An Operating Systems and Security Portfolio

By:

Abdelrahman Waheed

The Knowledge Hub
International University Institution of Egypt

Coventry University

Table of contents

Links: Onedrive

GitHub

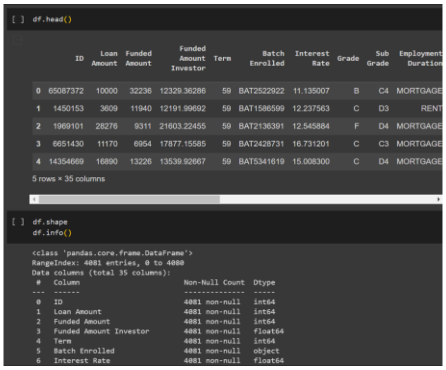
Introduction

Loan defaults pose a problem for banks and other financial institutions because they can result in losses and have an effect on overall economic health. Customers who don't pay back their loans cause banks to lose a lot of money every year. This has an impact on the economy, impeding both its stability and growth. This study focuses on using data-driven approaches to predict loan defaulters in order to address this issue. The study tries to create a model by examining various aspects of loan applicants, including the amount funded, location, remaining loan balance, credit history, and more. The 6 models(including Logistic Regression, SVM, RNN, KNN, and Decision Tree Classifier) will be created to assess each person's creditworthiness and predict their likelihood of loan default.
The model will be built using a loan dataset taken from https://www.kaggle.com/datasets/hemanthsai7/loandefault. There are 35 columns and 96376 entries in the dataset. Every row signifies a single loan entry, and the columns offer different pieces of information about the loan applicants and the specifics of their loans. There are both categorical (object) and numeric (int64 and float64) columns in the dataset.

Ther are few existing works that have also compared the performance of different machine learning models for loan status prediction. However there is no exiting work that compares and evaluates all 6 models together. Furthermore, the we utilizeed different techniques, including grid search and random search, to optimize hyperparameters and improve model performance. This demonstrates a more comprehensive approach compared to existing works that might use simpler parameter tuning methods.

Cleaning and Preprocessing

Upon loading the dataset, a preliminary exploration was conducted to understand its structure and composition. The dataset, consisting of 67,463 rows and 35 columns, presented a mix of numerical and categorical features..The df.head(), df.shape, and df.info() functions are used to display the first few rows of the dataset, its shape (number of rows and columns), and information about each column, including data types and missing values. After that we remove the missing values using the dropna function. After that we check for duplicates and remove them.



```
[ ] df.head()
```

	ID	Loan Amount	Funded Amount	Funded Amount Investor	Term	Batch Enrolled	Interest Rate	Grade	Sub Grade	Employment Duration
0	65087372	10000	32236	12329.36286	59	BAT2522922	11.135007	B	C4	MORTGAGE
1	1450153	3609	11940	12191.99692	59	BAT1586599	12.237563	C	D3	RENT
2	1969101	28276	9311	21603.22455	59	BAT2136391	12.545884	F	D4	MORTGAGE
3	6651430	11170	6954	17877.15585	59	BAT2428731	16.731201	C	C3	MORTGAGE
4	14354669	16890	13226	13539.92667	59	BAT5341619	15.008300	C	D4	MORTGAGE

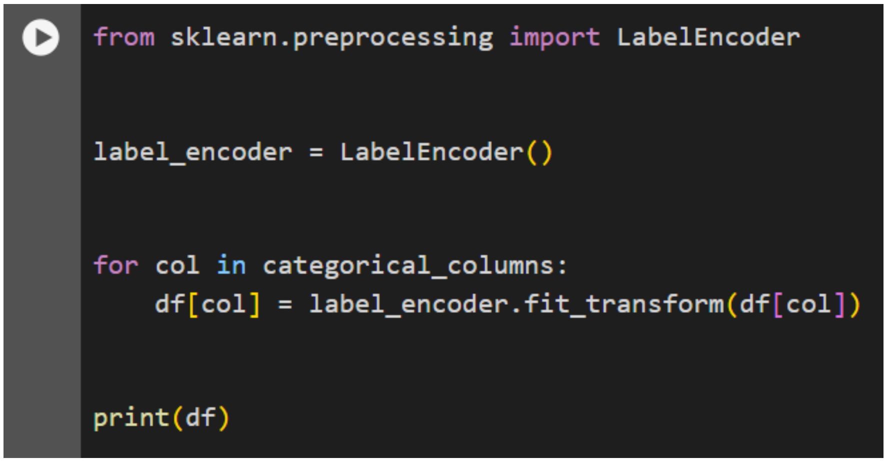
5 rows × 35 columns

```
[ ] df.shape
    df.info()
```

