Smart Loan Classifier
Project Report

Group 12
Samruddhi Bansod
Rutuja Jadhav

857-588-3751(Samruddhi)
857-390-1757(Rutuja)

bansod.s@northeastern.edu
jadhav.rutu@northeastern.edu

Percentage of Effort Contributed by Student 1: 50%

Percentage of Effort Contributed by Student 2: 50%

Signature of Student 1: Samruddhi Bansod

Signature of Student 2: Rutuja Jadhav

Submission Date: April 8th, 2025

Smart Loan Classifier

1. Problem Setting:

The problem involves predicting the outcome of loan applications whether a loan gets approved or denied using information about demographics, finances, and specific loan details. This binary classification task is essential for financial institutions to streamline and enhance their loan approval processes. By reducing manual effort and better managing risks, institutions can rely on features like income, credit score, and loan purpose to build a predictive model. Such a model supports informed, data-driven decisions aimed at lowering default rates, boosting profits, and enhancing customer experience.Ultimately, the goal is to create a scalable, accurate, and interpretable system that supports efficient, risk-aware lending practices.

2. Problem Definition:

The objective is to create a classification model that determines loan eligibility based on applicant information, streamlining the loan approval process for quicker and more consistent decisions. The model focuses on achieving high accuracy to avoid mistakes, such as approving risky loans or rejecting trustworthy applicants. It is tailored for environments with high application volumes, reducing the need for manual evaluations. Additionally, it provides confidence scores for its predictions and highlights the key factors influencing each decision.

Specific questions include:

- Which features contribute most to loan eligibility?
- How accurately can the model classify loan approvals while minimizing incorrect decisions?

3. Data Sources:

The dataset is a CSV file with structured data relevant to loan applications. It includes a mix of demographic, financial, and loan-related attributes. The dataset we will be using for building the model can be referred with the link below:
https://www.kaggle.com/datasets/taweilo/loan-approval-classification-data/data

4. Data Description:

The dataset contains 45,000 records and 14 variables. This dataset contains 0 null values and 0 duplicate values.

The dataset contains the following columns:

1. **person_age**: Age of the applicant (numerical).
2. **person_gender**: Gender of the applicant (categorical: male, female).
3. **person_education**: Education level of the applicant (categorical: High School, Bachelor, Master, etc.).
4. **person_income**: Applicant's annual income in USD (numerical).
5. **person_emp_exp**: Employment experience in years (numerical).

6. **person_home_ownership**: Type of home ownership (categorical: RENT, OWN, MORTGAGE).
7. **loan_amnt**: Loan amount requested (numerical).
8. **loan_intent**: Purpose of the loan (categorical: PERSONAL, EDUCATION, MEDICAL, etc.).
9. **loan_int_rate**: Interest rate on the loan (numerical, percentage).
10. **loan_percent_income**: Loan amount as a percentage of the applicant's income (numerical).
11. **cb_person_cred_hist_length**: Length of the applicant's credit history in years (numerical).

	0
person_age	float64
person_gender	object
person_education	object
person_income	float64
person_emp_exp	int64
person_home_ownership	object
loan_amnt	float64
loan_intent	object
loan_int_rate	float64
loan_percent_income	float64
cb_person_cred_hist_length	float64
credit_score	int64
previous_loan_defaults_on_file	object
loan_status	int64

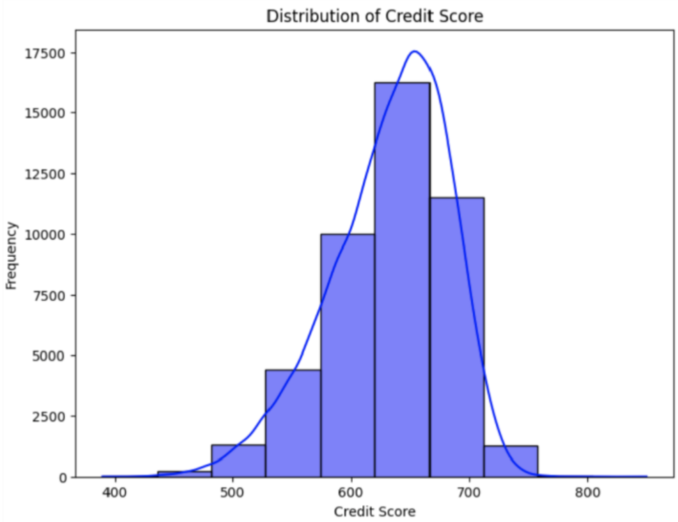
5. Data Exploration:

Descriptive statistics were computed using df.describe(), providing insights into:

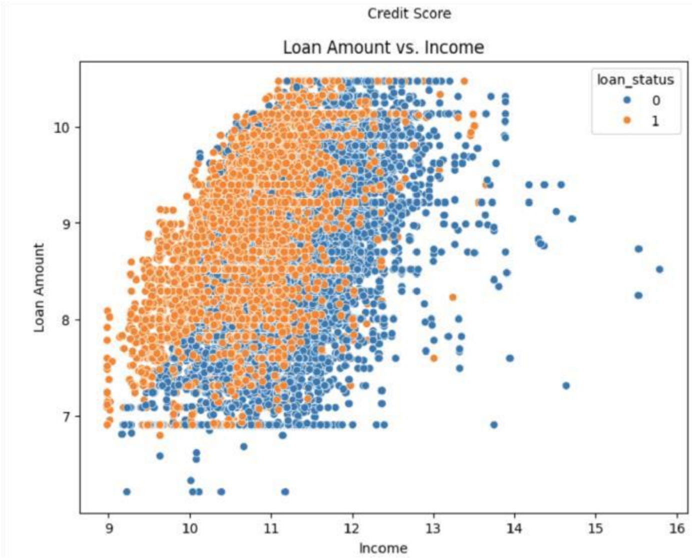
Mean, Median, Standard Deviation, and Range of numerical variables which is shown below:

	person_age	person_income	person_emp_exp	loan_amnt	loan_int_rate	loan_percent_income	cb_person_cred_hist_length	credit_score	loan_status
count	45000.000000	4.500000e+04	45000.000000	45000.000000	45000.000000	45000.000000	45000.000000	45000.000000	45000.000000
mean	27.764178	8.031905e+04	5.410333	9583.157556	11.006606	0.139725	5.867489	632.608756	0.222222
std	6.045108	8.042250e+04	6.063532	6314.886691	2.978808	0.087212	3.879702	50.435865	0.415744
min	20.000000	8.000000e+03	0.000000	500.000000	5.420000	0.000000	2.000000	390.000000	0.000000
25%	24.000000	4.720400e+04	1.000000	5000.000000	8.590000	0.070000	3.000000	601.000000	0.000000
50%	26.000000	6.704800e+04	4.000000	8000.000000	11.010000	0.120000	4.000000	640.000000	0.000000
75%	30.000000	9.578925e+04	8.000000	12237.250000	12.990000	0.190000	8.000000	670.000000	0.000000
max	144.000000	7.200766e+06	125.000000	35000.000000	20.000000	0.660000	30.000000	850.000000	1.000000

