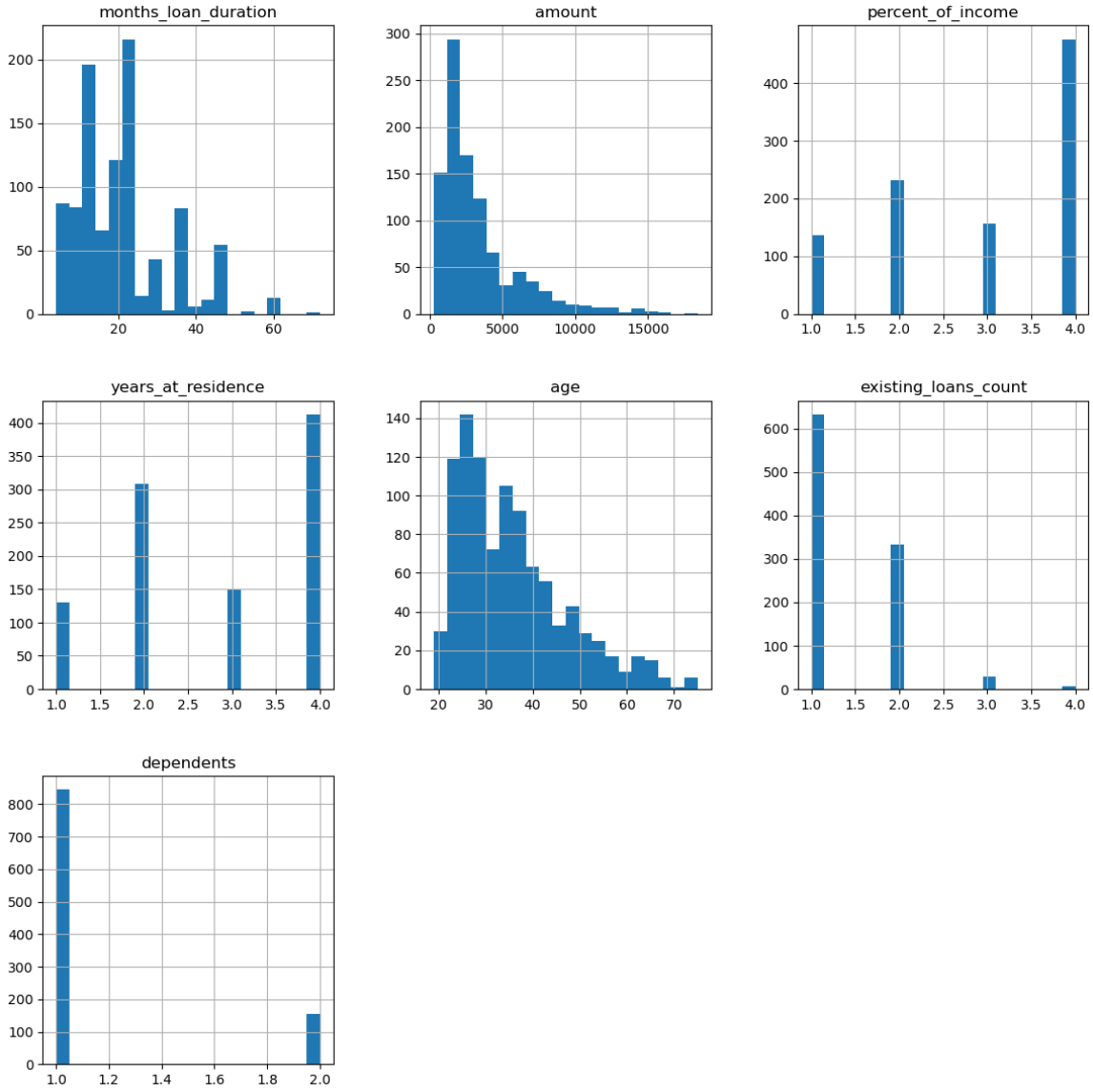


Figure 5: Numerical features histogram

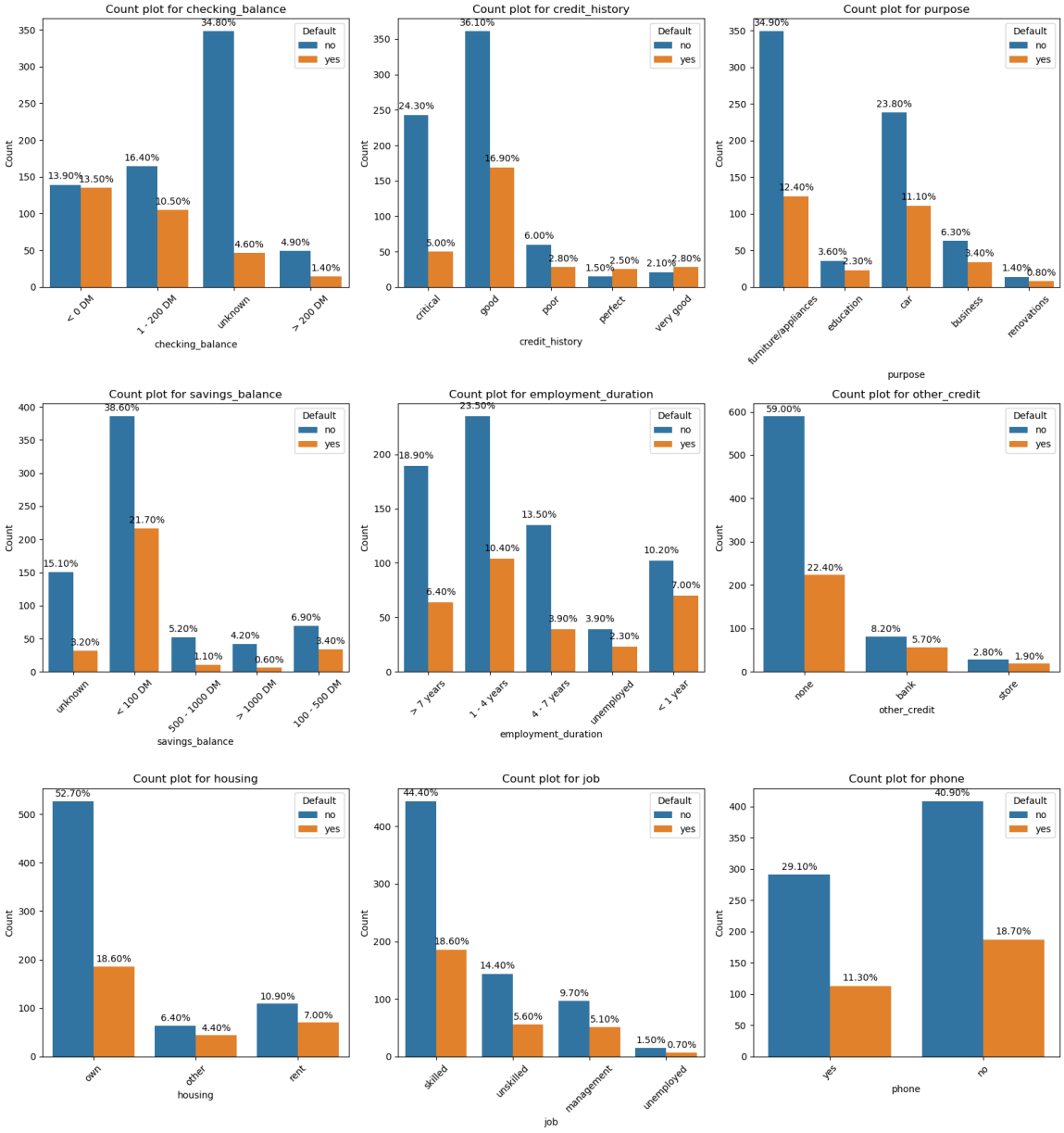


Figure 6: Categorical count plot with percentage

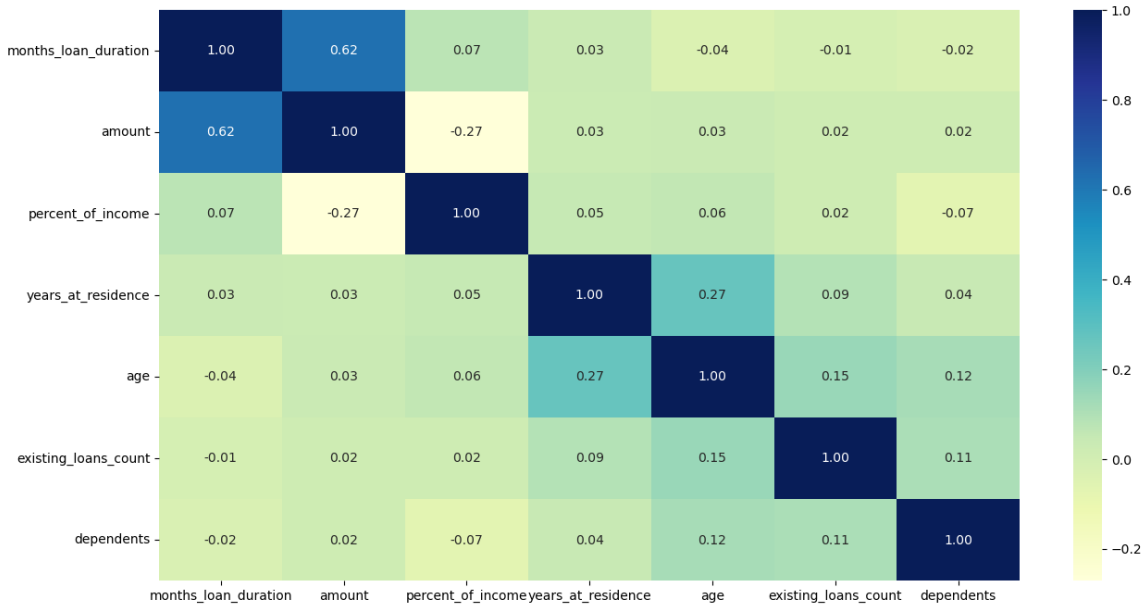
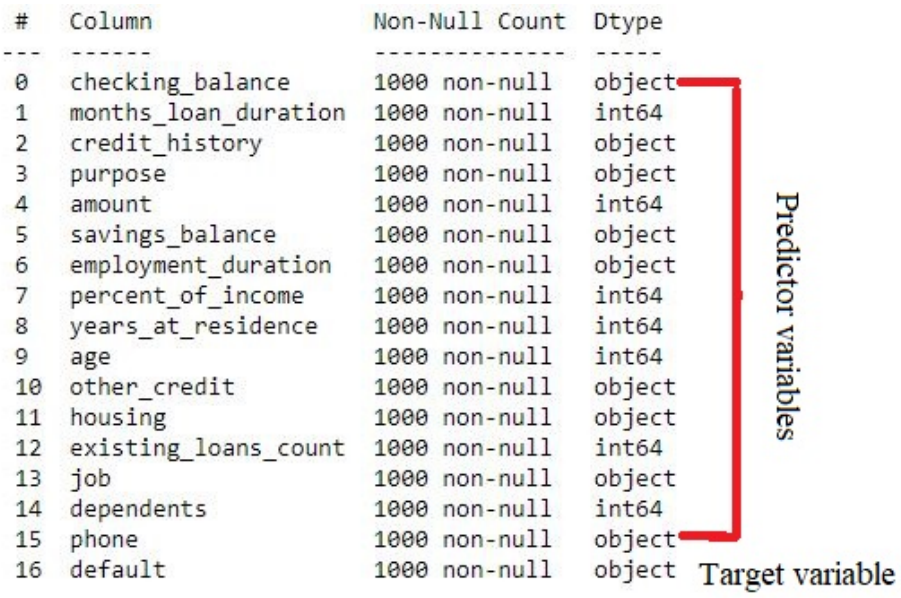


Figure 7: Correlation matrix

While analyzing the correlation matrix Figure 7, a positive correlation was observed between the amount of the loan and the duration of the loan. This finding suggests a possible trend where individuals opting for higher loan amounts also tend to choose longer loan tenures. However, aside from this relationship, no other significant correlations were found among the numerical features.