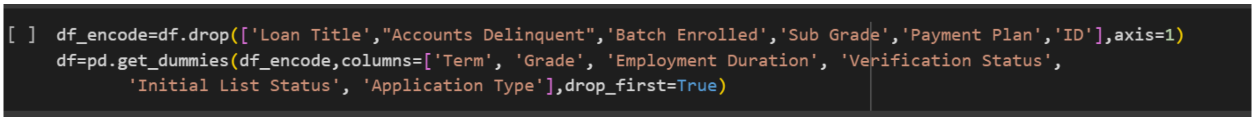
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4081 entries, 0 to 4080
Data columns (total 35 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   ID                            4081 non-null   int64
 1   Loan Amount                   4081 non-null   int64
 2   Funded Amount                 4081 non-null   int64
 3   Funded Amount Investor        4081 non-null   float64
 4   Term                          4081 non-null   int64
 5   Batch Enrolled                4081 non-null   object
 6   Interest Rate                 4081 non-null   float64
```

Encoding

The dataset has categorical data this means that data is represented in categories or groups. These data points fall into distinct, non-numeric categories, and they are often used to represent qualitative characteristics.to processes categorical data we have to preform encoding.in our case we will be using label encoding. Label encoding is a process of converting categorical data into numerical format by assigning a unique numerical label to each category or class. We us scikit-learn library's LabelEncoder to transform categorical columns into numeric labels. The original DataFrame is then updated with these numeric representations, effectively replacing the categorical values with their encoded counterparts. This label encoding is particularly useful when working with machine learning algorithms that require numeric input, as it ensures that the model can process and interpret categorical features. This encoding facilitates the integration of categorical information into machine learning models while adhering to their numeric input requirements. Then A few columns('Loan Title', "Accounts Delinquent", 'Batch Enrolled', 'Sub Grade', 'Payment Plan', 'ID') that are not as important for the prediction task are eliminated. One-hot encoding is used to encode categorical variables so that machine learning algorithms can use them efficiently in a numerical format. The data are now ready for additional analysis.



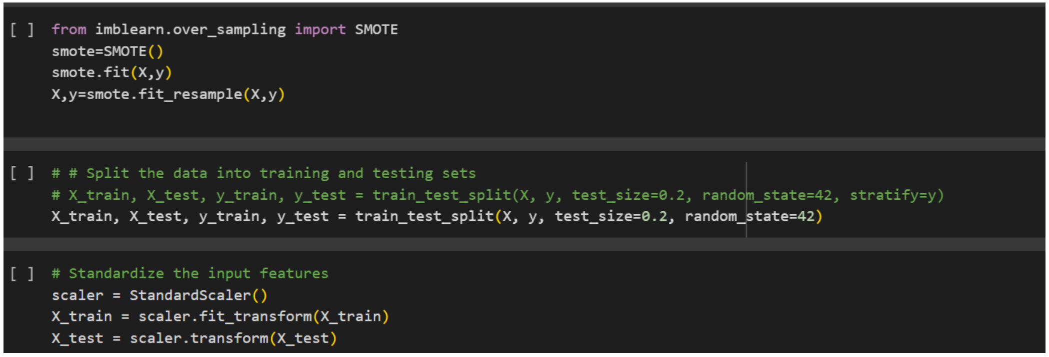
```
from sklearn.preprocessing import LabelEncoder


label_encoder = LabelEncoder()


for col in categorical_columns:
    df[col] = label_encoder.fit_transform(df[col])


print(df)
```



```
[ ]  df_encode=df.drop(['Loan Title',"Accounts Delinquent",'Batch Enrolled','Sub Grade','Payment Plan','ID'],axis=1)
     df=pd.get_dummies(df_encode,columns=['Term', 'Grade', 'Employment Duration', 'Verification Status',
            'Initial List Status', 'Application Type'],drop_first=True)
```

Handling imbalances

The dataset at first was imbalanced. Meaning that the distribution of classes in the target variable is not equal or roughly equal. Specifically, it means that one class (the minority class) has significantly fewer instances than the other class or classes (the majority class or classes). In our case the dataset was imbalanced because the approved loans is more prevalent than the Denied loans in the loan status column.to fix this we used oversampling. oversampling involves increasing the number of instances in the minority class to balance it with the majority class.The Synthetic Minority Over-sampling Technique (SMOTE) is applied to create synthetic instances of the minority class, balancing the class distribution. This balanced dataset is then divided into training and testing sets using the train_test_split function. The training set is used to train the machine learning models, while the testing set allows for the evaluation of model performance on unseen data. The split is stratified to maintain the proportion of different classes in both sets. We then standardized the features using StandardScaler. Standardization ensures that all features have a similar scale, preventing certain features with larger numerical values from dominating the learning process. This step is particularly important for algorithms that rely on distance metrics or gradient-based optimization.



```
[ ] from imblearn.over_sampling import SMOTE
    smote=SMOTE()
    smote.fit(X,y)
    X,y=smote.fit_resample(X,y)
```

```
[ ] # # Split the data into training and testing sets
    # X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

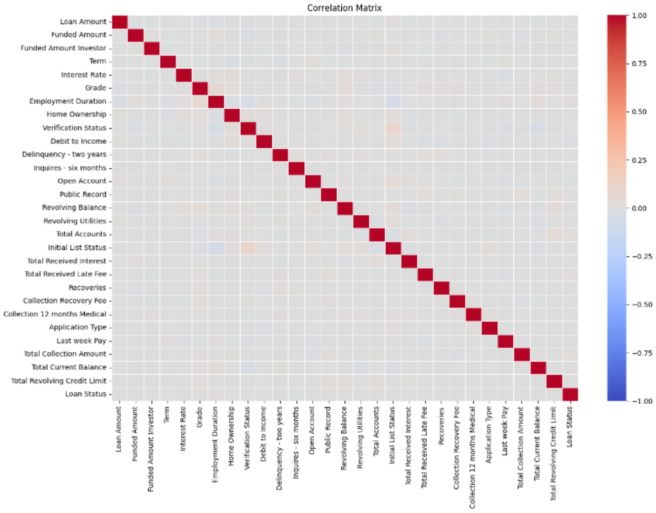
```
[ ] # Standardize the input features
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
```

Data Analysis

Descriptive statistics were computed, revealing insights into the central tendencies and dispersions of numerical features. The correlation matrix was visualized using a heatmap, providing an understanding of relationships between numeric variables. Multivariate analysis involved scatter plots, illustrating the relationship between loan amount and interest rate based on loan grades. Numerical columns were subjected to detailed statistical analyses, including histograms, count plots, and box plots, shedding light on the distribution and characteristics of key variables such as loan amount, interest rate, and loan grade.