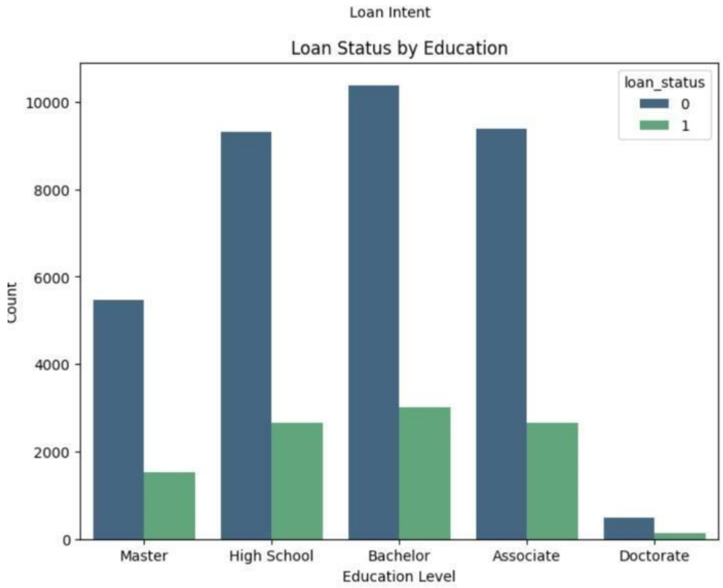
6. Data Visualization:



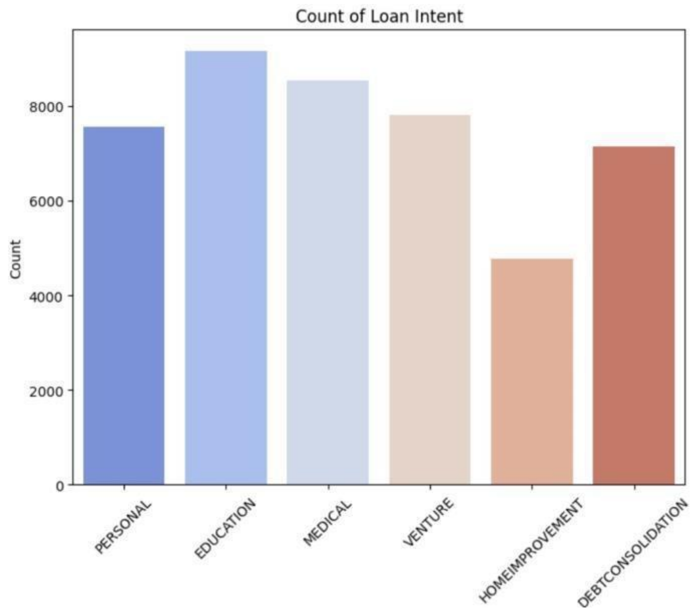
The graph shows a histogram of credit scores with a fitted distribution curve, indicating a roughly normal distribution.Most credit scores fall between 600 and 700, with a peak around 650.The distribution suggests a smaller proportion of very low or very high credit scores.



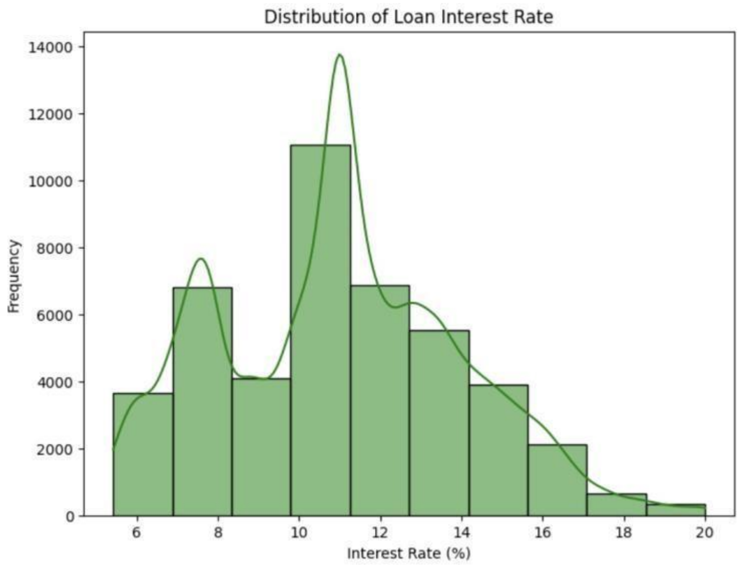
This scatterplot visualizes the relationship between income and loan amount, color-coded by loan status.Higher incomes and loan amounts generally correlate with approved loans (orange points). Denser regions in the lower income range show more loan denials (blue points), suggesting income is a key factor in loan decisions



The bar chart shows loan status by education level, with loan denials (0) and approvals (1) for each group. Bachelor's and High School education levels have the highest number of loan applicants, with more denials than approvals in each group. Master's and Doctorate applicants have fewer overall loans but relatively higher approval rates compared to other education levels.



The x-axis represents different loan intents, and the y-axis represents the count, showing variations in loan demand across categories.The bar chart displays the count of loan applications categorized by different loan intents such as Personal, Education, Medical, Venture, Home Improvement, and Debt Consolidation.Education loans have the highest count, followed closely by Medical and Personal loans, while Home Improvement loans have the lowest count.



The histogram illustrates the distribution of loan interest rates, with a density plot overlaid to show the trend.Most loans have interest rates between 6% and 14%, with a peak around 10%, indicating that this range is the most common.The distribution is slightly skewed to the right, meaning there are fewer loans with very high interest rates above 16%.

