# Final Report

Credit Card Application Prediction ML Final Report

View on GitHub

# Credit Card Application Prediction: ML Final Report

## Introduction/Background

Applications for a credit card are typically rejected or accepted after considering the applicant's income, age, and history of financial responsibility. Given a large collection of successful and unsuccessful applicants, we develop models to predict whether a person's application will be accepted.

## Literature Review

- Sun and Vasarheyli suggested that algorithms more sophisticated than logistic regression and Naive Bayes are needed to predict whether a credit card application will be accepted.[1]
- Contrary to this, Peela, Gupta, Rathod, Bose, and Sharma achieved an accuracy of 86% using logistic regression. However, they were predicting whether an applicant would default on their credit card, and not whether their application would be accepted.[2]
- Others took an alternative to a single-model approach, using an ensemble of base models.[3]

Manual review of large volumes of these applications is extremely time-consuming and prone to error.

## Problem Definition

The problem is to use logistic regression, Naive Bayes, Random Forest, XGBoost and possibly other machine learning algorithms to predict whether a given application for a credit card will be accepted. A sufficiently accurate algorithm could make the processing of such applications substantially more efficient.

# Motivations

## Criteria Optimization

Determining which characteristics of an application typically influence whether the applicant will be accepted.

## Improving Efficiency

Different credit card companies might value things differently. A machine-learning algorithm would be particularly useful, because each company could train the algorithm on its own dataset, so that it weights factors in alignment with the company.

# Methods

## Preprocessing Data

### Separating Columns

In this step, we are separating features into categorical and numerical features. For example:

- Categorical features: “ethnicity”, “citizen”
- Numerical features: “age”, “debt”

### Building the preprocessor

#### Numerical features (StandardScaler)

For numerical features, we are completing the process of standardization. This ensures:

- The model treats large numbers like “income” and smaller numbers like “age” similarly
- All values in a column are centered around 0 with similar spread
- Values are standardized by subtracting the mean and dividing by standard deviation using StandardScaler

#### Categorical Features (One-Hot Encoding)

- Each unique category is assigned its own column
- Example: If industry had “tech” and “healthcare”, each becomes a separate column

- A value of 1 is assigned to the appropriate column for each row

# ML Algorithms/Models Implemented

## Logistic Regression

- Chosen for binary classification
- Built to predict credit card application acceptance based on individual information

## Naive Bayes

- More sophisticated machine-learning algorithm for binary classification
- Assumes input variables are independent (potential limitation)

## Random Forest

- Ensemble learning method using multiple decision trees
- Features:
  - Bootstrap aggregating (bagging) for reduced overfitting
  - Random feature selection at each split
  - Majority voting for final predictions

## XGBoost

- Gradient boosting implementation optimized for performance
- Features:
  - Sequential tree building with gradient descent optimization
  - Built-in regularization to prevent overfitting
  - Efficient handling of missing values

# Results and Discussion

## Quantitative Results

### Logistic Regression

- Overall accuracy: 84.1%
- Precision: 85.2%

- Recall: 82.9%

### Naive Bayes

- Overall accuracy: 73.9%
- Precision: 76.6%
- Recall: 70%

### Random Forest

- Overall accuracy: 84.0%
- Precision: 84.0%
- Recall: 84.0%
- F1-score: 84.0%

### XGBoost

- Overall accuracy: 83.0%
- Precision: 83.0%
- Recall: 83.0%
- F1-score: 83.0%

## Analysis of Algorithm/Model

### Logistic Regression

- Achieved accuracy, precision, and recall rates all above 80%
- Suggests possible linear decision boundary between approved and non-approved applications
- Assumes linear relationship between independent variables and log-odds of acceptance

### Naive Bayes

- Performed worse than logistic regression
- Feature independence assumption likely invalid
- Potential for improvement with non-naive Bayes approach

### Random Forest

- Achieved comparable accuracy to Logistic Regression (84%)
- Well-balanced performance across all metrics

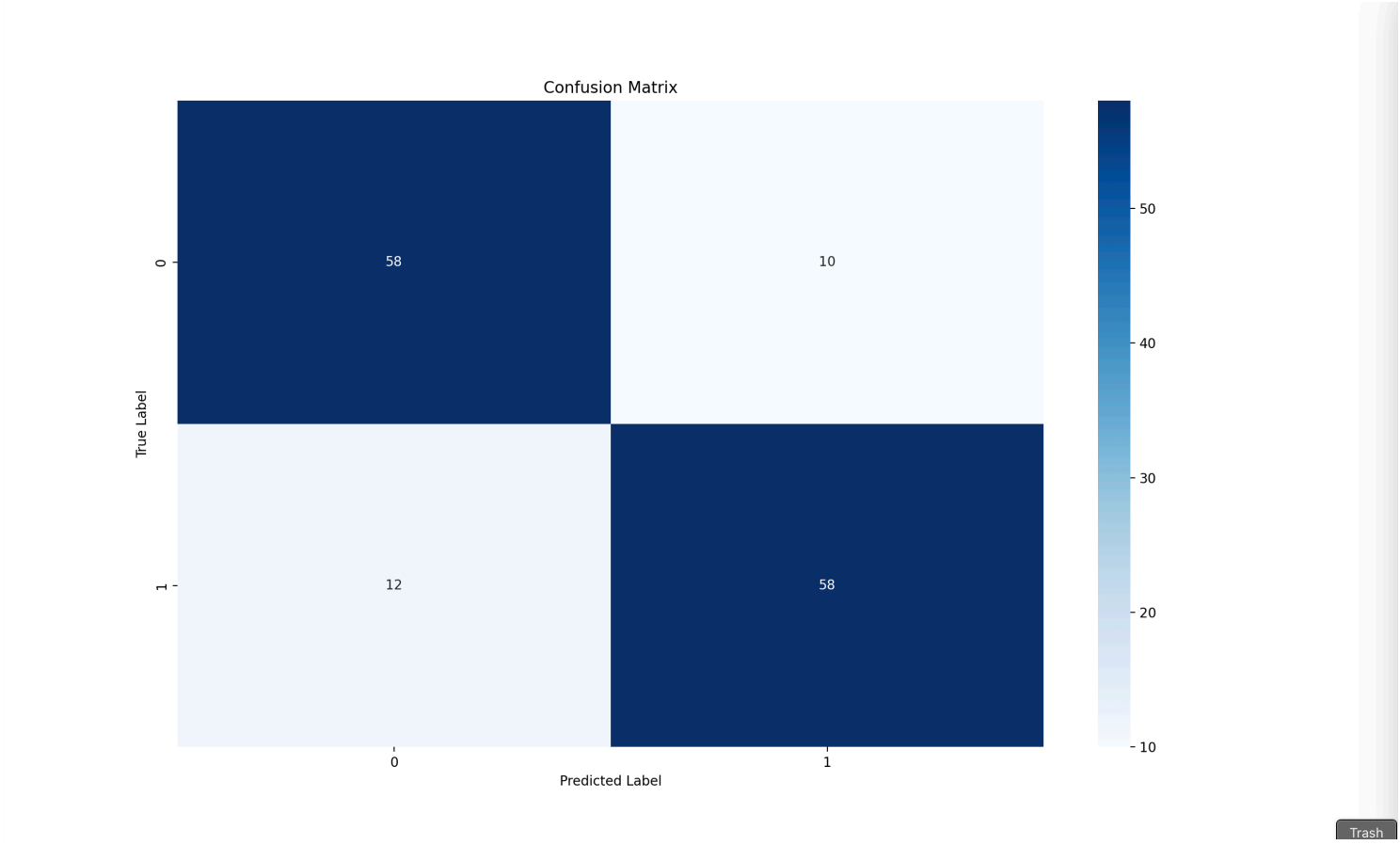
- Excellent feature importance insights
- Strong handling of non-linear relationships