2.3 Data Transformation

In this step we need to get the data ready so that machine learning models can understand it easily. Initially we select the target variable, which, in our situation is the "default" column and we have large set of predictor variable which is shown in Figure 8.After that we split the dataset into two parts for training and testing purposes. 80% for training and 20% for testing. Next, we transform the features to ensure they are suitable for model input. The dataset comprises both ordinal and nominal features, with 2 ordinal columns and 7 nominal columns identified. To facilitate transformation, we utilize the ColumnTransformer function from the sklearn.compose library. This function allows us to specify the transformation methods for ordinal and nominal features separately, ensuring appropriate handling of each feature type.



#	Column	Non-Null Count	Dtype
0	checking_balance	1000 non-null	object
1	months_loan_duration	1000 non-null	int64
2	credit_history	1000 non-null	object
3	purpose	1000 non-null	object
4	amount	1000 non-null	int64
5	savings_balance	1000 non-null	object
6	employment_duration	1000 non-null	object
7	percent_of_income	1000 non-null	int64
8	years_at_residence	1000 non-null	int64
9	age	1000 non-null	int64
10	other_credit	1000 non-null	object
11	housing	1000 non-null	object
12	existing_loans_count	1000 non-null	int64
13	job	1000 non-null	object
14	dependents	1000 non-null	int64
15	phone	1000 non-null	object
16	default	1000 non-null	object

Figure 8: Dataset variables considered

2.4 Machine learning Models

In this project we utilized seven primary models, for examination. We chose these models because they were well suited for the task and proved to be successful, in identifying the patterns within the data.

After assessing how each of these models performed on its own we utilized an ensemble learning technique known as a voting classifier that involves combining the predictions made by the seven models to reach a conclusion. By tapping into the combined knowledge of models the voting classifier strives to enhance accuracy and recall leading to a more dependable classification result.

The models we have used listed below :

- XGB Classifier: Utilizes gradient boosting framework for efficient tree boosting
- Gradient Boosting Classifier: Boosts weak learners sequentially to improve overall model performance.
- Logistic Regression: Predicts binary outcomes using a linear function.

- Linear Discriminant Analysis: Classifies samples by maximizing class separation through linear combinations of features.
- Quadratic Discriminant Analysis: Assumes quadratic decision boundaries for classification.
- SGD Classifierr: Employs stochastic gradient descent for training linear classifiers.
- Decision Tree Classifier Constructs tree structures to partition feature space for classification.
- Voting Classifier (Final Model) Combines predictions from multiple models for improved accuracy and robustness.

2.4.1 Model evaluation criterion

Model Evaluation criteria throws light on how the above mention models need to be analysed based on their performance metrics.

When evaluating how well a model performs it's crucial to think about the impact of making prediction mistakes. There are two situations to consider; when predicting a default when none occurs, leading to customer dissatisfaction, and failing to predict a default when one occurs, resulting in financial losses for the bank.

When we look closely from the bank side, the second scenario is more important. If a model doesn't predict a default the bank ends up giving out a loan that might result in financial losses. This highlights why it's essential to prioritize recall while also maintaining reasonable levels of accuracy and precision.

3 Results

In this section we will explore the performance metrics of each and every model and in the end we compare all models using ROC - AUC, AUC-PR and the bar plots

3.1 Model performance before Hyper-parameter Tuning

In this section, we will evaluate the performance of the models before hyper-parameter tuning (Figure 9 and Figure 10).This step is crucial to identify and eliminate any models performing below average.

	Logistic Regression	Linear Discriminant Analysis	Quadratic Discriminant Analysis	SGD Classifier	Decision Tree Classifier	Gradient Boosting Classifier	XGB Classifier
prior_error_rate	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000
total_error_rate	0.243333	0.246667	0.303333	0.353333	0.316667	0.260000	0.276667
recall	0.477778	0.477778	0.566667	0.544444	0.433333	0.500000	0.477778
false_negative_rate	0.522222	0.522222	0.433333	0.455556	0.566667	0.500000	0.522222
false_positive_rate	0.123810	0.128571	0.247619	0.309524	0.209524	0.157143	0.171429
true_negative_rate (specificity)	0.876190	0.871429	0.752381	0.690476	0.790476	0.842857	0.828571
precision	0.623188	0.614286	0.495146	0.429825	0.469880	0.576923	0.544304
negative_predictive_value	0.796537	0.795652	0.802030	0.779570	0.764977	0.797297	0.787330
accuracy	0.756667	0.753333	0.696667	0.646667	0.683333	0.740000	0.723333

Figure 9: Descriptive comparison of models before Hyper-parameter Tuning (Test Dataset)