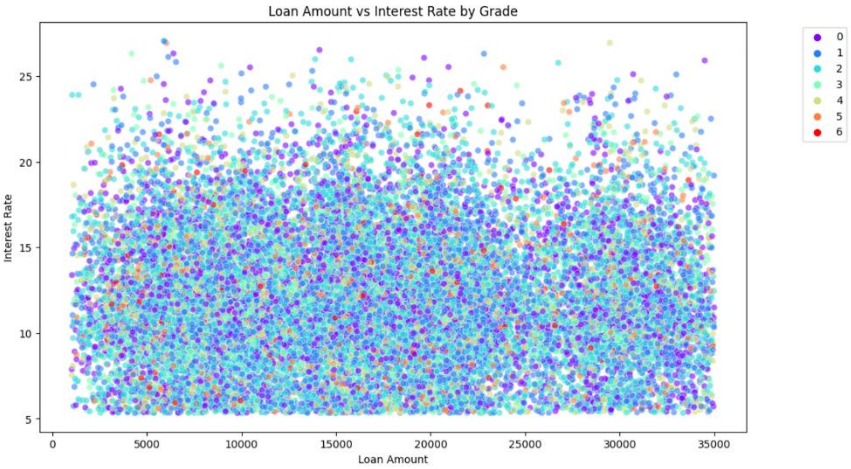
funding of loans correlation matrix

The correlation between various loan-related variables is displayed in this correlation matrix. A statistical indicator of the direction and strength of a relationship between two variables is correlation. It can have a value between -1 and 1, where a perfect negative correlation is represented by a value of 1, a perfect positive correlation by a value of 1, and no correlation by a value of 0.



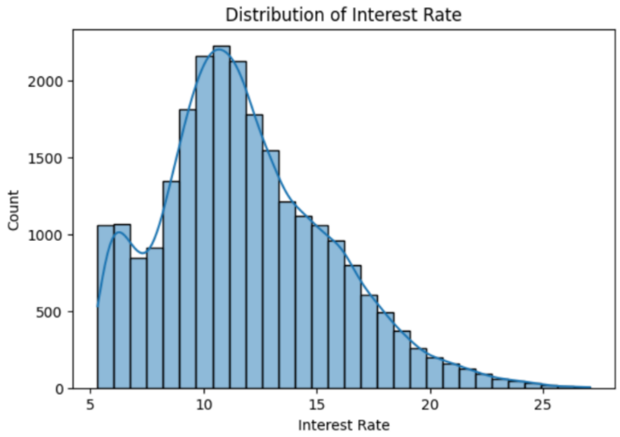
loan amount and interest rate correlation Scatterplot

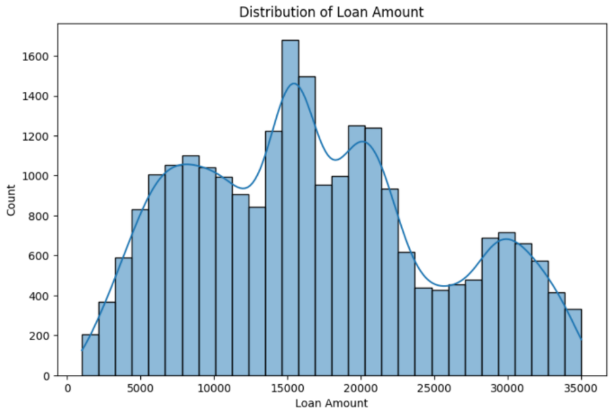
The relationship between loan amount and interest rate by grade is displayed in the chart. The loan amount is plotted on the x-axis, and the interest rate is plotted on the y-axis. Every dot on the diagram denotes a loan, and the color of the dot indicates the loan's grade. The graph indicates that the loan amount and interest rate have a positive relationship. This implies that the interest rate tends to rise along with the loan amount. This is due to the fact that larger loans usually have higher interest rates charged by lenders.



.

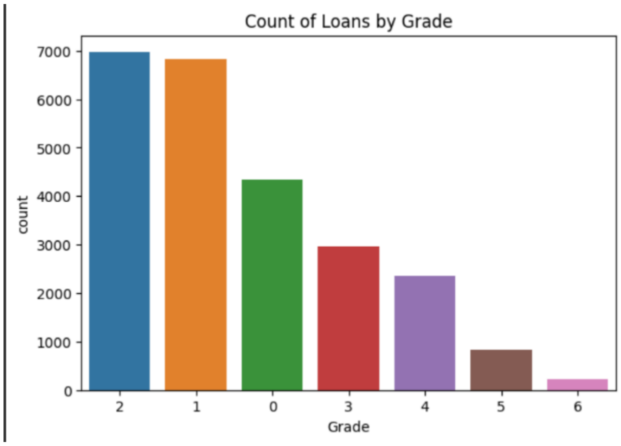
Histograms for loan and interest rates.



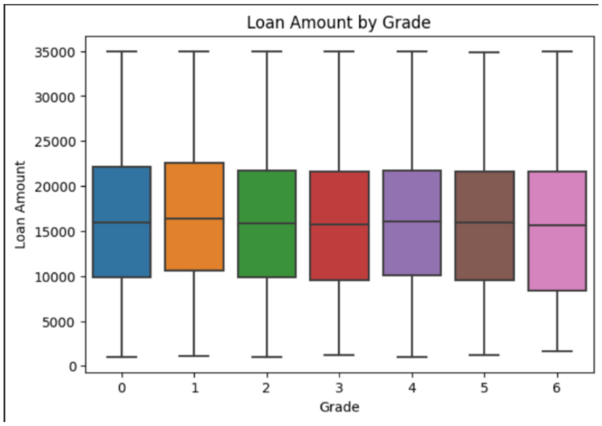


The charts provided depict the distribution of loan amounts by county and interest rates by loan grade in the United States. Both are histograms, illustrating the distribution of data. In the loan amount chart, the x-axis represents loan amounts, revealing a right-skewed distribution with a concentration of loans between $5,000 and $10,000, and a scarcity of loans above $30,000. Meanwhile, the interest rate chart displays a right-skewed distribution on the x-axis representing interest rates, with a prevalent range between 6% and 9%, and a scarcity of loans with interest rates surpassing 20%. These visualizations collectively suggest a pattern of more loans at the lower end of both the loan amount and interest rate spectrums.