### XGBoost

- Strong overall performance (83%)
- Very balanced metrics across classes
- Efficient handling of complex patterns
- Good at capturing feature interactions

## Visualizations Analysis

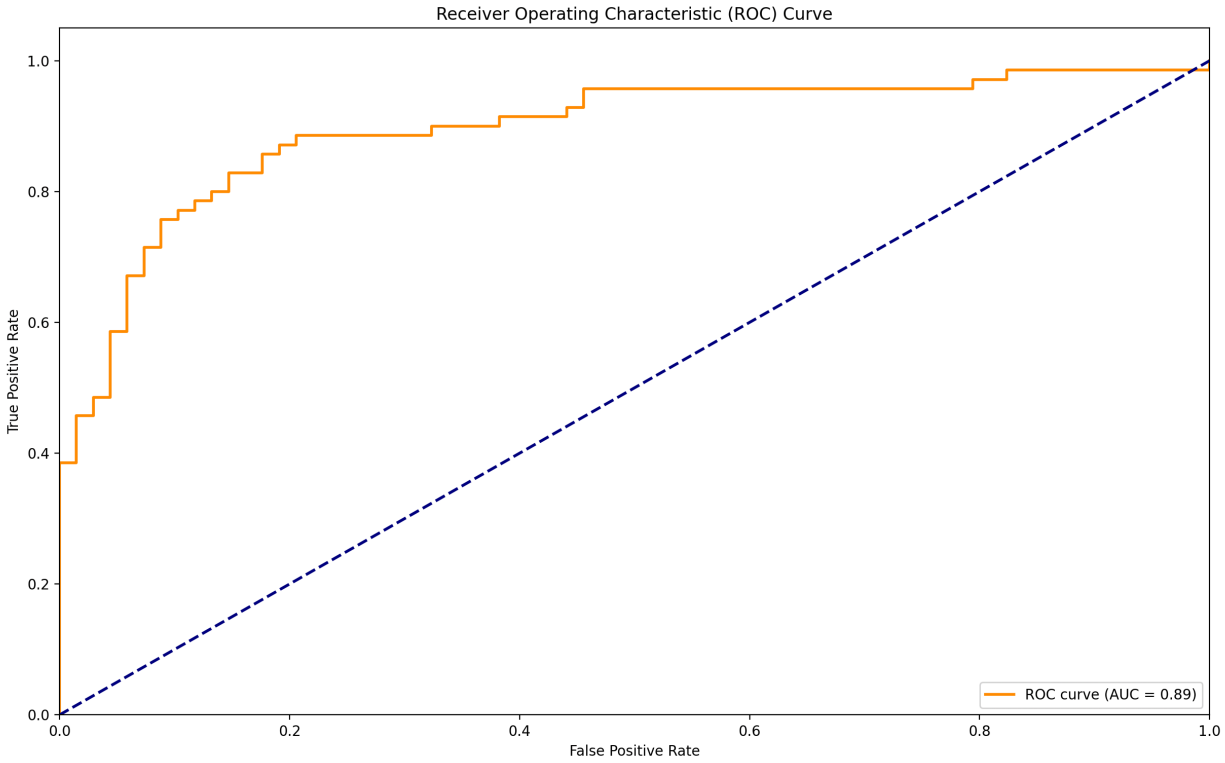
### Linear Regression Confusion Matrix




The confusion matrix provides a tabular summary of the model's classification performance. It shows the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). True Negatives (TN): The number of applicants correctly predicted as not approved for a loan False Positives (FP): The number of applicants incorrectly predicted as approved when they were actually not approved (Type I error) False Negatives (FN): The number of applicants incorrectly predicted as not approved when they were actually approved (Type II error) We see that there is a high TP and TN, indicating the model is doing a good job correctly classifying both the approved