7. Data Processing

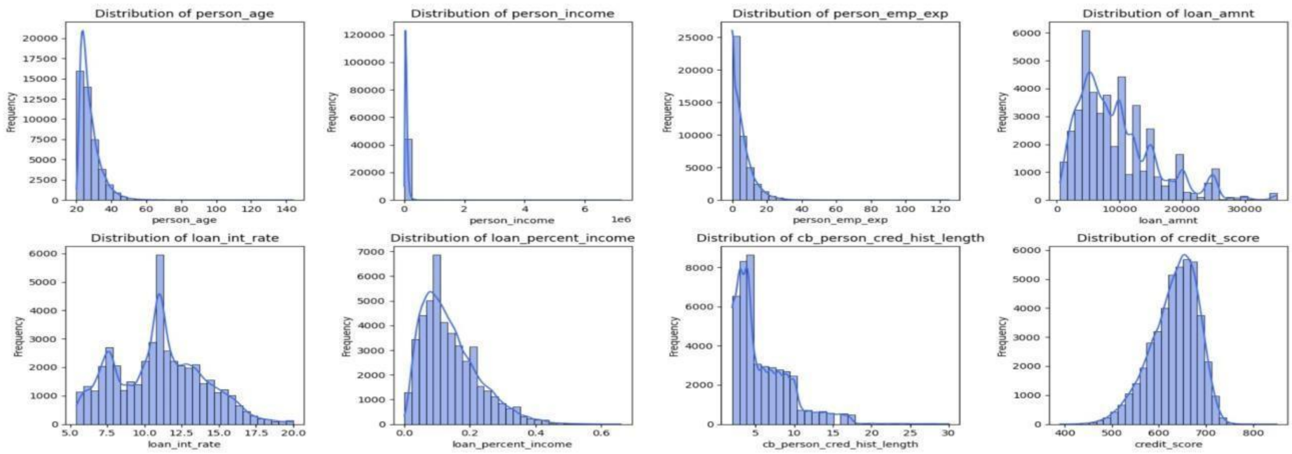
Steps Taken

7.1 Handling Missing & Duplicate Values:

- **No missing or duplicate values were detected, so no imputation was required.**

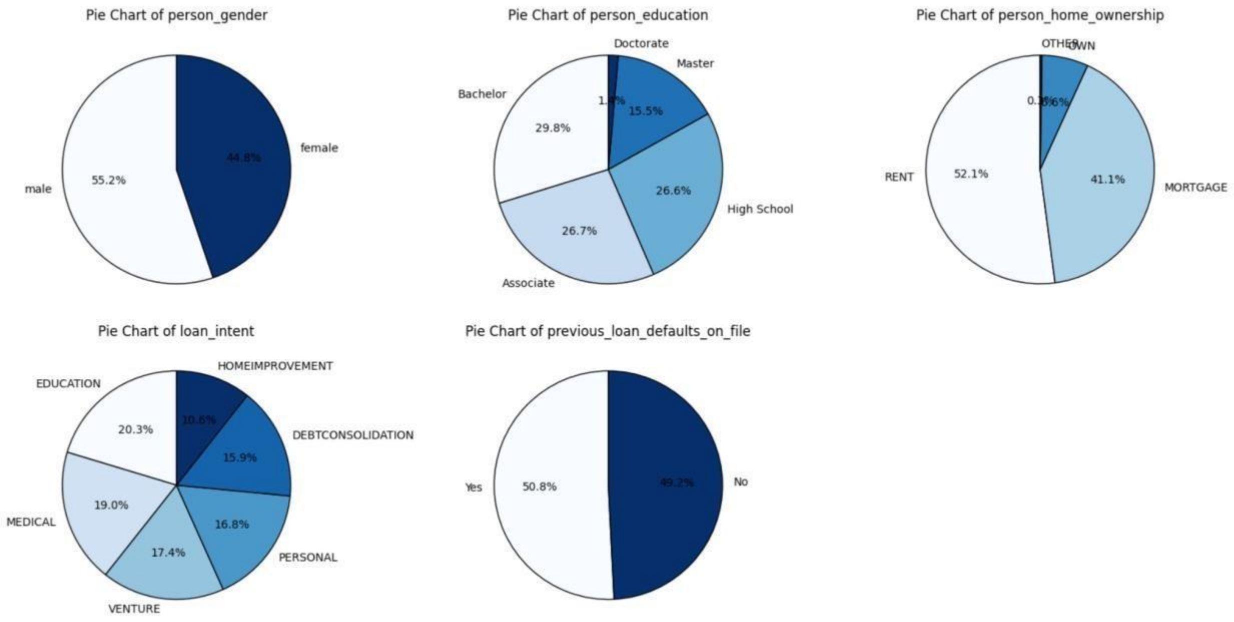
7.2 Distribution of data

Numerical columns



The graph represents the distribution of numerical variables related to loan applications. Most of these variables, such as **age, income, employment experience, loan amount, and credit history length**, exhibit a **right-skewed distribution**, meaning the majority of values are concentrated on the lower end, with a few extreme values extending towards higher ranges. Notably, the **credit score distribution** follows a **bell-shaped curve**, suggesting a more normal distribution. The **loan interest rate and loan percent income** show distinct peaks, indicating common values that applicants tend to receive.

