Shyam J Devani - SJD210003
Rudy Rajput - RXR210083

1. Project and Dataset Selection

For this project, we selected the Employer Data (Loan Approval and Risk Assessment) Kaggle dataset (https://www.kaggle.com/datasets/gmudit/employer-data). The dataset contains 20,000 rows and 36 features for demographic, employment, and financial information of loan applicants. It offers a classification problem (learning if a loan is approved) and regression problem (learning the applicant's risk score).

For this assignment, we applied the issue of classification in which the target variable is LoanApproved (0 = Not Approved, 1 = Approved). The predictors include factors such as Age, AnnualIncome, CreditScore, LoanAmount, DebtToIncomeRatio, etc., related to financial stability.

We intend to build and compare a number of tree-based models for loan approval outcome prediction and evaluate their performance based on metrics like Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

2. Tree Model Building

2.1 Pre-Processing

The data was first loaded into a pandas DataFrame from a public URL.
Summary of data showed no missing values and no inconsistency in the 36 columns. Categorical features such as EmploymentStatus, EducationLevel, MaritalStatus, HomeOwnershipStatus, and LoanPurpose were one-hot encoded.

The quantitative features were examined for distribution and correlation. Several key variables such as TotalDebtToIncomeRatio, AnnualIncome, and MonthlyIncome had right-skewed distributions since it would be true for most financial information. Features were normalized for comparison between models.

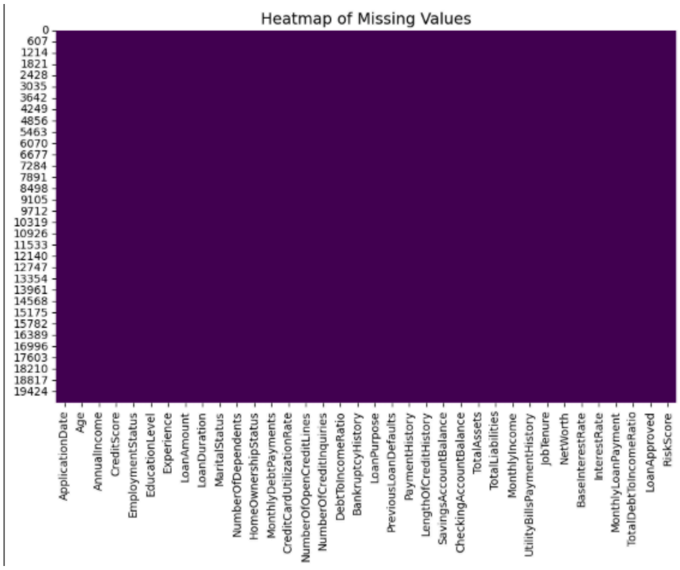
Correlation analysis indicated that DebtToIncomeRatio, CreditScore, and InterestRate were most correlated with loan approval, and they offered useful indications to learn from a model.

	ApplicationDate	Age	AnnualIncome	CreditScore	EmploymentStatus	EducationLevel	Experience	LoanAmount	LoanDuration	MaritalStatus	...
0	2018-01-01	45	39948	617	Employed	Master	22	13152	48	Married	...
1	2018-01-02	38	39709	628	Employed	Associate	15	26045	48	Single	...
2	2018-01-03	47	40724	570	Employed	Bachelor	26	17627	36	Married	...
3	2018-01-04	58	69084	545	Employed	High School	34	37898	96	Single	...
4	2018-01-05	37	103264	594	Employed	Associate	17	9184	36	Married	...

	MonthlyIncome	UtilityBillsPaymentHistory	JobTenure	NetWorth	BaseInterestRate	InterestRate	MonthlyLoanPayment	TotalDebtToIncomeRatio	LoanApproved	RiskScore
0	3329.000000	0.724972	11	126928	0.199652	0.227590	419.805992	0.181077	0	49.0
1	3309.083333	0.935132	3	43609	0.207045	0.201077	794.054238	0.389852	0	52.0
2	3393.666667	0.872241	6	5205	0.217627	0.212548	666.406688	0.462157	0	52.0
3	5757.000000	0.896155	5	99452	0.300398	0.300911	1047.506980	0.313098	0	54.0
4	8605.333333	0.941369	5	227019	0.197184	0.175990	330.179140	0.070210	1	36.0