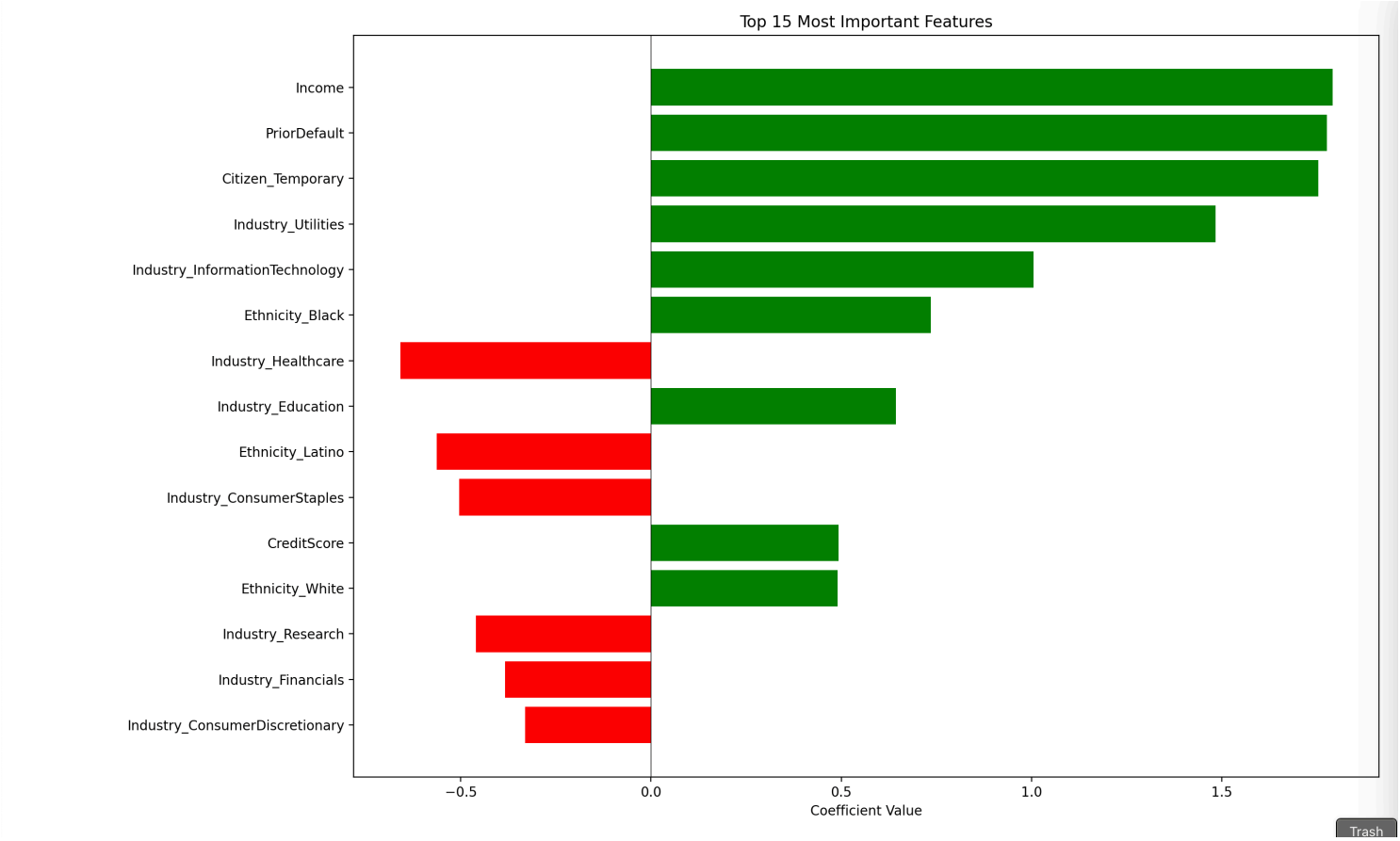
and not approved cases. Additionally, the low FP means the model is not approving cards for applicants who should not be approved. Ideally, we want to maximize TP and TN while minimizing FP and FN, which is evident in this image.