Bar chart and box plot of loans by grade



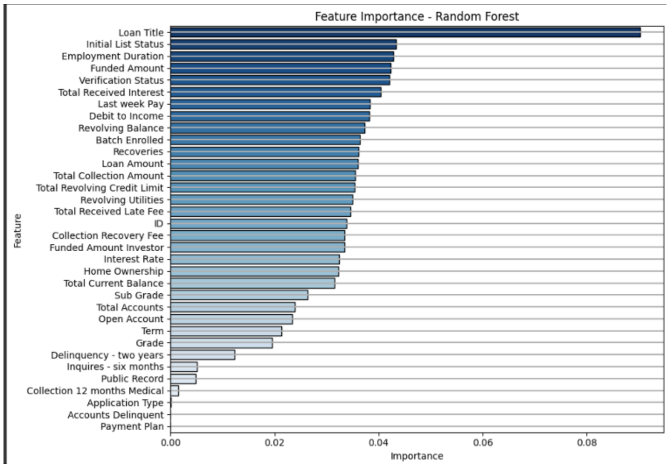
The bar chart illustrating the number of loans by grade reveals a trend where higher-graded loans, indicative of lower risk, are more prevalent, while there are fewer loans issued to borrowers with lower grades. Lenders prioritize borrowers with higher grades, reflecting a propensity to extend loans to those deemed less likely to default. Notably, the chart demonstrates a decreasing trend in the number of loans as the grade decreases. This decline is attributed to the fewer number of borrowers with lower grades, and lenders exercise greater selectivity in lending to this group. Concurrently, the box plot depicting loan amounts by grade provides additional insight into the relationship between loan grade and loan amount. The box plot highlights a discernible pattern: loans with lower grades tend to have smaller amounts. This correlation stems from lenders perceiving lower-grade loans as riskier, leading them to exercise caution and limit the loan amounts for this category.



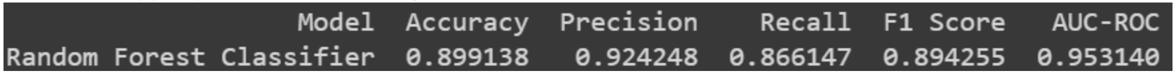
Creating and tuning models machine learning models

We will apply the following models: Decision Tree, K-Nearest Neighbors (KNN), Recurrent Neural Network (RNN), Support Vector Machine (SVM), Random Forest, and Logistic Regression. We will also perform hyperparameter tuning to find the best hyperparameters for optimal results .

1. Random Forest Classifier:
Random Forest is an ensemble learning technique that builds a multitude of decision trees during training and merges them together for more accurate and stable predictions. Each tree is constructed using a random subset of the features and the final prediction is made by voting (classification) or averaging (regression) the predictions of individual trees.
We also performed Feature Importance which measures the contribution of each input variable in making predictions with a trained model. The most important features in the model are employment duration, funding amount, debt to income ratio, and revolving balance. These features are all related to the borrower's ability to repay the loan. Employment duration indicates how long the borrower has been employed, which is a good measure of their job stability.

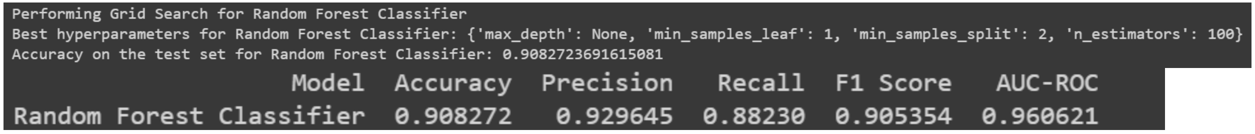


Random Forest is initially trained with default hyperparameters. Grid search and random search are then employed to find the optimal combination of hyperparameters, maximizing the model's performance. Here are some analysis of the models
Without hyperparameter tuning

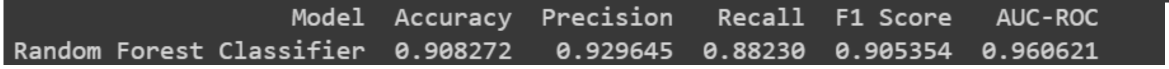


```
                    Model  Accuracy  Precision    Recall  F1 Score   AUC-ROC
Random Forest Classifier  0.899138   0.924248  0.866147  0.894255  0.953140
```