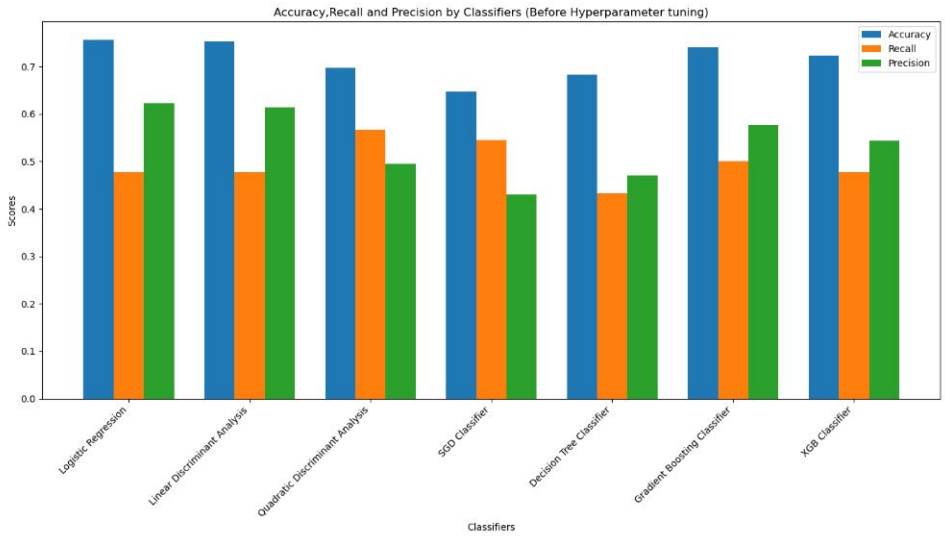
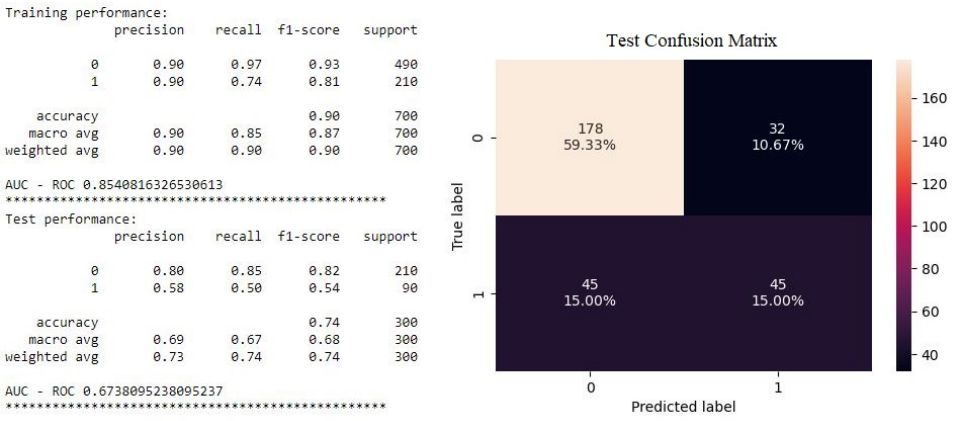


Figure 10: Graphical comparison of models before Hyperparameter Tuning (Test Dataset)

3.2 Individual performance of Models after Hyperparameter Tuning

Under this section we will be plotting the classification report and confusion metrics of 7 different primary models including votting classifier.

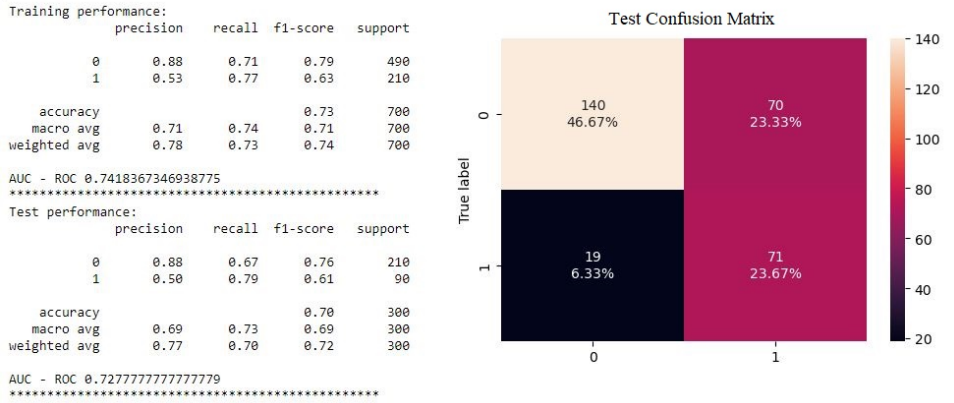
3.2.1 Gradient boosting Classifier



```
Training performance:
              precision    recall  f1-score   support

           0       0.90      0.97      0.93       490
           1       0.90      0.74      0.81       210

    accuracy                           0.90       700
   macro avg       0.90      0.85      0.87       700
weighted avg       0.90      0.90      0.90       700

AUC - ROC 0.8540816326530613
**************************************************
Test performance:
              precision    recall  f1-score   support

           0       0.80      0.85      0.82       210
           1       0.58      0.50      0.54        90

    accuracy                           0.74       300
   macro avg       0.69      0.67      0.68       300
weighted avg       0.73      0.74      0.74       300

AUC - ROC 0.6738095238095237
**************************************************
```


Figure 11: Classification report and Confusion matrix of Gradient boosting Classifier

3.2.2 Logistic Regression



```
Training performance:
              precision    recall  f1-score   support

           0       0.88      0.71      0.79       490
           1       0.53      0.77      0.63       210

    accuracy                           0.73       700
   macro avg       0.71      0.74      0.71       700
weighted avg       0.78      0.73      0.74       700

AUC - ROC 0.7418367346938775
**************************************************
Test performance:
              precision    recall  f1-score   support

           0       0.88      0.67      0.76       210
           1       0.50      0.79      0.61        90

    accuracy                           0.70       300
   macro avg       0.69      0.73      0.69       300
weighted avg       0.77      0.70      0.72       300

AUC - ROC 0.7277777777777779
**************************************************
```


Figure 12: Classification report and Confusion matrix of Logistic Regression

3.2.3 Linear Discriminant Analysis

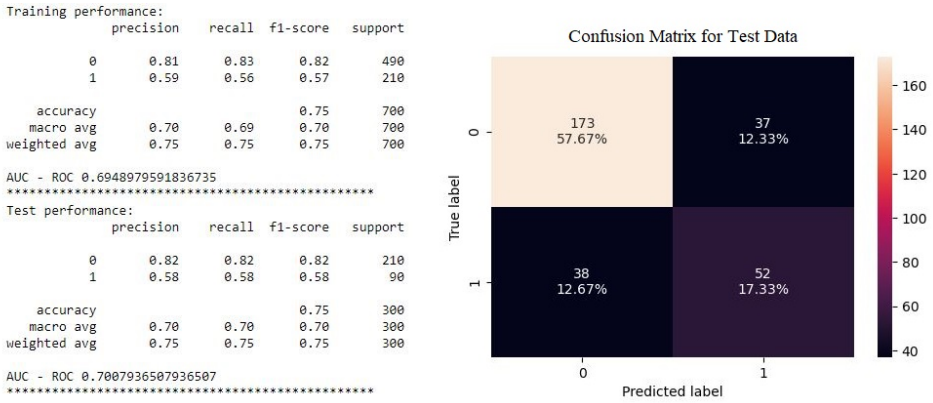


Figure 13: Classification report and Confusion matrix of Linear Discriminant Analysis