## Linear Regression ROC Curve



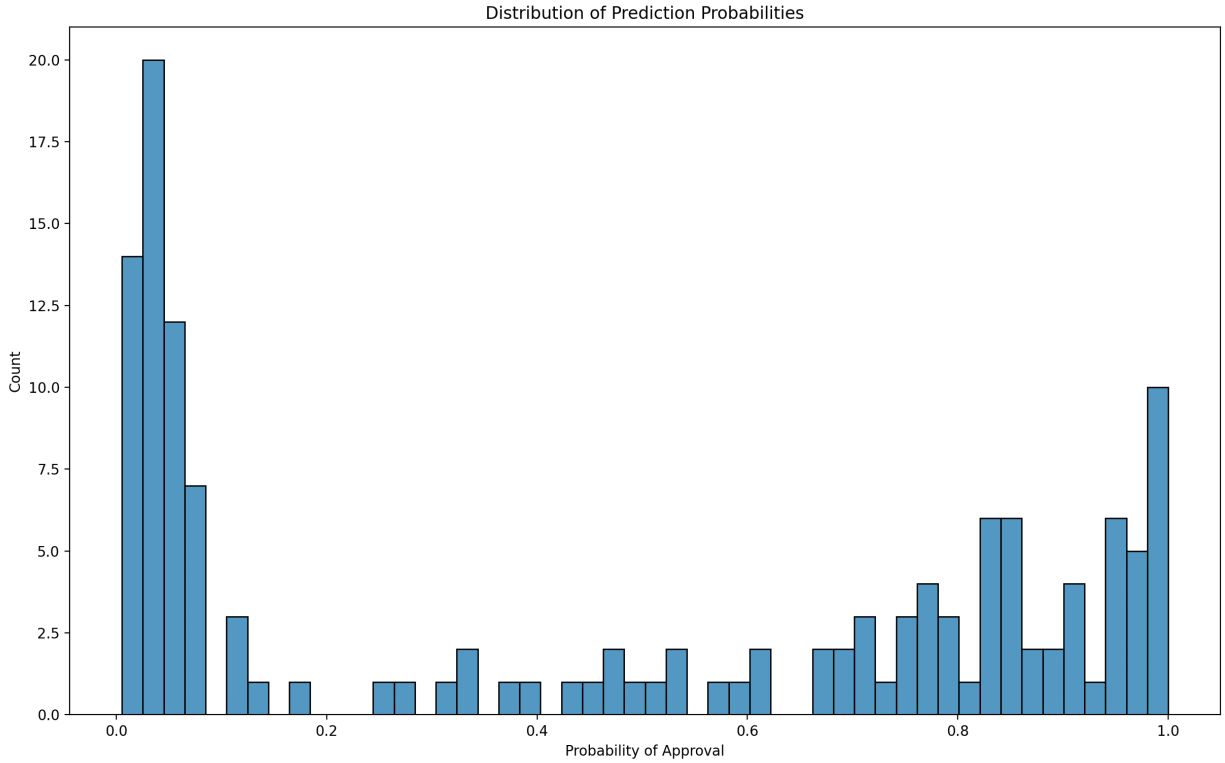

We see that the ROC curve is quite far from the dashed line, indicating that this model is doing a good job with classification. This means that most of the True Positives are captured while the false positives are minimized. The area under the curve (AUC) is also very high and reflects the good nature of the model.

## Linear Regression Feature Importance



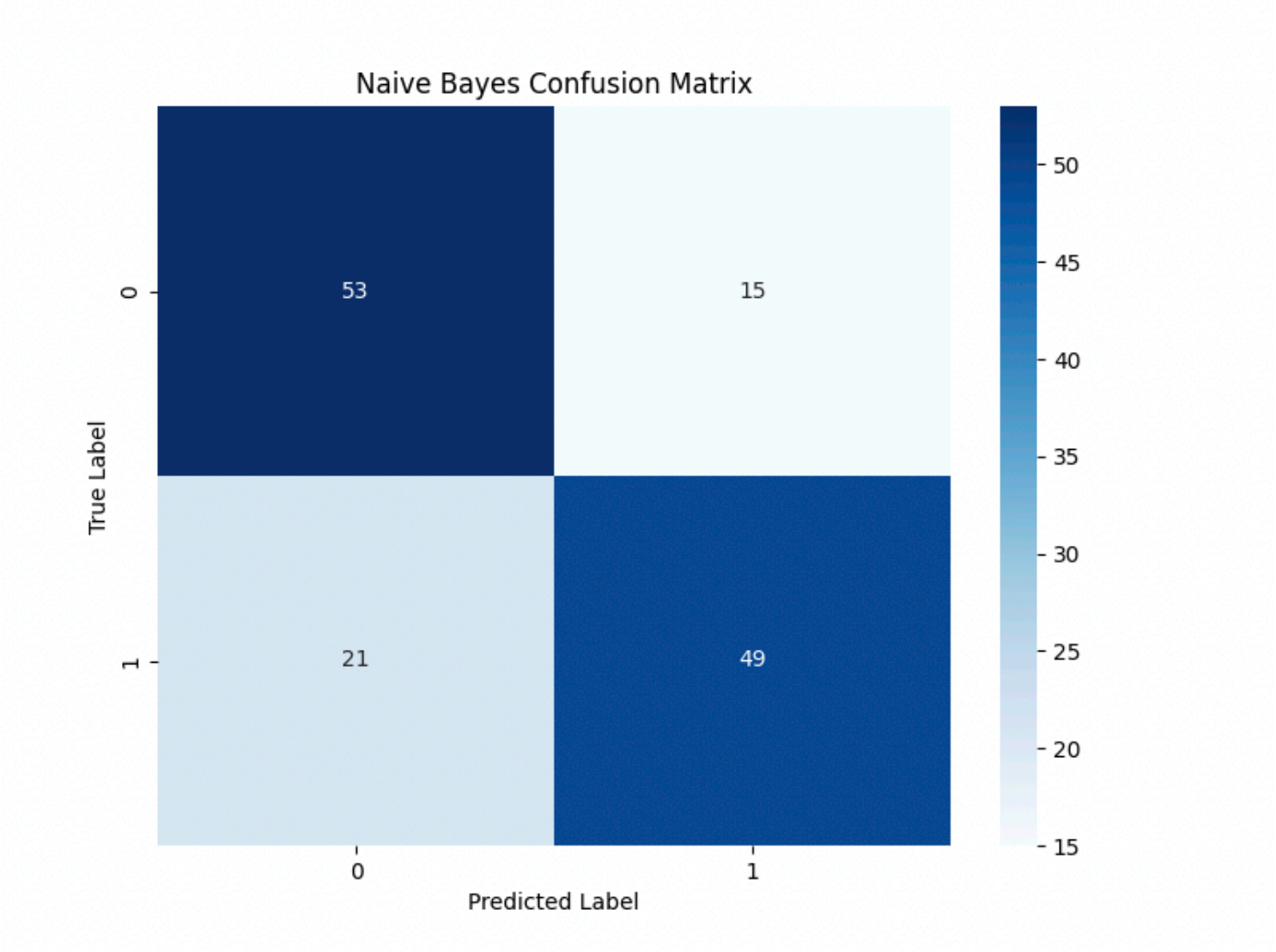

In this graph, we see a distribution of the features that have the greatest impact on the model's performance. For the green features, as they increase, the probability of credit card approval also increases. The opposite is true for the red features.