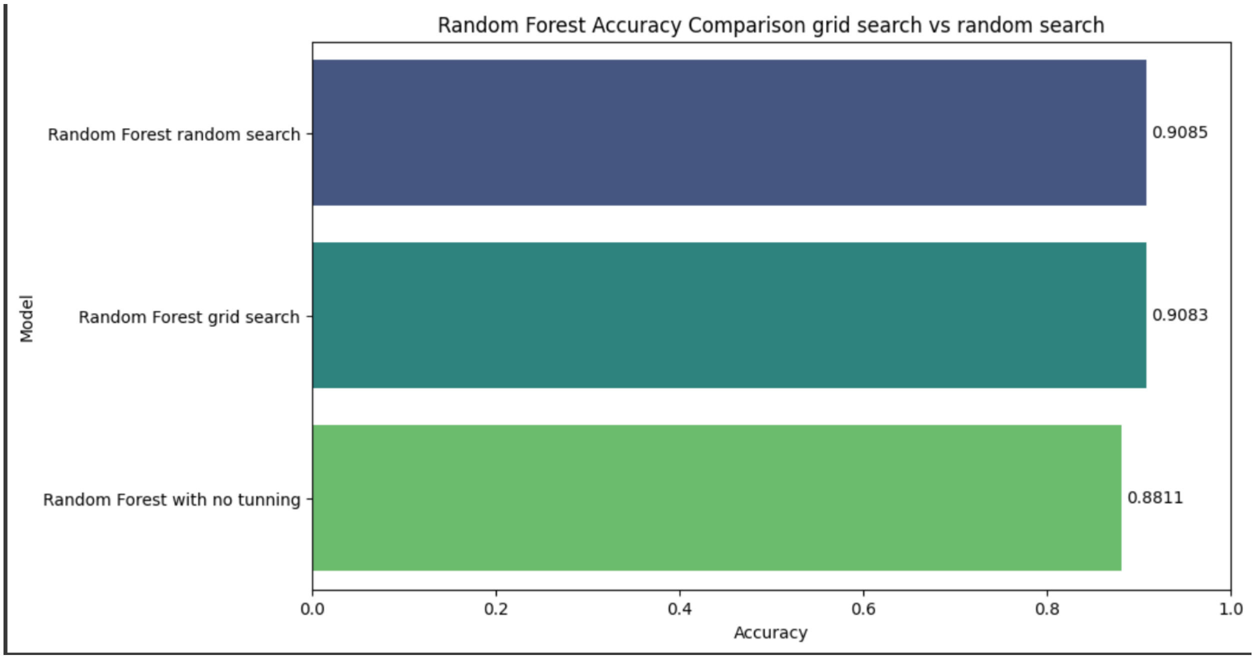
with hyperparameter tuning



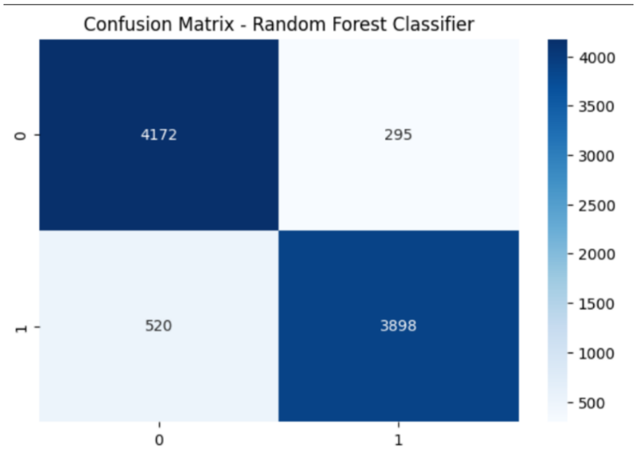
```
Performing Grid Search for Random Forest Classifier
Best hyperparameters for Random Forest Classifier: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Accuracy on the test set for Random Forest Classifier: 0.9082723691615081
```



```
                    Model  Accuracy  Precision   Recall  F1 Score   AUC-ROC
Random Forest Classifier  0.908272   0.929645  0.88230  0.905354  0.960621
```



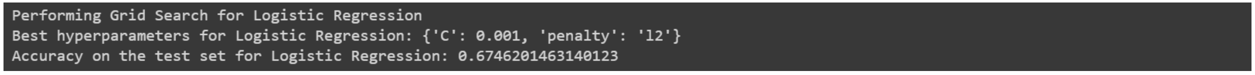
The confusion matrix for the random forest classifier shows that the model correctly predicted 4172 positive examples and 3898 negative examples. It also incorrectly predicted 295 negative examples as positive and 1500 positive examples as negative.



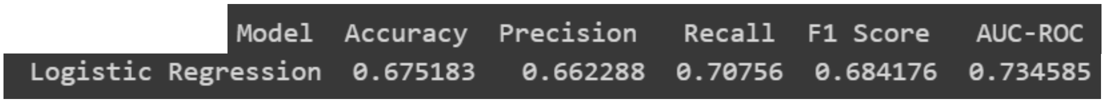
2. Logistic Regression:

Logistic Regression is a linear model for binary classification that predicts the probability of an instance belonging to a particular class. It uses the logistic function to constrain the output between 0 and 1, making it suitable for binary classification tasks.

Grid search is employed to find the optimal combination of hyperparameters, optimizing the model's performance.

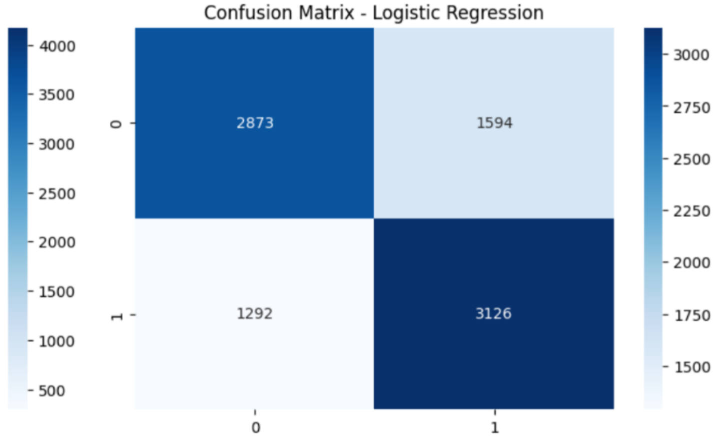


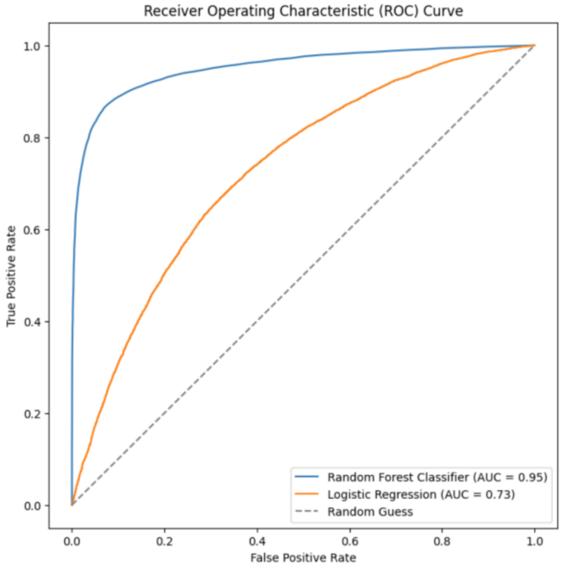
```
Performing Grid Search for Logistic Regression
Best hyperparameters for Logistic Regression: {'C': 0.001, 'penalty': 'l2'}
Accuracy on the test set for Logistic Regression: 0.6746201463140123
```



Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
Logistic Regression	0.675183	0.662288	0.70756	0.684176	0.734585

The confusion matrix for the logistic regression classifier shows that the model correctly predicted 2873 positive examples and 3126 negative examples. It also incorrectly predicted 1594 negative examples as positive and 1750 positive examples as negative



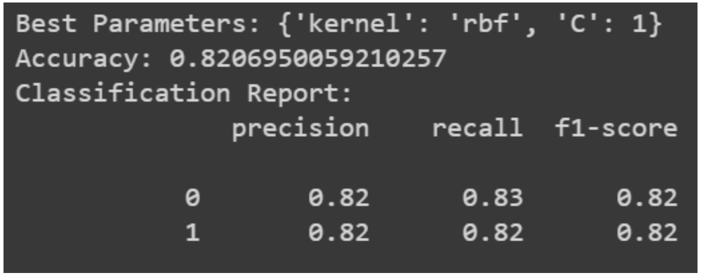


The random forest classifier has an AUC of 0.95, which is significantly higher than the AUC of 0.73 for the logistic regression classifier. This means that the random forest classifier is much better at correctly identifying positive cases without also incorrectly identifying negative cases as positive.

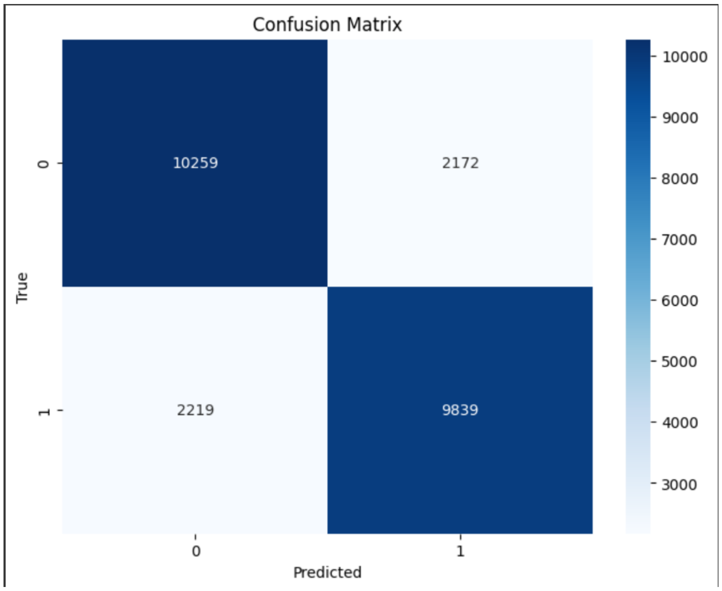
3. Suport Vector Machine (SVM):

Support Vector Machine (SVM) is a powerful supervised learning algorithm that finds a hyperplane that best separates classes in a high-dimensional space.

The SVM model is trained on the dataset, and randomized search is employed to find the optimal hyperparameters, improving the model's performance.



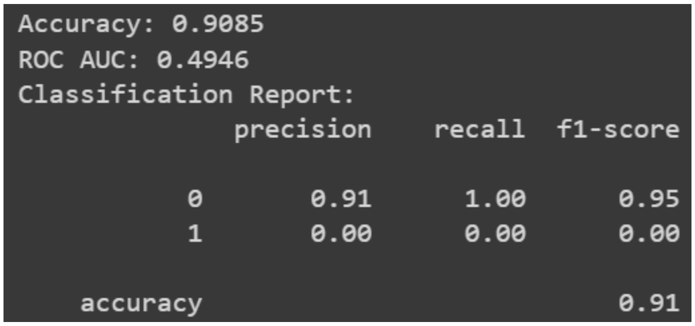
```
Best Parameters: {'kernel': 'rbf', 'C': 1}
Accuracy: 0.8206950059210257
Classification Report:
              precision    recall  f1-score

           0       0.82      0.83      0.82
           1       0.82      0.82      0.82
```



4. Recurrent Neural Network (RNN):

Recurrent Neural Networks (RNNs) are a class of neural networks designed for sequence-based data. They have connections with cycles, allowing information persistence over time.

The RNN model is trained on sequential data using the Adam optimizer and binary cross-entropy loss. Its performance is evaluated using metrics such as accuracy, F1 score, and ROC AUC.

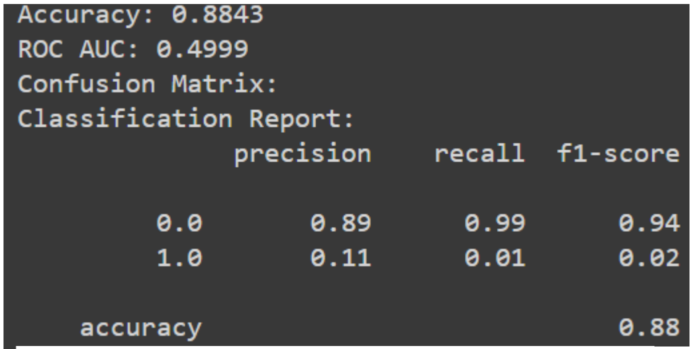


```
Accuracy: 0.9085
ROC AUC: 0.4946
Classification Report:
              precision    recall  f1-score

           0       0.91      1.00      0.95
           1       0.00      0.00      0.00

    accuracy                           0.91
```