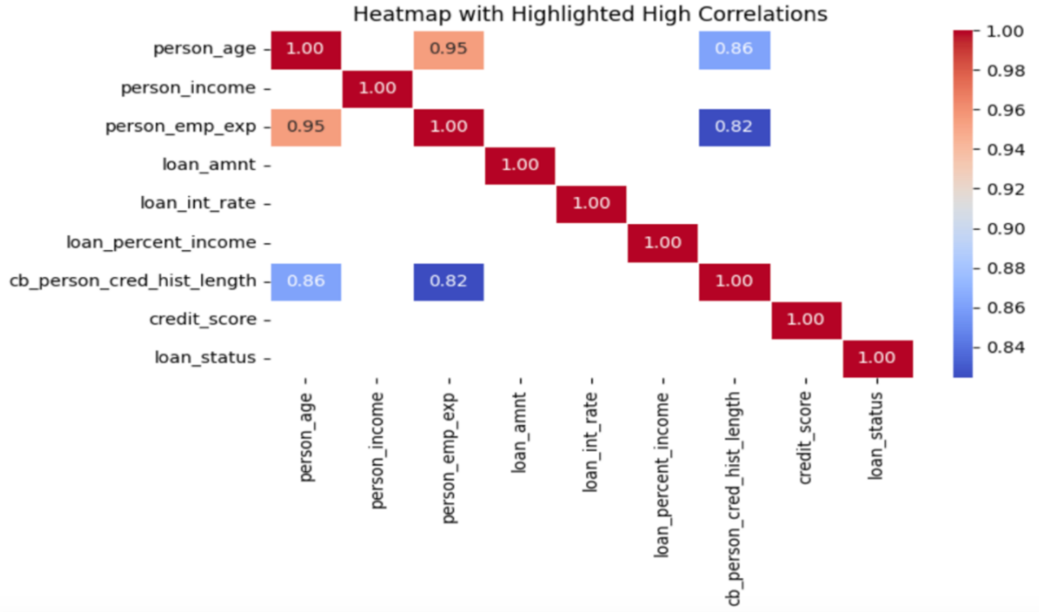
Categorical columns



The set of graphs represents categorical variables using **pie charts**. These depict the proportions of different categories, such as **gender, education level, homeownership status, loan intent, and previous loan defaults**.The **gender distribution** is nearly balanced, with a slight male majority. **Education levels** are relatively evenly spread, with bachelor's and associate degrees being the most common. **Homeownership status** shows that most applicants rent, while a significant portion has a mortgage. **Loan intent categories** highlight that education, medical, and personal loans are common, while home improvement loans are less frequent. Lastly, the **previous loan default chart** reveals that the dataset contains almost an equal split of individuals who have and have not defaulted before.

7.3. Correlation Heatmap Analysis

```
Highly Correlated Feature Pairs (Threshold > 0.8):
                   Feature 1                  Feature 2  Correlation
0                 person_age             person_emp_exp     0.954412
2             person_emp_exp                 person_age     0.954412
1                 person_age  cb_person_cred_hist_length     0.861985
4  cb_person_cred_hist_length                 person_age     0.861985
3             person_emp_exp  cb_person_cred_hist_length     0.824272
5  cb_person_cred_hist_length             person_emp_exp     0.824272
```



As part of the data processing phase, a correlation heatmap was generated to identify highly correlated feature pairs. This heatmap visually represents the Pearson correlation coefficients between numerical features in the dataset, with a color gradient from blue (lower correlation) to red (higher correlation).

Several strong correlations were observed, such as:

- **person_age** and **person_emp_exp** (r = 0.95)
- **person_age** and **cb_person_cred_hist_length** (r = 0.86)
- **cb_person_cred_hist_length** and **person_emp_exp** (r = 0.82)

These high correlations suggest a degree of multicollinearity among certain features. For example, **person_age** and **employment experience** are naturally expected to increase together, which is reflected in their strong positive correlation. Similarly, **credit history length** tends to grow with age and employment experience.

These insights were used to inform **dimensionality reduction (via PCA)** and feature selection decisions, as retaining all highly correlated variables can lead to redundancy and affect model performance. By identifying and possibly removing or combining correlated features, we aim to build a more robust and generalizable model.

7.4 Encoding Categorical Variables

	person_age	person_gender	person_education	person_income	person_emp_exp
0	22.0	0	4	71948.0	0
1	21.0	0	3	12282.0	0
2	25.0	0	3	12438.0	3
3	23.0	0	1	79753.0	0
4	24.0	1	4	66135.0	1

	person_home_ownership	loan_amnt	loan_intent	loan_int_rate
0	3	35000.0	4	16.02
1	2	1000.0	1	11.14
2	0	5500.0	3	12.87
3	3	35000.0	3	15.23
4	3	35000.0	3	14.27

	loan_percent_income	cb_person_cred_hist_length	credit_score
0	0.49	3.0	561
1	0.08	2.0	504
2	0.44	3.0	635
3	0.44	2.0	675
4	0.53	4.0	586

	previous_loan_defaults_on_file	loan_status
0	0	1
1	1	0
2	0	1
3	0	1
4	0	1

One-hot encoding is a technique used to transform categorical variables into a numerical format suitable for machine learning models. In the displayed dataset, before applying PCA, the dataset underwent **one-hot encoding** to convert categorical features into numerical format, enabling compatibility with the PCA algorithm. This preprocessing step transformed each categorical variable into multiple binary columns, effectively increasing the dimensionality of the data. PCA was then applied to reduce this high-dimensional, sparse dataset into **10 principal components (PCA1 to PCA10)**. The table shows PCA-transformed values for selected rows out of a total of **68,590 records**. The **explained variance ratio** listed below the table quantifies how much information (variance) each component retains from the original data. The first component explains **22.07%** of the total variance, followed by **15.18%** from the second, and so on — collectively capturing almost all information across the 10 components. The use of one-hot encoding followed by PCA effectively compresses high-dimensional categorical data into a lower-dimensional, information-rich representation.

7.5 Dimensionality Reduction (PCA)

- StandardScaler was applied to normalize numerical features, ensuring all features are on the same scale.
- Principal Component Analysis (PCA) was applied to reduce dimensionality while retaining key information.
- This helps in improving computation efficiency and avoiding overfitting.