5 rows × 36 columns

```
(20000, 36)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 36 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ApplicationDate             20000 non-null  object
 1   Age                         20000 non-null  int64
 2   AnnualIncome                20000 non-null  int64
 3   CreditScore                 20000 non-null  int64
 4   EmploymentStatus            20000 non-null  object
 5   EducationLevel              20000 non-null  object
 6   Experience                  20000 non-null  int64
 7   LoanAmount                  20000 non-null  int64
 8   LoanDuration                20000 non-null  int64
 9   MaritalStatus               20000 non-null  object
 10  NumberOfDependents          20000 non-null  int64
 11  HomeOwnershipStatus         20000 non-null  object
 12  MonthlyDebtPayments         20000 non-null  int64
 13  CreditCardUtilizationRate   20000 non-null  float64
 14  NumberOfOpenCreditLines     20000 non-null  int64
 15  NumberOfCreditInquiries     20000 non-null  int64
 16  DebtToIncomeRatio           20000 non-null  float64
 17  BankruptcyHistory           20000 non-null  int64
 18  LoanPurpose                 20000 non-null  object
 19  PreviousLoanDefaults        20000 non-null  int64
 20  PaymentHistory              20000 non-null  int64
 21  LengthOfCreditHistory       20000 non-null  int64
 22  SavingsAccountBalance       20000 non-null  int64
 23  CheckingAccountBalance      20000 non-null  int64
 24  TotalAssets                 20000 non-null  int64
 25  TotalLiabilities            20000 non-null  int64
 26  MonthlyIncome               20000 non-null  float64
 27  UtilityBillsPaymentHistory  20000 non-null  float64
 28  JobTenure                   20000 non-null  int64
 29  NetWorth                    20000 non-null  int64
 30  BaseInterestRate            20000 non-null  float64
 31  InterestRate                20000 non-null  float64
 32  MonthlyLoanPayment          20000 non-null  float64
 33  TotalDebtToIncomeRatio      20000 non-null  float64
 34  LoanApproved                20000 non-null  int64
 35  RiskScore                   20000 non-null  float64
dtypes: float64(9), int64(21), object(6)
memory usage: 5.5+ MB
None
```



2.2 Model Construction

Four tree-based classifiers were trained and tuned using GridSearchCV with cross-validation:

1. **Decision Tree Classifier**

```
Confusion Matrix:
 [[4245  321]
 [ 349 1085]]

Classification Report:
              precision    recall  f1-score   support

           0     0.9240    0.9297    0.9269      4566
           1     0.7717    0.7566    0.7641      1434

    accuracy                         0.8883      6000
   macro avg     0.8479    0.8432    0.8455      6000
weighted avg     0.8876    0.8883    0.8880      6000

ROC AUC: 0.8431612958798615
```

2. **Random Forest Classifier**

```
Fitting 3 folds for each of 20 candidates, totalling 60 fits
RF Best Params: {'n_estimators': 300, 'min_samples_split': 5, 'min_samples_leaf': 2, 'max_features': None, 'max_depth': None, 'criterion': 'gini', 'bootstrap': True}

Confusion Matrix:
 [[4370  196]
 [ 210 1224]]

Classification Report:
              precision    recall  f1-score   support

           0     0.9541    0.9571    0.9556      4566
           1     0.8620    0.8536    0.8577      1434

    accuracy                         0.9323      6000
   macro avg     0.9081    0.9053    0.9067      6000
weighted avg     0.9321    0.9323    0.9322      6000

ROC AUC: 0.9785593107994266
```

3. **AdaBoost Classifier**

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Ada Best Params: {'learning_rate': 0.5, 'n_estimators': 300}

Confusion Matrix:
 [[4454  112]
 [ 190 1244]]

Classification Report:
              precision    recall  f1-score   support

           0     0.9591    0.9755    0.9672      4566
           1     0.9174    0.8675    0.8918      1434

    accuracy                         0.9497      6000
   macro avg     0.9382    0.9215    0.9295      6000
weighted avg     0.9491    0.9497    0.9492      6000