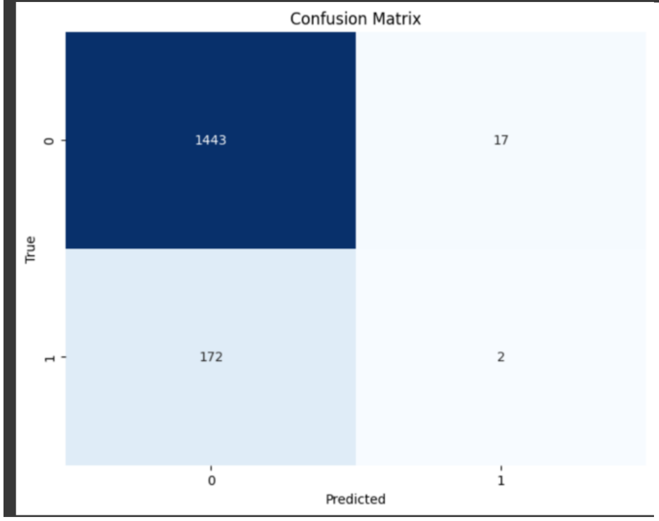
5. K-Nearest Neighbors (KNN):

Definition:

K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm used for classification and regression tasks. It classifies an instance based on the majority class of its k-nearest neighbors in the feature space.KNN is trained on the dataset, and its performance is evaluated using accuracy, confusion matrix, and a detailed classification report.



```
Accuracy: 0.8843
ROC AUC: 0.4999
Confusion Matrix:
Classification Report:
              precision    recall  f1-score

         0.0       0.89      0.99      0.94
         1.0       0.11      0.01      0.02

    accuracy                           0.88
```



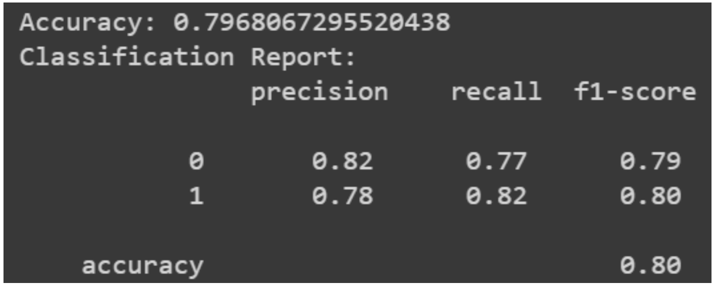
6. Decision Tree:

Definition:

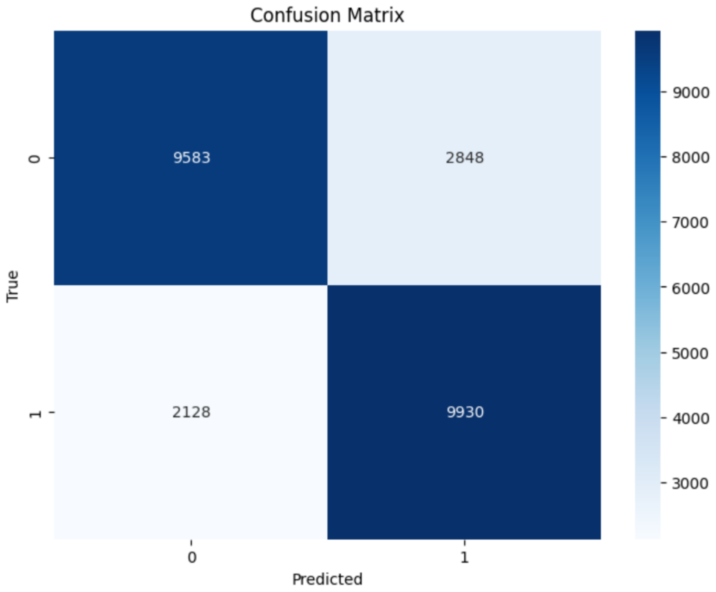
A Decision Tree is a flowchart-like structure where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents the outcome.

Training and Evaluation:

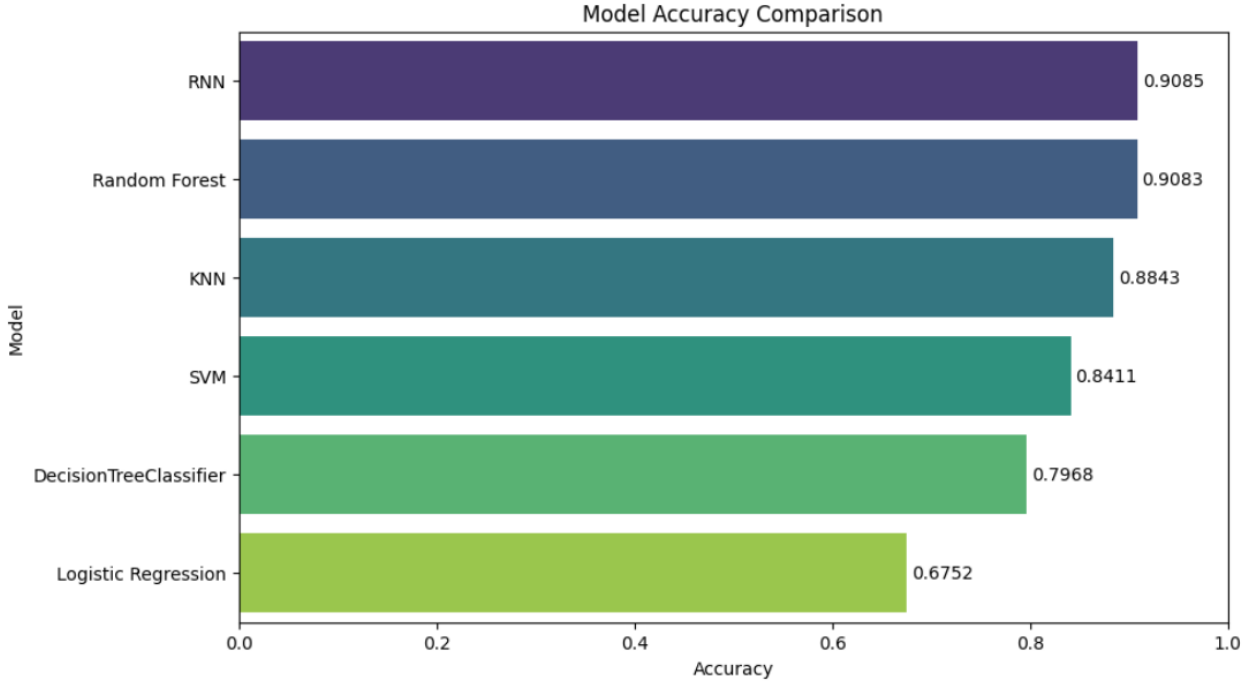
The Decision Tree model is trained on the dataset, and its performance is evaluated using accuracy, confusion matrix, and a detailed classification report.