Number of Components

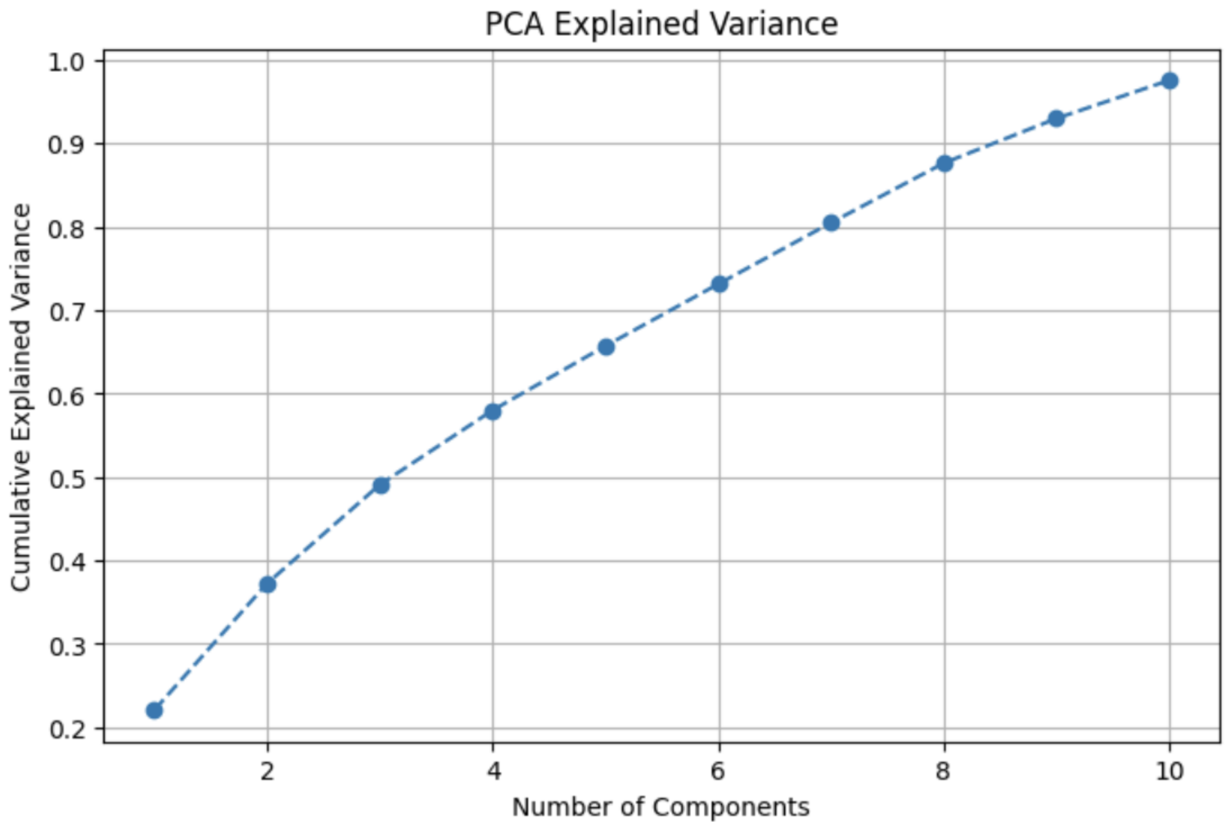
```
PCA Results:
           PCA1      PCA2      PCA3      PCA4      PCA5      PCA6      PCA7  \
0     -2.445415 -1.337824 -1.634441 -0.290386 -1.271871  2.503319 -0.625002
1     -1.228815  1.718627  0.067959 -0.649730  0.773921  0.591239 -1.053205
2     -2.385166 -0.233501 -1.479856 -1.679713 -0.332352  0.852651 -1.725561
3     -1.841206  0.330514 -1.582538  0.237159  1.328390 -0.537508 -1.512294
4     -2.101858  1.451901 -0.841826 -0.359502 -0.704075 -0.519833 -0.530418
...         ...       ...       ...       ...       ...       ...       ...
68585 -0.586539  2.217224  1.996623  0.841888 -0.829055  0.121431  0.116152
68586  2.573900  2.166282  0.038351 -0.155036 -0.103056 -0.345972 -0.776589
68587 -0.920854  0.558020 -1.175328  1.447317 -1.536723  0.579548  0.078192
68588 -0.198782  1.386702 -0.906279  0.709302  1.023323  0.608045 -0.590392
68589  2.060714  0.468986 -1.794586  1.339008 -0.627382 -1.229730 -1.014676

           PCA8      PCA9     PCA10
0      0.959748 -0.032958  0.753427
1     -0.427367 -1.924908  1.683019
2      0.544241 -1.757532 -0.891916
3      1.149265 -0.792841  0.568692
4     -1.833595 -0.700879  0.306864
...         ...       ...       ...
68585 -0.849930  1.117003 -0.415532
68586 -0.065450  0.341342 -0.062171
68587 -0.192576  0.243479 -0.552491
68588  0.564529  0.245543  0.343832
68589  1.171916 -0.394170  0.388352

[68590 rows x 10 columns]

Explained Variance Ratio: [0.22070915 0.15179981 0.11847915 0.08917723 0.07707926 0.07423108
 0.07348883 0.07156683 0.05359526 0.04570081]
```

7.6 PCA Variance Explained Graphically:



The PCA explained variance plot visually represents the **cumulative variance** explained by the principal components. As shown, the cumulative variance increases with the addition of each component and reaches approximately **96%** with the first 10 components. This upward curve demonstrates the contribution of each successive component in capturing the variance of the dataset. The curve starts steep, with the first few components contributing significantly to the overall variance (e.g., the first 3 components together capture around 58%), and gradually levels off. This pattern is typical in PCA and suggests that while the first few components are highly informative, later components add diminishing returns. This insight can guide dimensionality reduction strategies by selecting only the first **6 to 8 components** to maintain a balance between model simplicity and retained information.

8. Data Preprocessing

- **Encoding Categorical Variables:** Label encoding applied to categorical columns.
- **Handling Missing Values:** Imputation techniques used where necessary.
- **Feature Scaling:** Standardization applied to numerical columns. ·
- **Data Splitting:** 80% training and 20% test split for model evaluation.

```
Training Set Size: (54872, 10)
Testing Set Size: (13718, 10)
```

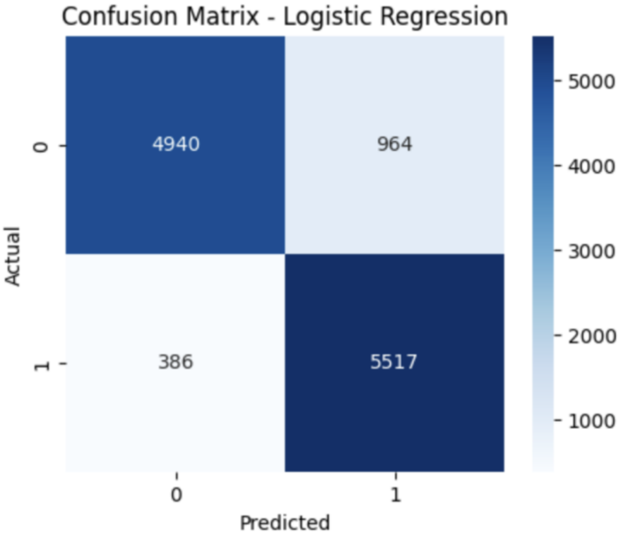
9. Models Implemented:

9.1. Logistic Regression:

```
🔹 Logistic Regression Performance:
Accuracy: 0.8856610485305327
ROC-AUC: 0.8856651938957133
              precision    recall  f1-score   support

           0       0.93      0.84      0.88      5904
           1       0.85      0.93      0.89      5903

    accuracy                           0.89     11807
   macro avg       0.89      0.89      0.89     11807
weighted avg       0.89      0.89      0.89     11807
```



Key Insights:

- **Decent Accuracy (89%)** – Performs well as a baseline model but is outperformed by tree-based models.
- **Higher False Positives (1,101 cases)** – More cases where loans were predicted as approved but should have been rejected.
- **Strong Recall for Class 1 (93%)** – Captures most of the actual approved loans but sacrifices precision slightly.