ROC AUC: 0.9888642388010099
```

4. **XGBoost Classifier**

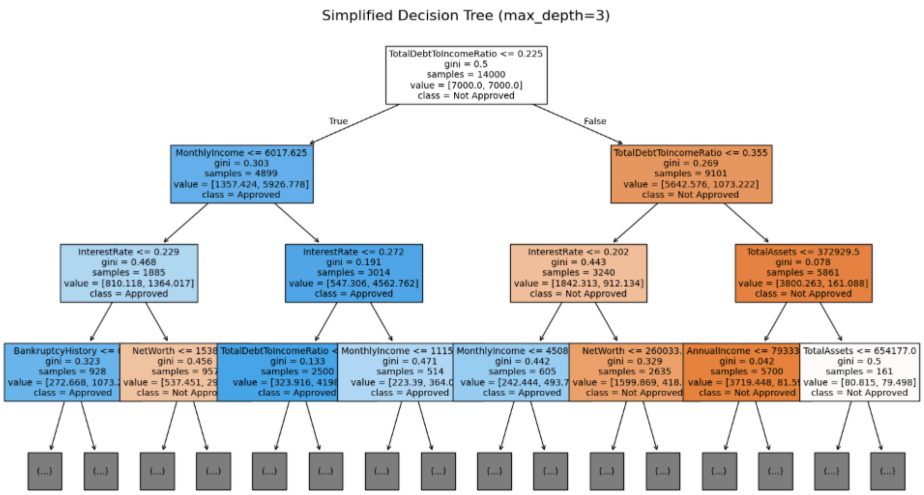
```
Fitting 3 folds for each of 20 candidates, totalling 60 fits
XGB Best Params: {'subsample': 0.9, 'min_child_weight': 5, 'max_depth': 3, 'learning_rate': 0.1, 'gamma': 1.0, 'colsample_bytree': 1.0}

Confusion Matrix:
 [[4351  215]
 [  65 1369]]

Classification Report:
              precision    recall  f1-score   support

           0     0.9853    0.9529    0.9688      4566
           1     0.8643    0.9547    0.9072      1434

    accuracy                         0.9533      6000
   macro avg     0.9248    0.9538    0.9380      6000
weighted avg     0.9564    0.9533    0.9541      6000