## Linear Regression Prediction Probabilities



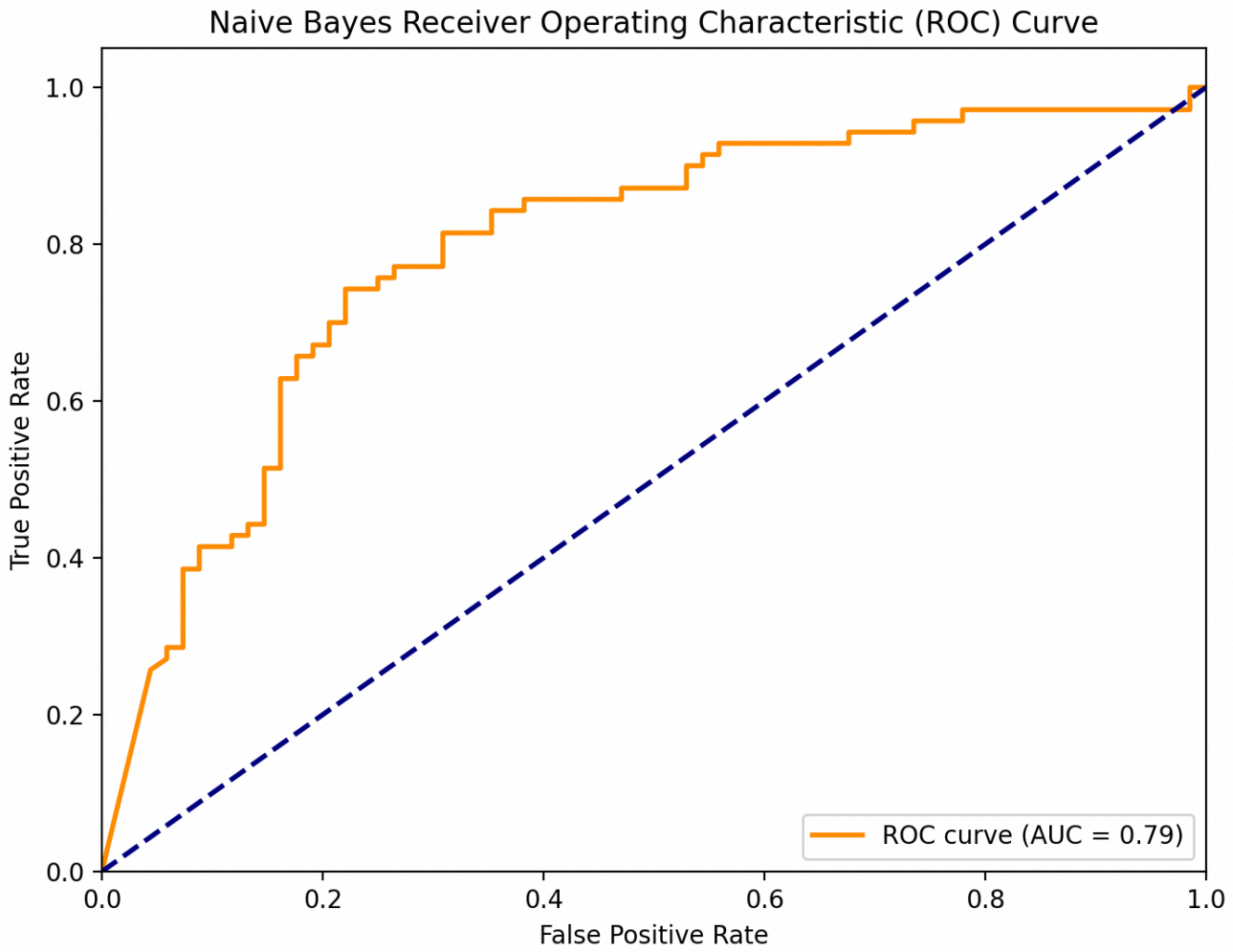

The prediction probability distribution is fairly bimodal with peaks near 0 and 1, suggesting that the model is confident in its predictions for approving/not approving applications.