3.2.4 Quadratic Discriminant Analysis

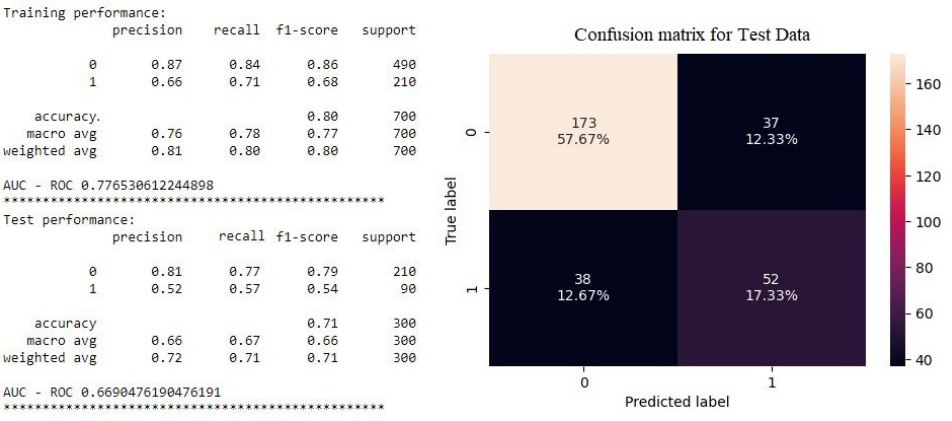


Figure 14: Classification report and Confusion matrix of Quadratic Discriminant Analysis

3.2.5 SGD Classifier

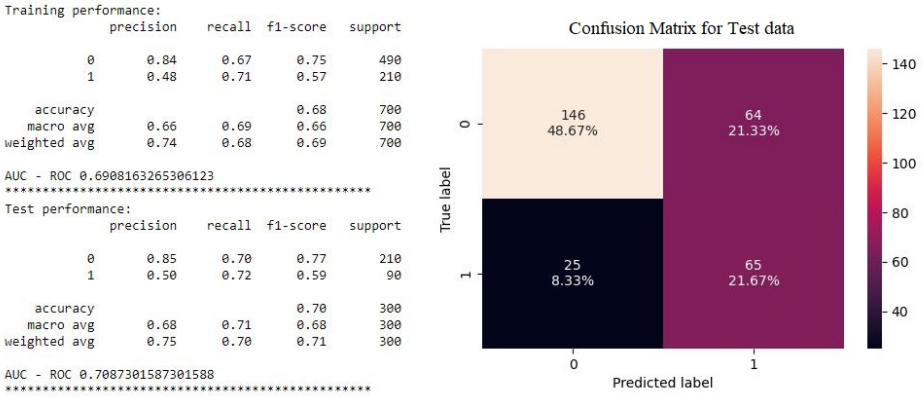


Figure 15: Classification report and Confusion matrix of SGD Classifier

3.2.6 Decision Tree Classifier

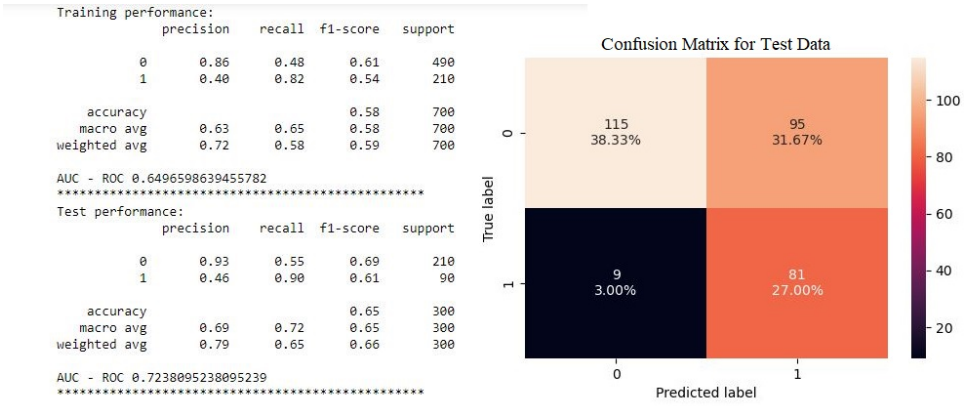


Figure 16: Classification report and Confusion matrix of Decision Tree Classifier

3.2.7 XGBoost Classifier

```
Training performance:
              precision    recall  f1-score   support

           0       0.86      0.48      0.61       490
           1       0.40      0.82      0.54       210

    accuracy                           0.58       700
   macro avg       0.63      0.65      0.58       700
weighted avg       0.72      0.58      0.59       700

AUC - ROC 0.6496598639455782
**************************************************
Test performance:
              precision    recall  f1-score   support

           0       0.93      0.55      0.69       210
           1       0.46      0.90      0.61        90

    accuracy                           0.65       300
   macro avg       0.69      0.72      0.65       300
weighted avg       0.79      0.65      0.66       300

AUC - ROC 0.7238095238095239
**************************************************
```

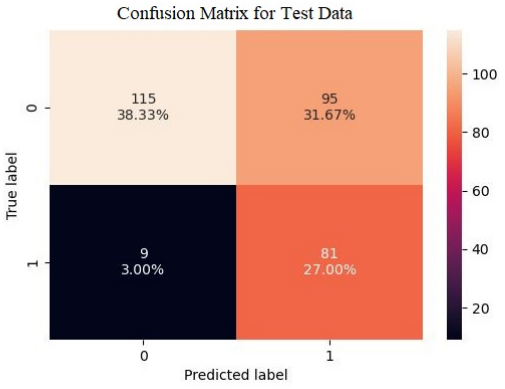


Figure 17: Classification report and Confusion matrix of XGBoost Classifier

3.3 Model performance Comparison and Final Model Results

In this section, we will evaluate the performance of the models after hyperparameter tuning . expectations, particularly in terms of recall.