ROC AUC: 0.9926133736043071
```

All models were evaluated on a train-test split (70-30) with balanced class weighting to address the slight imbalance in approval outcomes (76% Not Approved, 24% Approved). Hyperparameters such as max_depth, n_estimators, learning_rate, and min_samples_leaf were optimized for best performance.

2.3 Tree Visualization

To understand how each model makes its predictions, decision trees were visualized for all four classifiers — Decision Tree, Random Forest, AdaBoost, and XGBoost.

The Decision Tree Classifier produced the most interpretable single model. The root node split was on TotalDebtToIncomeRatio, meaning that candidates with lower debt relative to income were more likely to be approved. Splits followed on MonthlyIncome, InterestRate, and NetWorth, reflecting significant lending reasoning, where higher income and lower interest rates increase approval probability. The lone tree readily shows how the model separates approved from not-approved candidates.

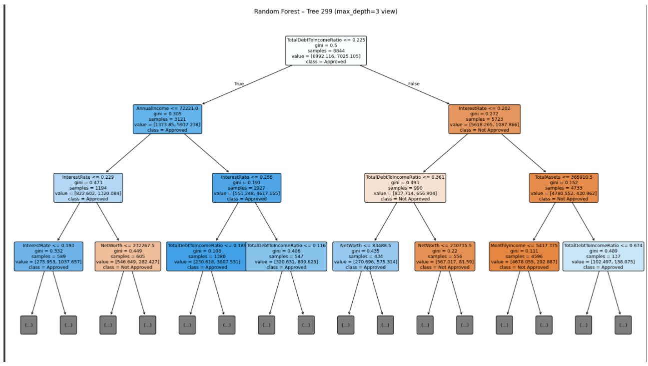


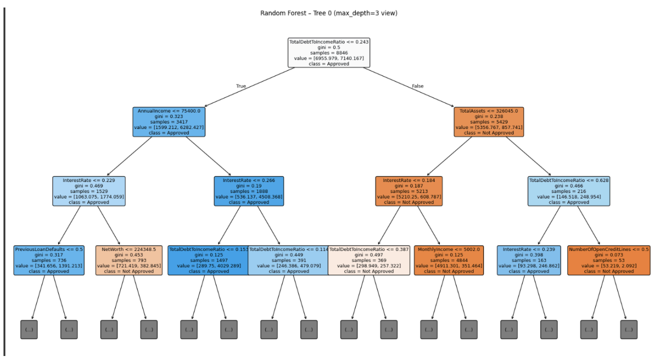
For the Random Forest Classifier, three representative trees (Tree 0, Tree 150, and Tree 299) were plotted, each limited to a maximum depth of 3 for readability.

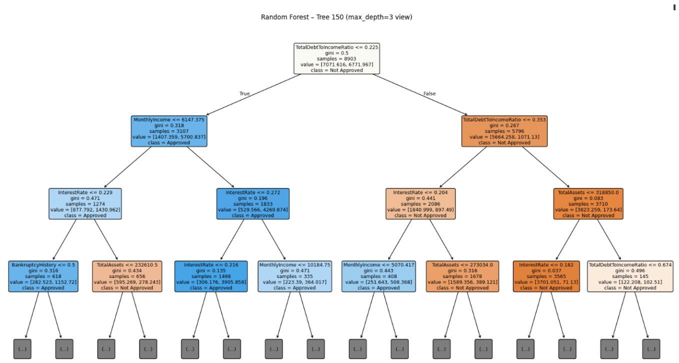
- All trees began with TotalDebtToIncomeRatio as the dominant root split, confirming its importance across the ensemble.

- Secondary splits frequently appeared on AnnualIncome, MonthlyIncome, and TotalAssets, identifying high-earning and asset-rich applicants as lower-risk.

- Additional splits on InterestRate, NetWorth, and PreviousLoanDefaults refined subgroups of borderline applicants.
  These trees demonstrate how Random Forest combines many shallow, diverse learners to reduce variance and achieve stable accuracy.





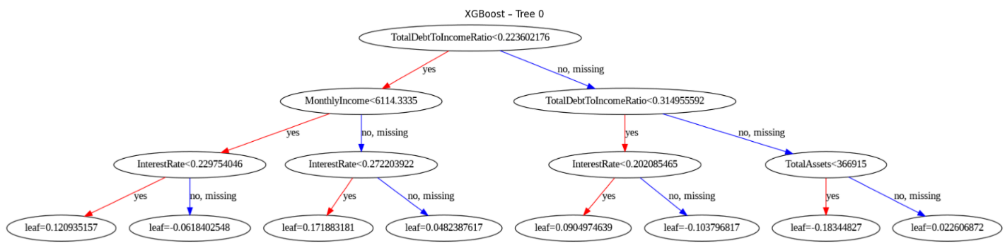


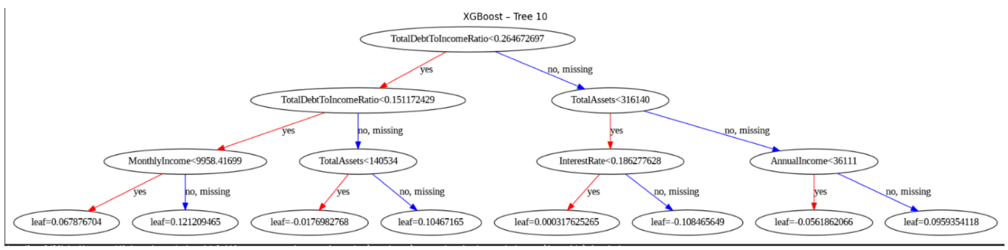
For the XGBoost Classifier, three trees (Tree 0, Tree 10, and Tree 399) were visualized to highlight the gradient-boosting process.

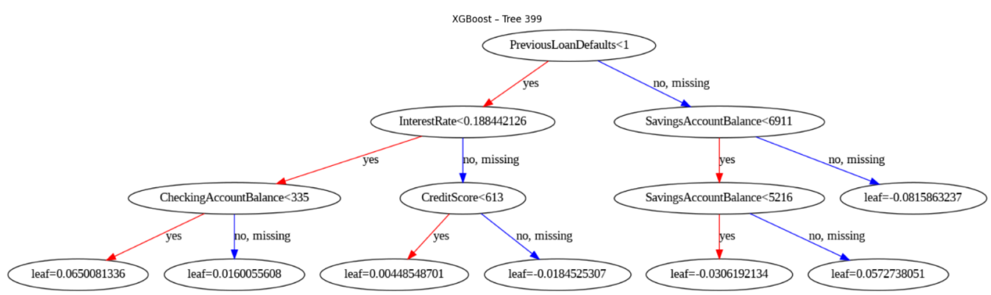
- Tree 0 began with TotalDebtToIncomeRatio, followed by MonthlyIncome, InterestRate, and TotalAssets, reflecting the model's foundational decision pattern.

- Tree 10 refined these patterns with new splits on TotalAssets and AnnualIncome, capturing subtle income-to-debt relationships.

- Tree 399 focused on PreviousLoanDefaults, CreditScore, and SavingsAccountBalance, showing how later iterations fine-tune edge cases to optimize performance.
  Together, these trees illustrate XGBoost's ability to learn from residual errors and incrementally build a highly accurate ensemble.





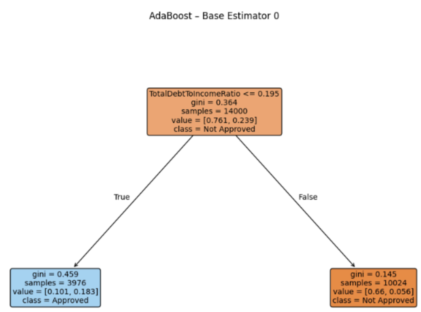