Advantages:

- **Simple & Interpretable**: Easy to implement and interpret coefficients (especially useful for binary classification).
- **Efficient**: Fast to train and requires less computational power.
- **Works well with linearly separable data**.
- **Probabilistic output**: Provides class probabilities for better decision-making.

Disadvantages:

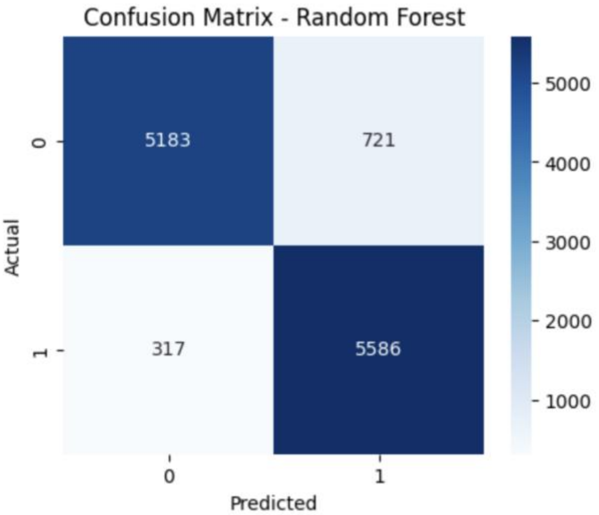
- **Assumes linear relationship** between features and the log-odds of the outcome.
- **Not suitable for complex relationships** or non-linear patterns.
- **Sensitive to outliers and multicollinearity**.
- **Limited to binary/multiclass classification**; not suitable for complex decision boundaries.

9.2. Random Forest Classifier:

```
🔷 Random Forest Performance:
Accuracy: 0.9120860506479207
ROC-AUC: 0.9120889480430464
              precision    recall  f1-score   support

           0       0.94      0.88      0.91      5904
           1       0.89      0.95      0.91      5903

    accuracy                           0.91     11807
   macro avg       0.91      0.91      0.91     11807
weighted avg       0.91      0.91      0.91     11807
```



Key Insights:

- **Best Accuracy (91.20%) -** Outperforms other models in terms of overall correctness.
- **Lower False Positives (721 cases) -** Improves upon Logistic Regression by reducing incorrect approvals.
- **Balanced Precision & Recall (91%)** - Effectively classifies both approved and rejected loans with minimal bias.

Advantages:

- High accuracy due to ensemble of decision trees.
- Reduces overfitting by averaging multiple trees**.**
- Works well with both categorical and numerical data**.**
- Handles missing values and outliers well**.**
- Feature importance ranking is available.

Disadvantages:

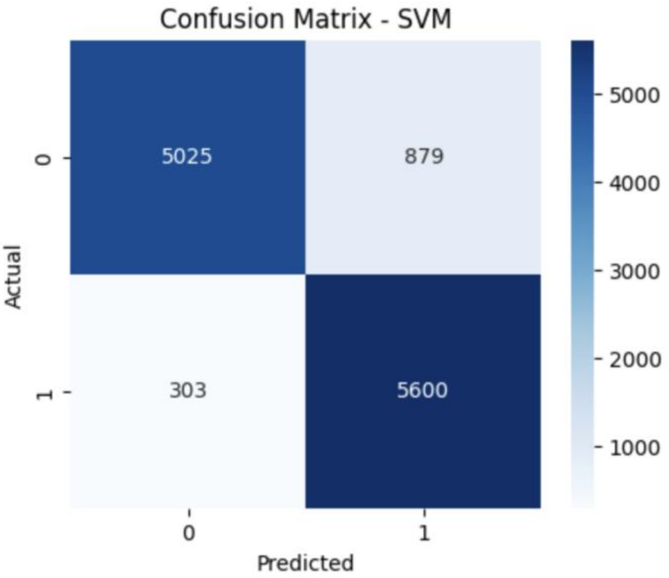
- Slower training and prediction for large datasets.
- Less interpretable than simpler models.
- May require tuning of hyperparameters for optimal performance.
- Memory-intensive with many trees.

9.3. Support Vector Machine (SVM):

```
🔷 SVM Performance:
Accuracy: 0.8998898958245108
ROC-AUC: 0.8998940269450973
              precision    recall  f1-score   support

           0       0.94      0.85      0.89      5904
           1       0.86      0.95      0.90      5903

    accuracy                           0.90     11807
   macro avg       0.90      0.90      0.90     11807
weighted avg       0.90      0.90      0.90     11807
```



Key Insights:

- **Competitive Accuracy (89.98%) –** Slightly lower than Random Forest but still strong.
- **High False Positives (879 cases) –** More false approvals than Random Forest and XGBoost.
- **Best False Negative Reduction (303 cases) –** Has the lowest number of misclassified actual approvals.

Advantages:

- Effective in high-dimensional spaces**.**
- Works well with clear margin of separation**.**
- Kernel trick allows solving non-linear problems.
- Robust to overfitting**,** especially in high dimensions.