	Gradient Boosting Classifier	Logistic Regression	Linear Discriminant Analysis	Quadratic Discriminant Analysis	SGD Classifier	Decision Tree Classifier	XGB Classifier
prior_error_rate	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000
total_error_rate	0.256667	0.296667	0.250000	0.290000	0.296667	0.346667	0.346667
recall	0.500000	0.788889	0.577778	0.566667	0.722222	0.900000	0.900000
false_negative_rate	0.500000	0.211111	0.422222	0.433333	0.277778	0.100000	0.100000
false_positive_rate	0.152381	0.333333	0.176190	0.228571	0.304762	0.452381	0.452381
true_negative_rate (specificity)	0.847619	0.666667	0.823810	0.771429	0.695238	0.547619	0.547619
precision	0.584416	0.503546	0.584270	0.515152	0.503876	0.460227	0.460227
negative_predictive_value	0.798206	0.880503	0.819905	0.805970	0.853801	0.927419	0.927419
accuracy	0.743333	0.703333	0.750000	0.710000	0.703333	0.653333	0.653333

Figure 18: Statistical descriptive comparison of models after Hyperparameter Tuning Except Voting classifier

3.3.1 ROC curve and PR Curve

Area under curve for ROC and PR curve of all models Except Voting classifier

	Model	AUC_PR
6	Decision Tree Classifier	0.695114
3	Linear Discriminant Analysis	0.645326
2	Logistic Regression	0.619596
1	Gradient Boosting Classifier	0.577192
5	SGD Classifier	0.574972
0	XGB Classifier	0.568823
4	Quadratic Discriminant Analysis	0.559405

	Model	AUC-ROC
3	Linear Discriminant Analysis	0.807037
2	Logistic Regression	0.802540
5	SGD Classifier	0.795899
1	Gradient Boosting Classifier	0.771376
0	XGB Classifier	0.770238
4	Quadratic Discriminant Analysis	0.767037
6	Decision Tree Classifier	0.723810

Figure 19: Area under ROC and PR curve of all models Except Voting classifier

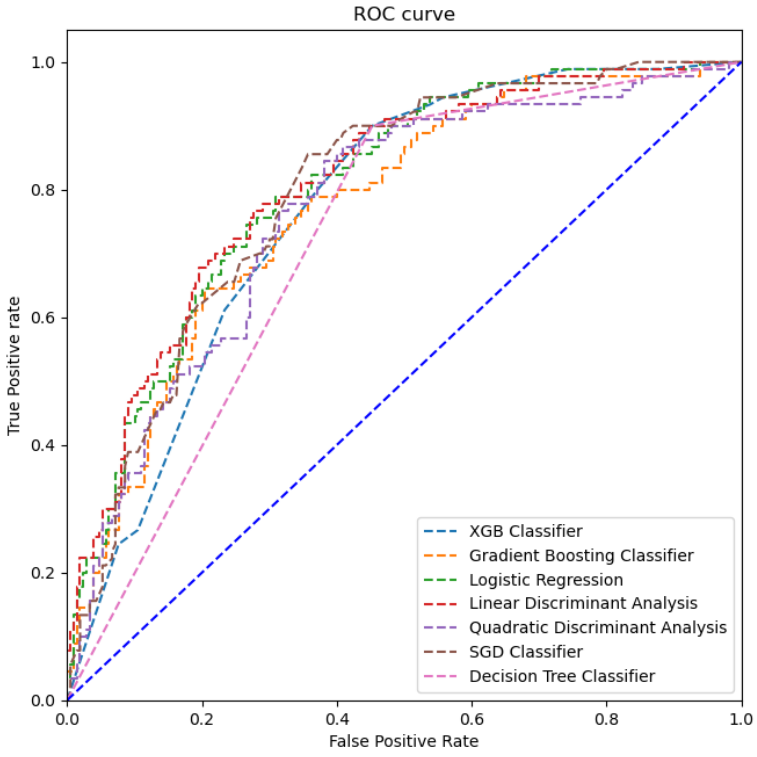


Figure 20: ROC curve of all models Except Voting classifier

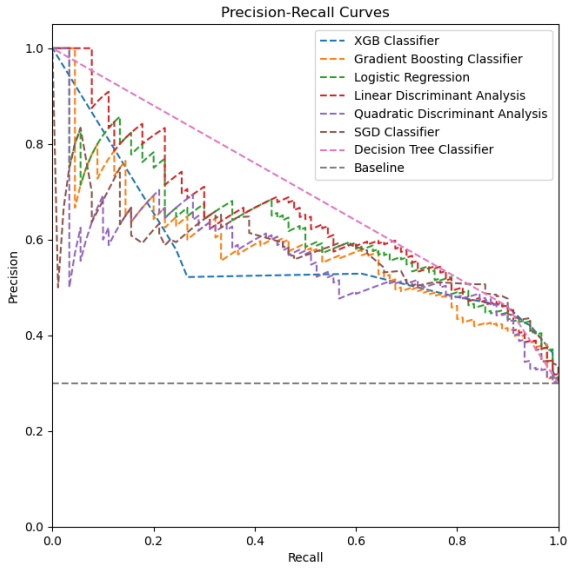


Figure 21: Precision Recall curve of all models Except Voting classifier

3.3.2 Voting Classifier Result (Final Model)

```
Training performance:
              precision    recall  f1-score   support

           0       0.84      0.74      0.79       490
           1       0.53      0.67      0.59       210

    accuracy                           0.72       700
   macro avg       0.68      0.71      0.69       700
weighted avg       0.75      0.72      0.73       700

AUC - ROC 0.7071428571428571
**************************************************
Test performance:
              precision    recall  f1-score   support

           0       0.88      0.75      0.81       210
           1       0.57      0.77      0.65        90

    accuracy                           0.76       300
   macro avg       0.73      0.76      0.73       300
weighted avg       0.79      0.76      0.76       300

AUC - ROC 0.7595238095238095
**************************************************
```