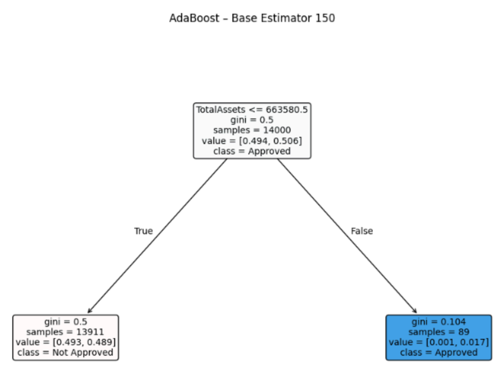
The AdaBoost Classifier visualizations (Base Estimators 0, 150, and 299) illustrate the model's sequential learning behavior.

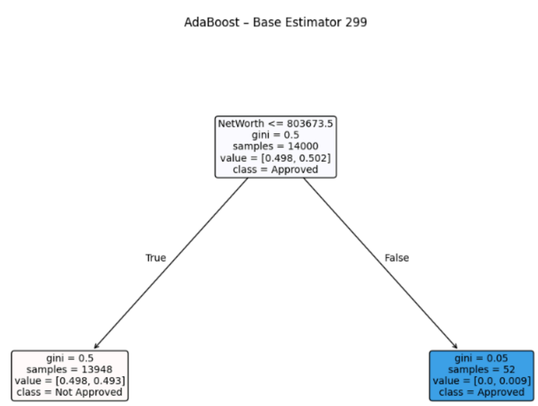
- Base Estimator 0 relied almost entirely on TotalDebtToIncomeRatio, drawing a single clear boundary between approved and not-approved applicants.

- Base Estimator 150 introduced TotalAssets as an additional criterion, suggesting that asset value becomes a key correction factor as boosting continues.

- Base Estimator 299 emphasized NetWorth, fine-tuning the model's focus on applicants with substantial assets.
  These incremental changes reflect AdaBoost's mechanism — each successive tree corrects the mistakes of earlier ones, gradually improving predictive accuracy through weighted re-training.







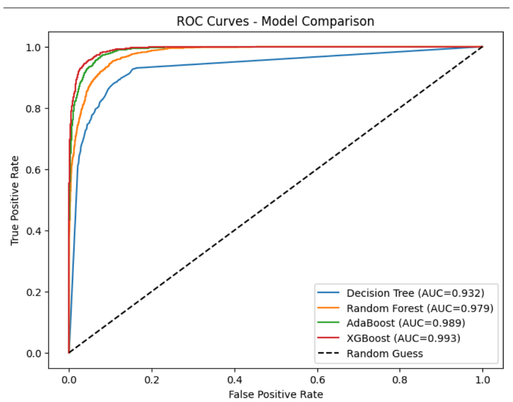
Across all models, the most consistent and influential predictors were TotalDebtToIncomeRatio, Income (Monthly and Annual), InterestRate, and TotalAssets. The recurrence of these variables across different algorithms confirms their strong predictive value in determining loan-approval outcomes.

2.4 Result Analysis

Model performance was evaluated on the test set using Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics.

The XGBoost Classifier achieved the best results with the highest AUC (0.993) and a well-balanced precision-recall tradeoff. Both XGBoost and AdaBoost outperformed the Random Forest and Decision Tree, demonstrating that boosting methods handle bias and variance more effectively.

Feature importance plots further highlighted TotalDebtToIncomeRatio, AnnualIncome, and LoanDuration as the most significant predictors of loan approval. This aligns closely with the visualized trees, where these same attributes consistently appeared near the top splits.



	Accuracy	Precision	Recall	F1	ROC AUC
XGBoost	0.9533	0.8643	0.9547	0.9072	0.9926
AdaBoost	0.9497	0.9174	0.8675	0.8918	0.9889
Random Forest	0.9323	0.8620	0.8536	0.8577	0.9786
Decision Tree	0.8908	0.7250	0.8752	0.7930	0.9321

3. Conclusion

For all experiments, XGBoost was the top performer with enhanced generalization power and most stable performance across metrics. The results suggest that income-based characteristics and debt ratio are key drivers of the probability of loan approval. The project demonstrated the end-to-end pipeline of:

- Data cleaning and transformation,
- Model tuning with GridSearchCV,
- Comparative evaluation of ensemble methods, and
- Visual interpretation through ROC curves and decision trees.

Future improvements could include using SMOTE for better class balance, SHAP analysis for model explainability, and exploring the regression aspect of predicting RiskScore.