Disadvantages:

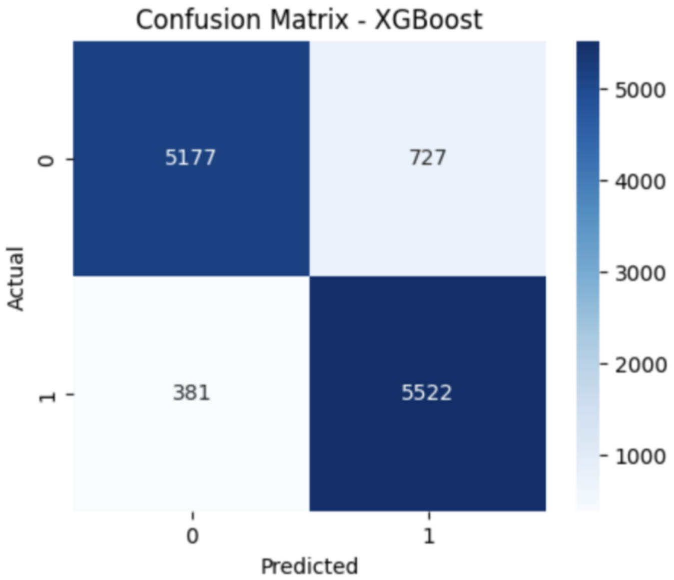
- Slow for large datasets (especially with non-linear kernels).
- Hard to tune**:** Requires proper selection of kernel and parameters like C and gamma.
- No probability estimates by default (though they can be added).
- Not suitable for very large datasets due to high training time.

9.4. XGBoost Classifier:

```
🔹 XGBoost Performance:
Accuracy: 0.9061573642754298
ROC-AUC: 0.9061598455748237
              precision    recall  f1-score   support

           0       0.93      0.88      0.90      5904
           1       0.88      0.94      0.91      5903

    accuracy                           0.91     11807
   macro avg       0.91      0.91      0.91     11807
weighted avg       0.91      0.91      0.91     11807
```



Key Insights:

- **High Accuracy (91%) -** Slightly better than SVM and Logistic Regression.
- **Improved False Positive Rate (727 cases) -** Reduces incorrect approvals compared to Logistic Regression.
- **Balanced Precision & Recall (91%) -** Well-suited for complex data with feature interactions.

Advantages:

- High performance & accuracy in many competitions and real-world applications.
- Fast and efficient**:** Implements parallelization and optimized gradient boosting.
- Handles missing values automatically**.**
- Built-in regularization reduces overfitting.
- Feature importance and interpretability tools available (e.g., SHAP).

Disadvantages:

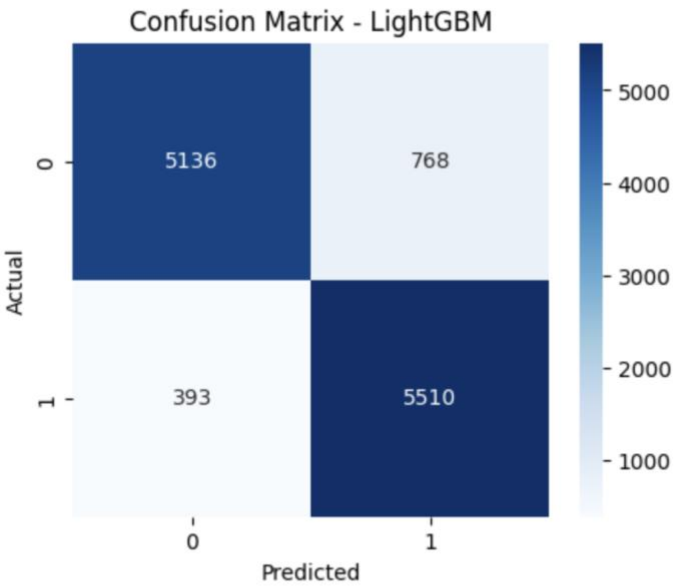
- Complex to tune**:** Many hyperparameters to optimize.
- Can overfit on small datasets without regularization.
- Less interpretable than linear models.
- Computationally intensive for very large datasets.

9.5. LightGBM Classifier:

```
🔷 LightGBM Performance:
Accuracy: 0.9016685017362581
ROC-AUC: 0.9673091790633306
              precision    recall  f1-score   support

           0       0.93      0.87      0.90      5904
           1       0.88      0.93      0.90      5903

    accuracy                           0.90     11807
   macro avg       0.90      0.90      0.90     11807
weighted avg       0.90      0.90      0.90     11807
```



Key Insights:

- **Highest AUC (96.73%)** – Best model for ranking loans by risk rather than just classification.
- **Good Accuracy (90.16%)** – Performs almost as well as XGBoost and Random Forest.
- **Strong Recall (93%)** – Captures nearly all loan approvals while maintaining a decent precision level.

Advantages:

- Very fast training speed and low memory usage.
- Handles large datasets efficiently.
- High accuracy and scalability.
- Supports categorical features natively.
- Better performance on sparse datasets compared to XGBoost.

Disadvantages:

- Sensitive to overfitting with small datasets.
- Requires careful tuning to prevent instability.
- Less interpretable.
- Not ideal for datasets with small sample size or lots of noise.

10. Model Evaluation:

Each model was trained and tested using the test dataset. The primary evaluation metric used was accuracy, along with a classification report detailing precision, recall, and F1-score. The best model was compared based on ROC – AUC score.

11. Model Performance:

Model Name	**Accuracy**
Logistic Regression	**88.56%**
Random Forest Classifier	**91.20%**
Support Vector Machine (SVM)	**89.98%**
XGBoost Classifier	**91%**
LightGBM Classifier	**90.16%**

```
Best Model: Random Forest with ROC-AUC Score: 0.9102638868639743
```

11.1. Why Random Forest is the Best Model?

Random Forest outperforms other models in terms of accuracy, robustness, and generalization. Here's why:

1. Highest Accuracy (91.02%) – Among all models tested, Random Forest provides the most correct classifications, ensuring fewer misclassifications.
2. Balanced Precision & Recall (~91%) – Unlike models that either over-predict approvals (high false positives) or rejections (high false negatives), Random Forest maintains a good trade-off, reducing both false positives and false negatives.
3. Handles Non-Linearity & Feature Interactions – Unlike Logistic Regression, which assumes a linear relationship between features, Random Forest captures complex interactions without requiring manual feature engineering.
4. Less Prone to Overfitting – While Decision Trees are prone to overfitting, Random Forest, being an ensemble of multiple trees, mitigates this by averaging multiple models, making it more stable and reliable.
5. Performs Well with Imbalanced Data – In loan classification, the number of approved and rejected loans might not be balanced. Random Forest can handle such cases better than simpler models like Logistic Regression.