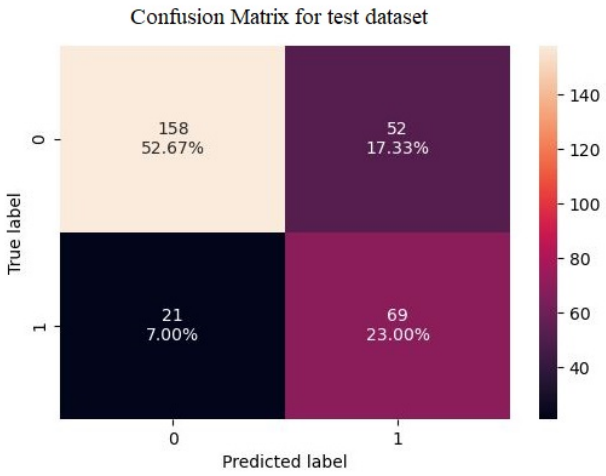


Figure 22: Classification report and Confusion matrix for Voting classifier

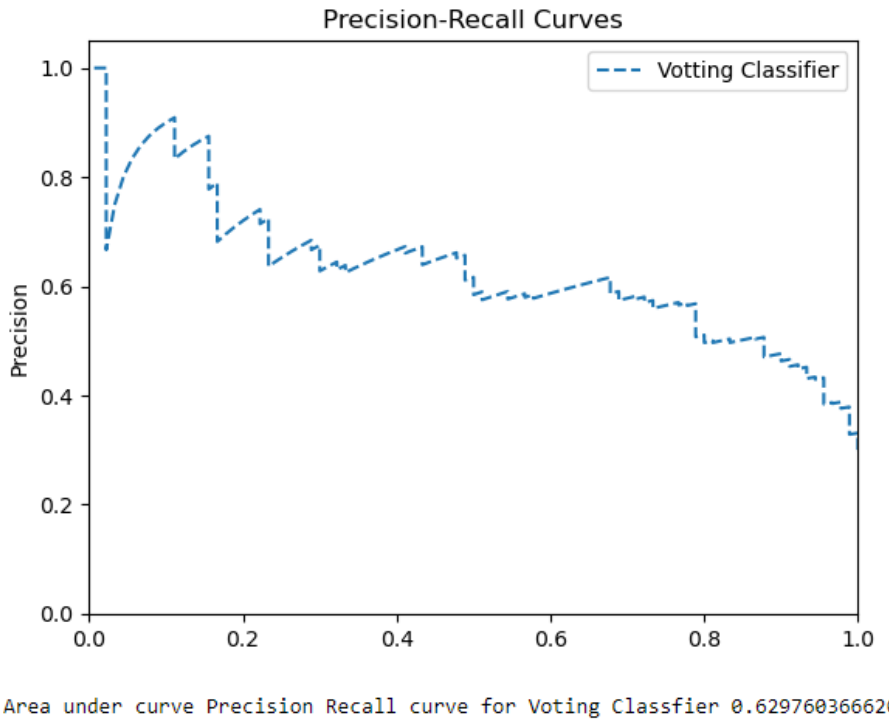


Figure 23: Auc-PR and PR curve for voting classifier

3.3.3 Final Result

	Gradient Boosting Classifier	Logistic Regression	Linear Discriminant Analysis	Quadratic Discriminant Analysis	SGD Classifier	Decision Tree Classifier	XGB Classifier	Voting Classifier
prior_error_rate	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000
total_error_rate	0.256667	0.296667	0.250000	0.290000	0.296667	0.346667	0.346667	0.243333
recall	0.500000	0.788889	0.577778	0.566667	0.722222	0.900000	0.900000	0.766667
false_negative_rate	0.500000	0.211111	0.422222	0.433333	0.277778	0.100000	0.100000	0.233333
false_positive_rate	0.152381	0.333333	0.176190	0.228571	0.304762	0.452381	0.452381	0.247619
true_negative_rate (specificity)	0.847619	0.666667	0.823810	0.771429	0.695238	0.547619	0.547619	0.752381
precision	0.584416	0.503546	0.584270	0.515152	0.503876	0.460227	0.460227	0.570248
negative_predictive_value	0.798206	0.880503	0.819905	0.805970	0.853801	0.927419	0.927419	0.882682
accuracy	0.743333	0.703333	0.750000	0.710000	0.703333	0.653333	0.653333	0.756667

Figure 24: Statistical descriptive comparison of models after Hyperparameter Tuning including Voting classifier