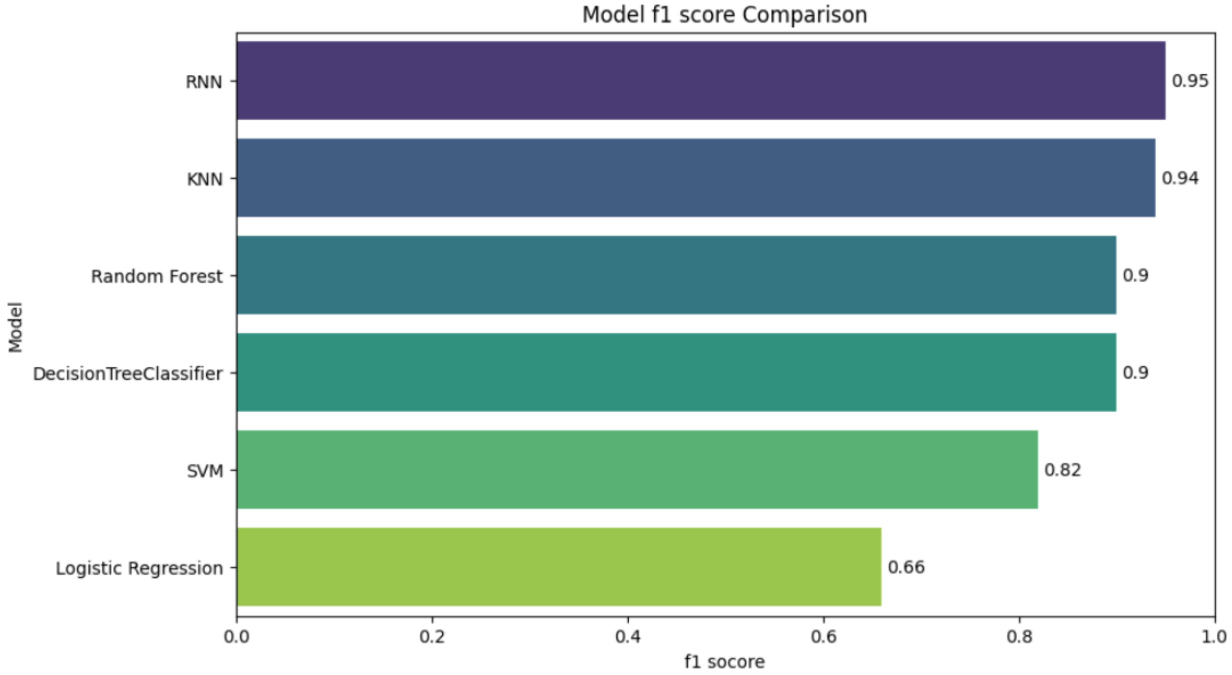


```
Accuracy: 0.7968067295520438
Classification Report:
              precision    recall  f1-score

           0       0.82      0.77      0.79
           1       0.78      0.82      0.80

    accuracy                           0.80
```



Model Comparison and Discussion:





In comparing the performance metrics of six distinct machine learning models applied to the loan status prediction task, a comprehensive evaluation emerges. The Random Forest and RNN models stand out prominently, achieving high accuracy scores of 90.83% and 90.85%, respectively. These models are complemented by robust F1 scores of 0.9 and 0.95, underlining their proficiency in striking a balance between precision and recall. Logistic Regression and SVM, while demonstrating moderate accuracy at 67.52% and 84.11%, respectively, display F1 scores of 0.66 and 0.82. These findings suggest that while Logistic Regression may struggle to capture the intricacies of the dataset, the SVM model achieves a commendable balance between precision and recall. The KNN model delivers an accuracy of 88.43%, paired with a solid F1 score of 0.94, showcasing its efficacy in capturing local patterns within the data. The Decision Tree model, with an accuracy of 79.68% and an F1 score of 0.90, presents a respectable performance but lags slightly behind ensemble models like Random Forest.

Conclusion

In summary, this study aimed to address loan defaults by utilizing data-driven approaches and developing predictive models with six machine learning algorithms. The analysis involved cleaning and preprocessing the dataset, exploring its characteristics, and tuning the models. Random Forest and RNN emerged as top performers, achieving high accuracy and F1 scores. Logistic Regression and SVM demonstrated moderate accuracy but displayed a balance between precision and recall. KNN showcased efficacy in capturing local patterns, while Decision Tree presented a respectable performance. The study provides valuable insights into the strengths and limitations of each model, offering financial institutions guidance in implementing more accurate credit evaluation systems. By leveraging advanced data-driven models, institutions can better assess the creditworthiness of loan applicants and mitigate the impact of defaults, contributing to a more stable economic environment.