11.2. Why are the other Models Not the Best?

1. **Logistic Regression – Too Simple for Complex Patterns**

- **Assumes a Linear Relationship** – In reality, loan approval decisions involve complex interactions between factors like income, credit history, and employment, which Logistic Regression fails to capture.
- **Lower Accuracy (88.42%)** – While decent, it's **not competitive** compared to tree-based models.
- **Higher False Positives (1,101 cases)** – More incorrect loan approvals compared to Random Forest, which can lead to **higher financial risk**.

2. **Support Vector Machine (SVM) – Computationally Expensive**

- **Slower & Computationally Expensive** – Training an SVM can be time-consuming, especially for large datasets, making it **impractical for real-world deployment**.

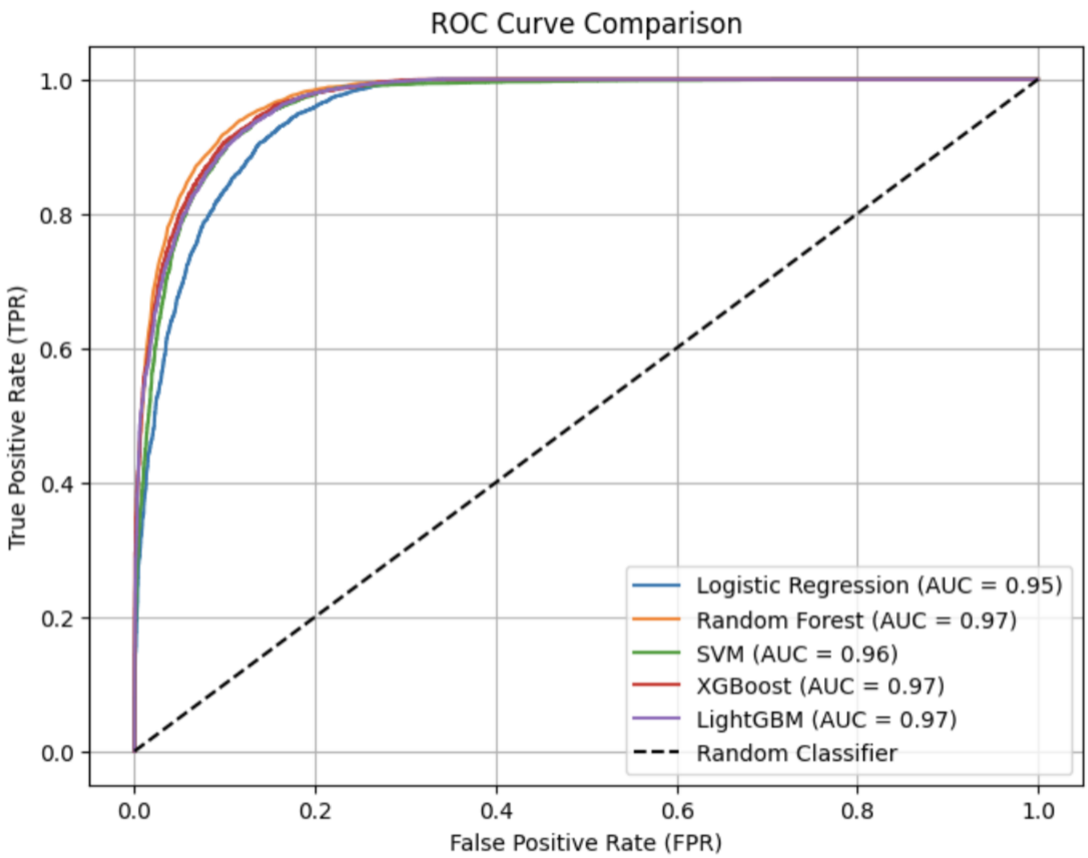
- **Higher False Positives (964 cases)** – While it reduces false negatives, it still misclassifies a significant number of rejected loans as approved.
- **Not Easily Interpretable** – Unlike Random Forest, which provides feature importance, SVM models are often treated as black boxes.

3. **XGBoost – Great but Not the Best**

- **Slightly Lower Accuracy than Random Forest (90.29%)** – While XGBoost is a powerful model, Random
  Forest still outperforms it in overall classification.
- **Higher False Negatives (501 cases)** – Compared to Random Forest (407 false negatives), XGBoost is more likely to **miss actual loan approvals**, potentially leading to **lost business opportunities**.
- **More Complex & Requires Tuning** – XGBoost requires careful hyperparameter tuning to perform well, making it harder to deploy without expert intervention.

4. **LightGBM – Best AUC but Not Best for Classification**

- **Lower Accuracy than Random Forest (90.07%)** – Despite having the highest AUC (96.74%), its overall classification accuracy is **still slightly lower** than Random Forest.
- **Prone to Overfitting** – LightGBM can be sensitive to noisy data, leading to overfitting if not properly tuned.
- **Not as Interpretable as Random Forest** – While LightGBM is powerful, it lacks the intuitive feature importance insights that Random Forest provides.



The Receiver Operating Characteristic (ROC) curve provides a graphical representation of the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) for various classification models. The closer the curve follows the top-left corner, the better the model's performance in distinguishing between classes. The Area Under the Curve (AUC) quantifies this performance, with a higher value indicating a stronger classifier.

From the ROC curve above, the following observations can be made:

- Logistic Regression achieved an AUC of 0.95, indicating good performance.
- SVM performed slightly better with an AUC of 0.96.
- Random Forest, XGBoost, and LightGBM all achieved the highest AUC of 0.97, suggesting superior classification capabilities.
- The Random Classifier (dashed line) represents a baseline model with an AUC of 0.50, which is equivalent to random guessing

12. Conclusion:

Based on the ROC curve analysis and the provided model performance metrics, the Random Forest Classifier emerges as the best overall model for this classification task. It offers the highest accuracy (91.06%), a balanced precision-recall trade-off (~91%), and lower false positives (824 cases) compared to other models.

12.1 **Alternative Considerations:**

- **LightGBM and XGBoost** also perform exceptionally well with **AUC = 0.97** and strong recall, making them ideal for scenarios where ranking the risk of loans is more important than pure classification.
- **SVM** has the lowest false negatives, making it a good choice if minimizing missed approvals is crucial.
- **Logistic Regression**, while interpretable, falls short in accuracy and false positive reduction.

13. Final Recommendation:

For optimal loan classification performance, **Random Forest Classifier is the best choice** due to its **highest accuracy, balanced precision-recall, and strong AUC score**. However, **XGBoost and LightGBM are excellent alternatives**, especially when computational efficiency and risk ranking are priorities.