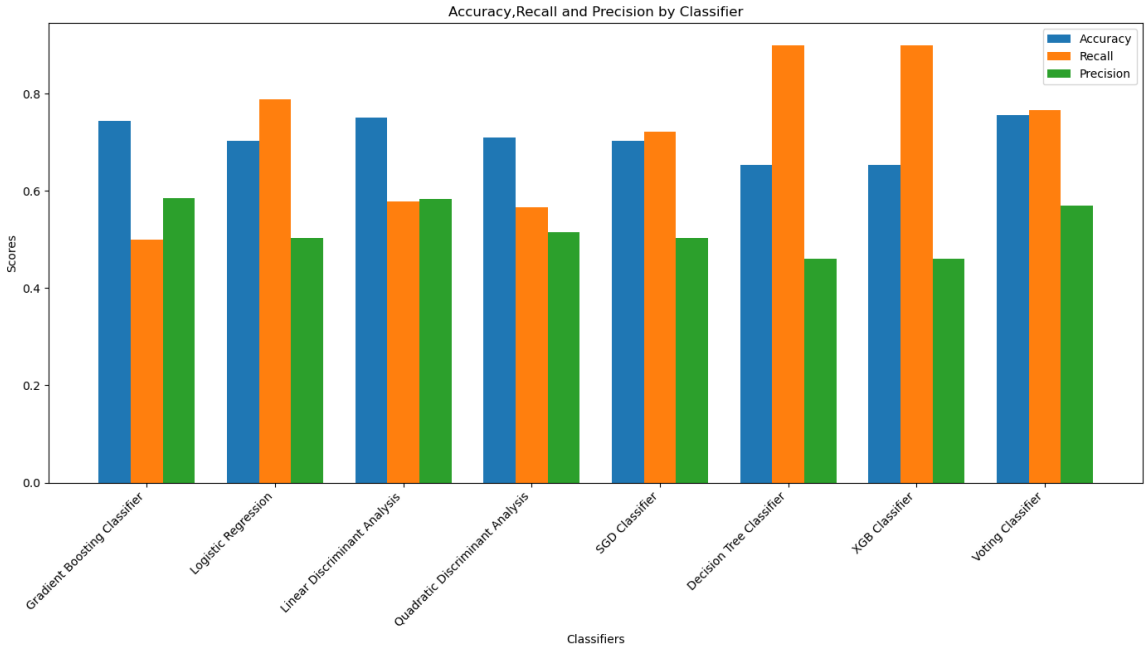


Figure 25: Graphical comparison of models after Hyperparameter Tuning including Voting classifier

4 Discussion

4.1 Model performance

4.1.1 Analysis of all models before Tuning

Table from Figure 9 shows the descriptive statistics on the all model performance before hyperparameter tuning. After analyzing the model performance metrics reveal varied results across recall, accuracy, and precision.QDA have high recall of 0.57 compared to other models, followed by Logistic Regression and Linear Discriminant Analysis. Gradient Boosting Classifier and Logistic Regression score highest with 0.74 and 0.76 accuracy, respectively. However, accuracy alone may not fully represent model performance, especially in imbalanced datasets. Logistic Regression and Gradient Boosting Classifier exhibit notable precision, indicating their ability to minimize false positives.

Given Figure 10 shows less performance, particularly in recall (almost all model values are below 0.6), hyperparameter tuning is essential to optimize model behavior and enhance predictive capabilities. This iterative process ensures improved performance across key metrics, making the models more reliable for our project need.

4.1.2 Analysis of all models after Tuning

Initially, we analyze Figure 18 to identify the models demonstrating superior performance in recall (2.4.1),accuracy and AUC-PR(In our situation the positive class is significantly underrepresented, AUC-PR provides a more reliable assessment of model performance. It reflects the precision-recall trade-offs inherent in imbalanced datasets, offering a clearer picture of how well the model distinguishes between the classes). Subsequently, we examine the confusion matrix of the top-performing models, while also assessing their AUC-ROC and AUC-PR scores. When we analyse Figure 18 it is evident that XGB and Decision tree classifier have high recall compared to any other models but the total error is high, precision is low and accuracy is 65% but when we look at the AUC-PR (0.69) Decision tree performs well while XGB value is about 0.56. Also we can see the number of false negative for both of this is 9 (only 3%) from confusion matrix depicted in figure 16 and figure 17 because of less accuracy and high total error we are unable to select both of this model.

Contrary to XGB and the decision tree, SGD performs well with an accuracy of 70%, but its AUC-PR is 0.57, which is comparatively lower, and it has a medium number of false negatives (FN = 25, see Figure 15). We cannot select this model because of its lower recall (0.72), which falls below 0.75, and its less AUC-PR value. However, the logistic regression model has a high recall of 0.79, good accuracy of 70%, and a good AUC-PR value. Nonetheless, its precision is very low at 0.5, and the total error rate is high. Compared to any other primary model, logistic regression performs well (see Figure 12). Since the precision is very low and the total error is on the higher side, we are unable to select this model. Gradient boost classifier, Quadratic Discriminant Analysis (QDA), and Linear Discriminant Analysis models are not performing well compared to other models.

4.1.3 Final Model Voting Classifier Analysis

From the above section, it is evident that while one model may excel in certain metrics, it falls short significantly below the desired or optimum values in others.In this situation we can use ensemble technique such as Voting Classifier. In this project we are using Soft voting which is the Prediction for the output class in soft voting is based on the average probability assigned to that class. In our project we are giving selective models to the voting classifier for optimal performance.The models are selected based on the top three models from AUC-PR, AUC-ROC (see see Figure 19) and highest recall models.The selected models are XGB(High recall),Decision tree classifier(High recall,Auc-PR High),LDA(AUC -PR),Logistic regression(AUC-PR), SGD(AUC-ROC).

Figure 22 shows the classification report of the voting classifier and the confusion matrix. When we analyze the report of the Voting classifier model, it shows great accuracy (76%) and recall (0.77), which are above 75%. By looking at the confusion matrix, we

observe very few false negatives, accounting for only 21, making it the lowest total error rate (0.24) we have ever encountered among all models (see Figure 24). It is important to note that the model achieved the highest F1 score without compromising the AUC-PR (0.63) (see Figure 23), and it achieved the second-highest precision of 0.57, which is outstanding. In Figure 25, the plot shows the comparison of the voting classifier with other models.

4.1.4 Limitation and potential future directions

Potential future directions entail delving deeper into hyperparameter tuning. While this endeavor demands substantial computational power and time investment, it holds the promise of enhancing the performance of boosting models. By systematically adjusting hyper-parameters, we can unlock the full potential of these models, ultimately optimizing their predictive capabilities.

5 Conclusion

In conclusion, this project addresses the critical challenge faced by the banking sector in predicting loan default risk. Through comprehensive data analysis and model evaluation, we have identified key factors influencing loan default and explored various machine learning algorithms' performance in this domain. Despite individual model limitations, the ensemble approach, particularly the Voting Classifier, emerges as a promising solution, showcasing high accuracy, recall, and precision while minimizing false negatives. The project underscores the importance of Exploratory data Analysis, model selection, and hyper-parameter tuning in improving predictive accuracy and recall. Moving forward, further research in hyper-parameter tuning and exploring advanced ensemble techniques could significantly enhance predictive capabilities, providing valuable insights for risk management in the banking sector.