## Naive-Bayes Confusion Matrix



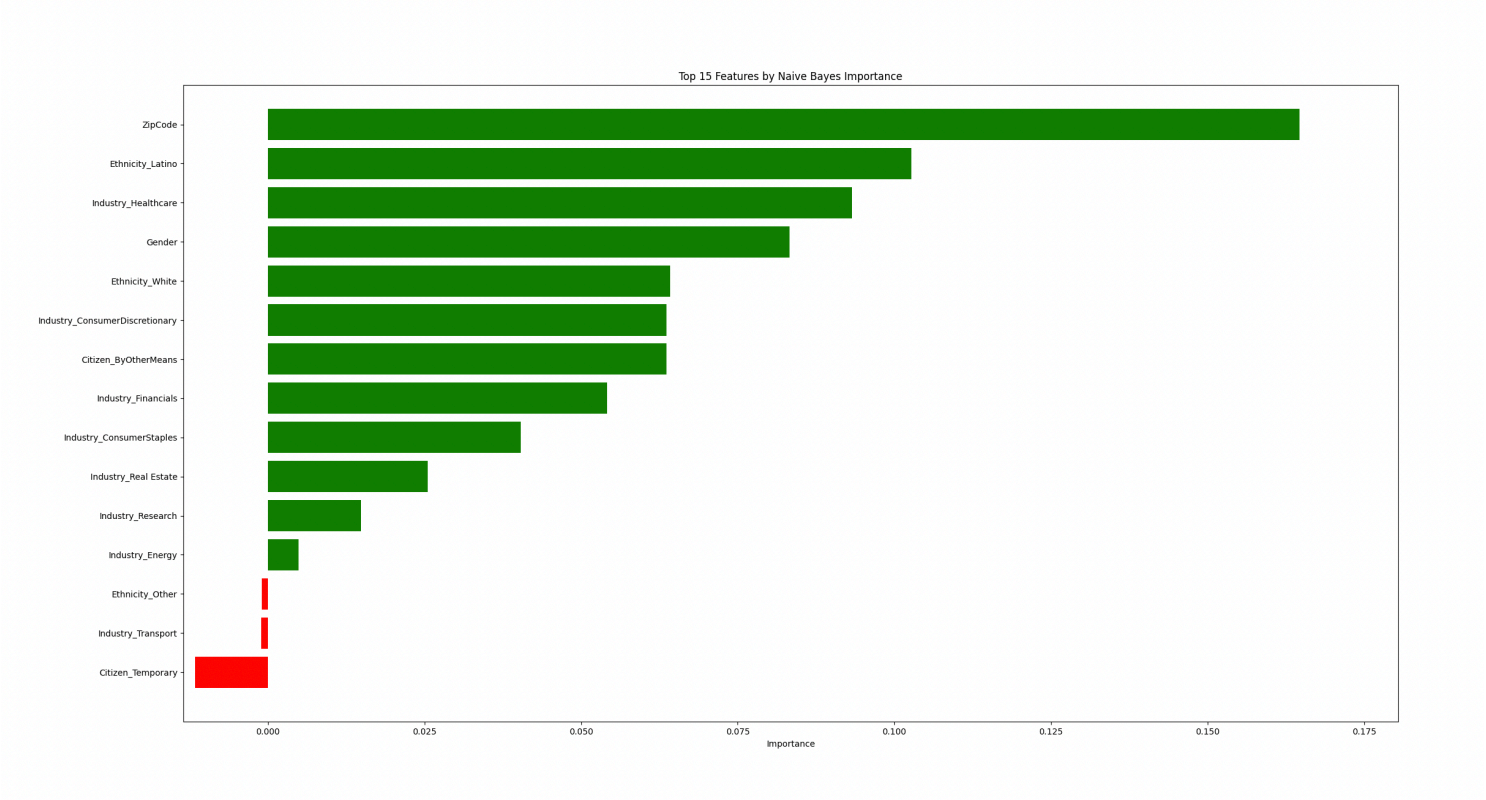

From this confusion matrix, we see that the number of True Positives (TP) in the top left corner and True Negatives (TN) in the bottom right are relatively high, while FP and FN are relatively minimized. This is our ideal output in this scenario, as we want to approve credit cards when they need to be approved and vice versa. These numbers are not as ideal as Logistic Regression.

## Naive-Bayes ROC Curve



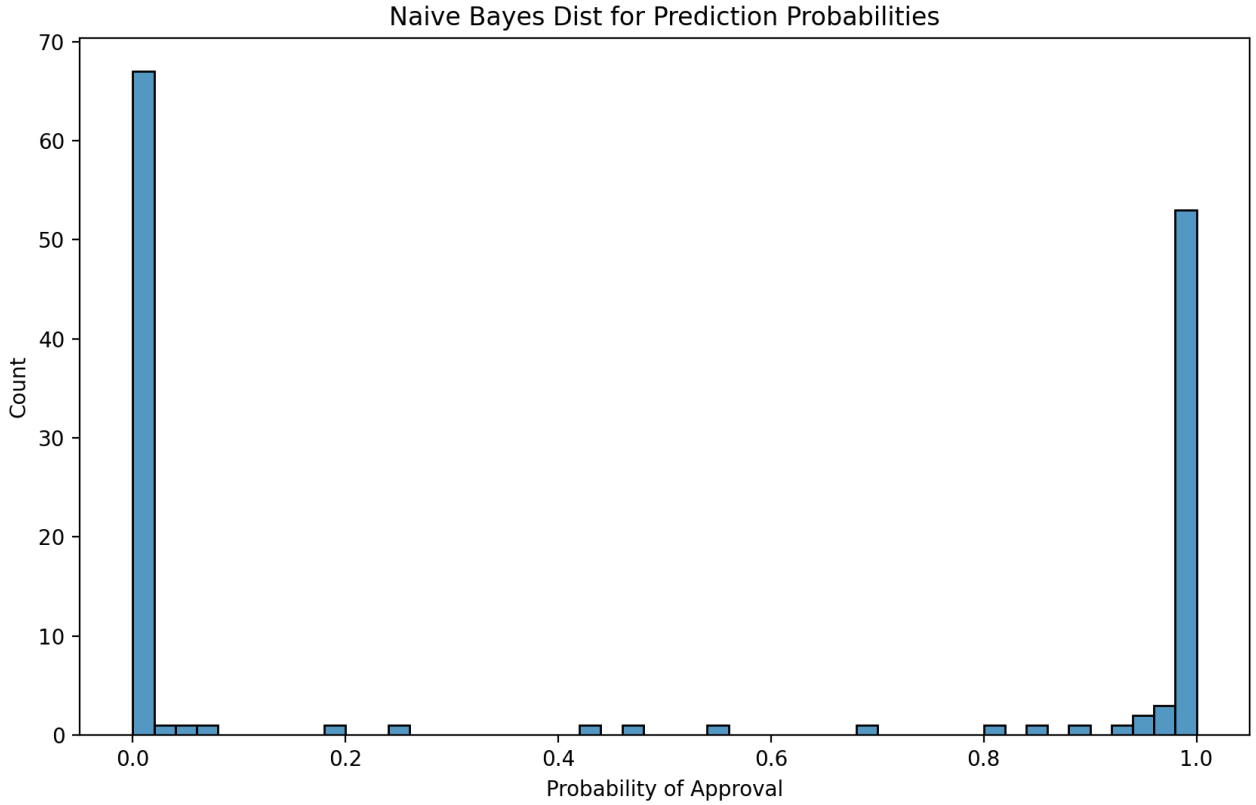

The ROC curve here is not as ideal as Logistic Regression, but it still gives promising results with an AUC of 0.79.

## Naive-Bayes Feature Importance



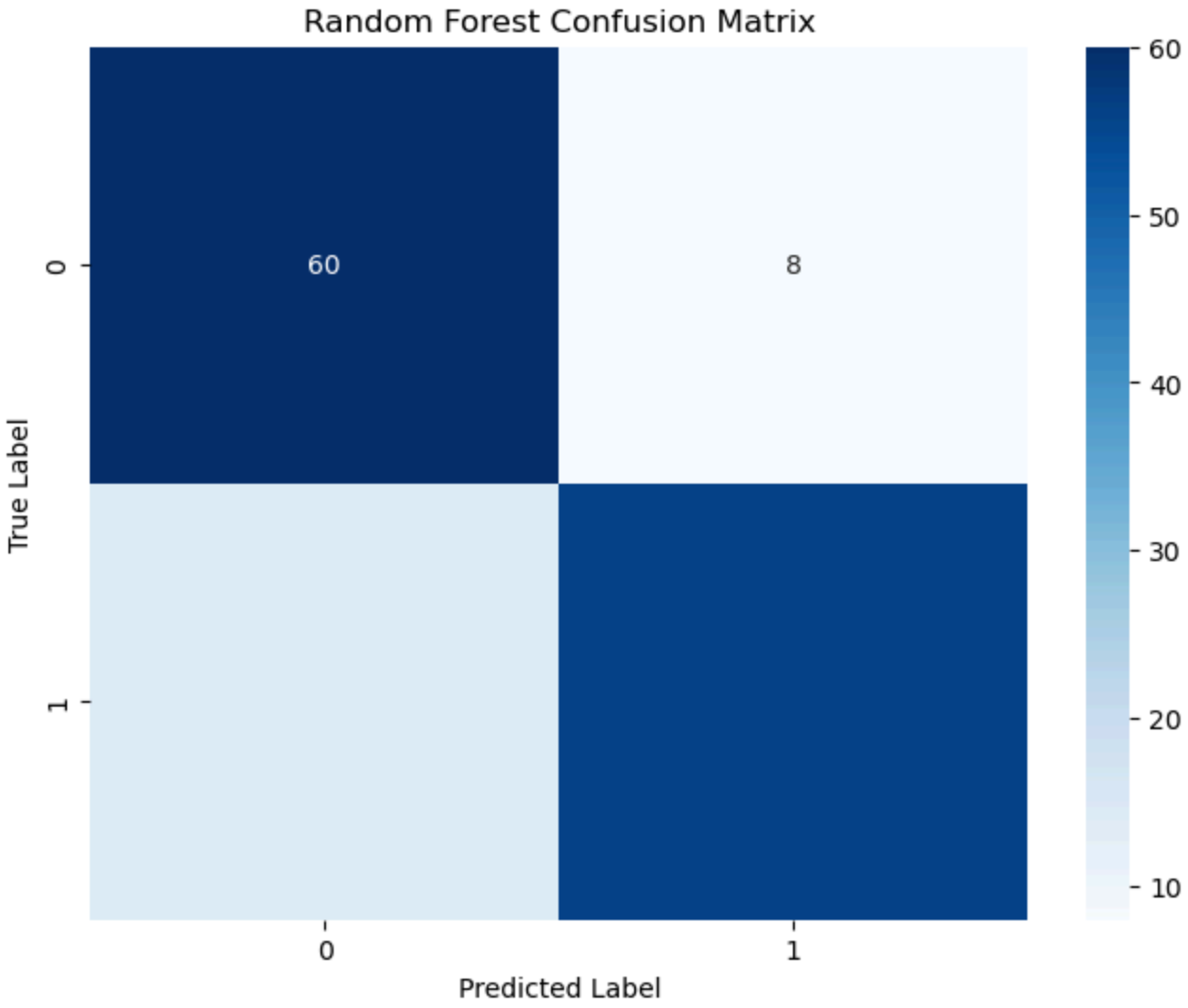

The Naive Bayes gives most importance to zipcode, ethnicity, and being in the healthcare industry, which is vastly different from the feature importance graph for logistic regression.

## Naive-Bayes Prediction Probabilities



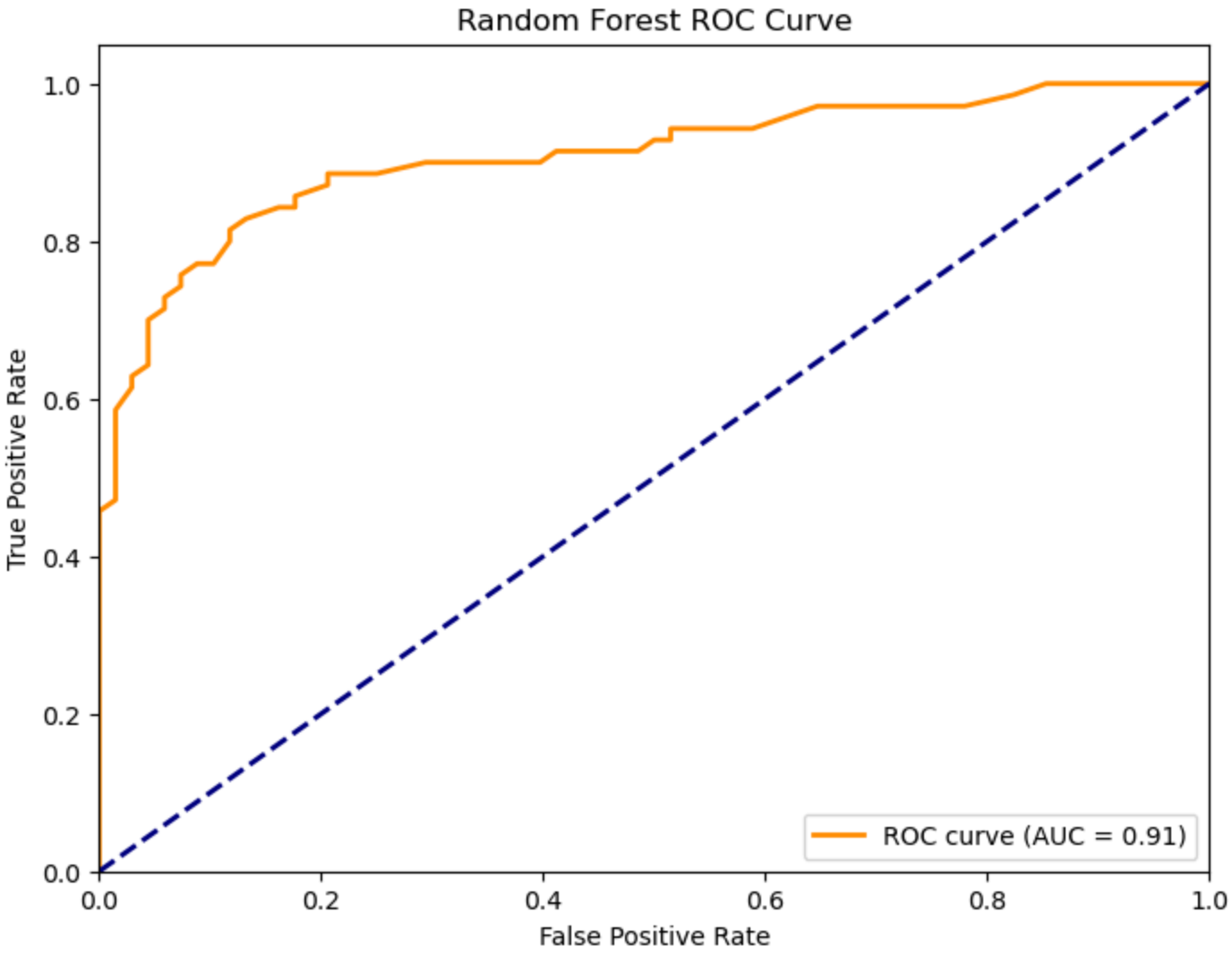

We see peaks at 0 and 1 with a very apparent bimodal appearance, so the model is extremely confident in its decisions to approve or disapprove of credit card applications. This is a good sign.

## Random Forest Confusion Matrix



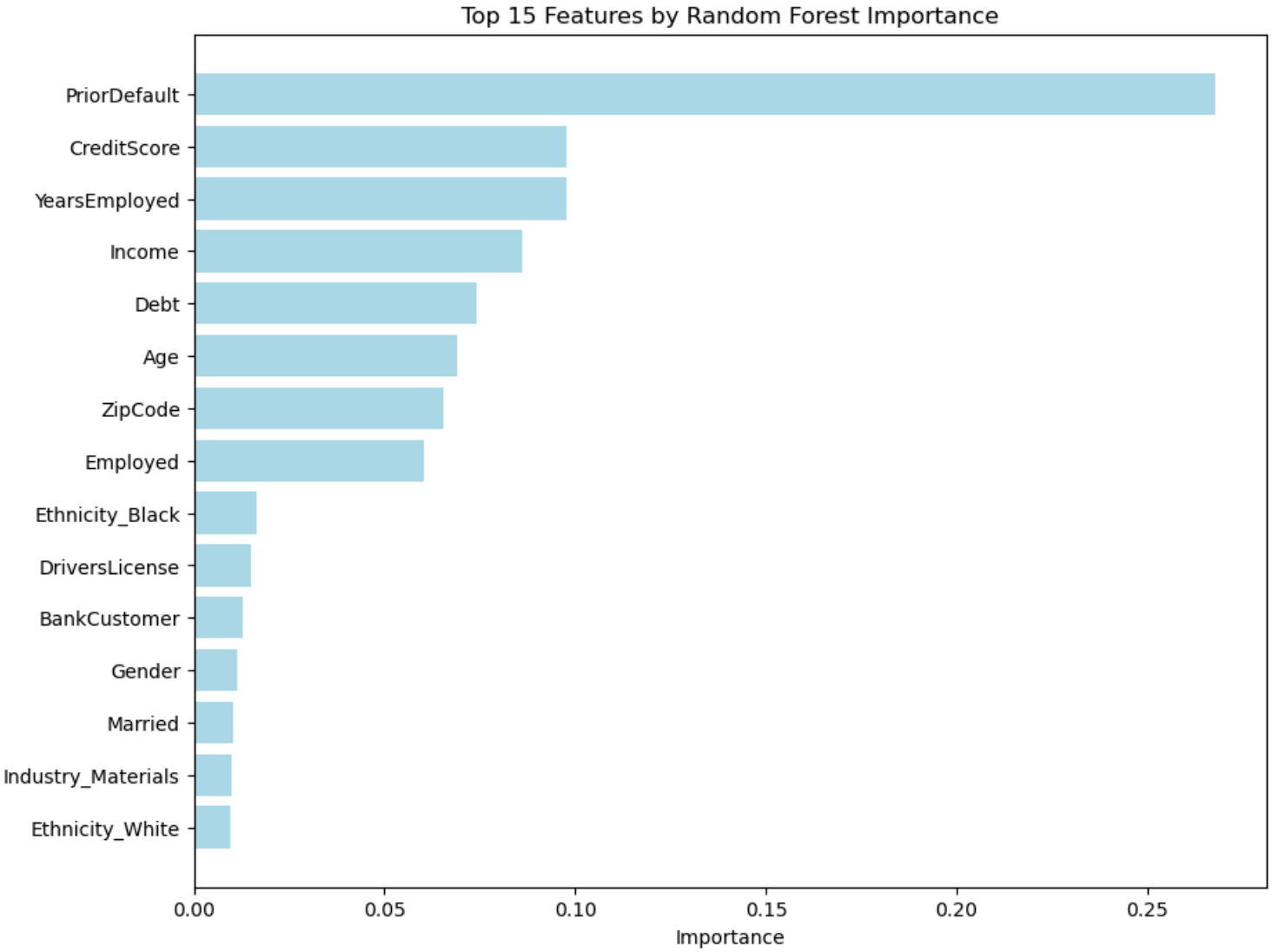

The confusion matrix demonstrates excellent performance with 60 true positives and very few misclassifications. This indicates that the Random Forest model is doing a strong job of correctly classifying both approved and rejected applications, with a good balance between false positives and false negatives.

## Random Forest ROC Curve



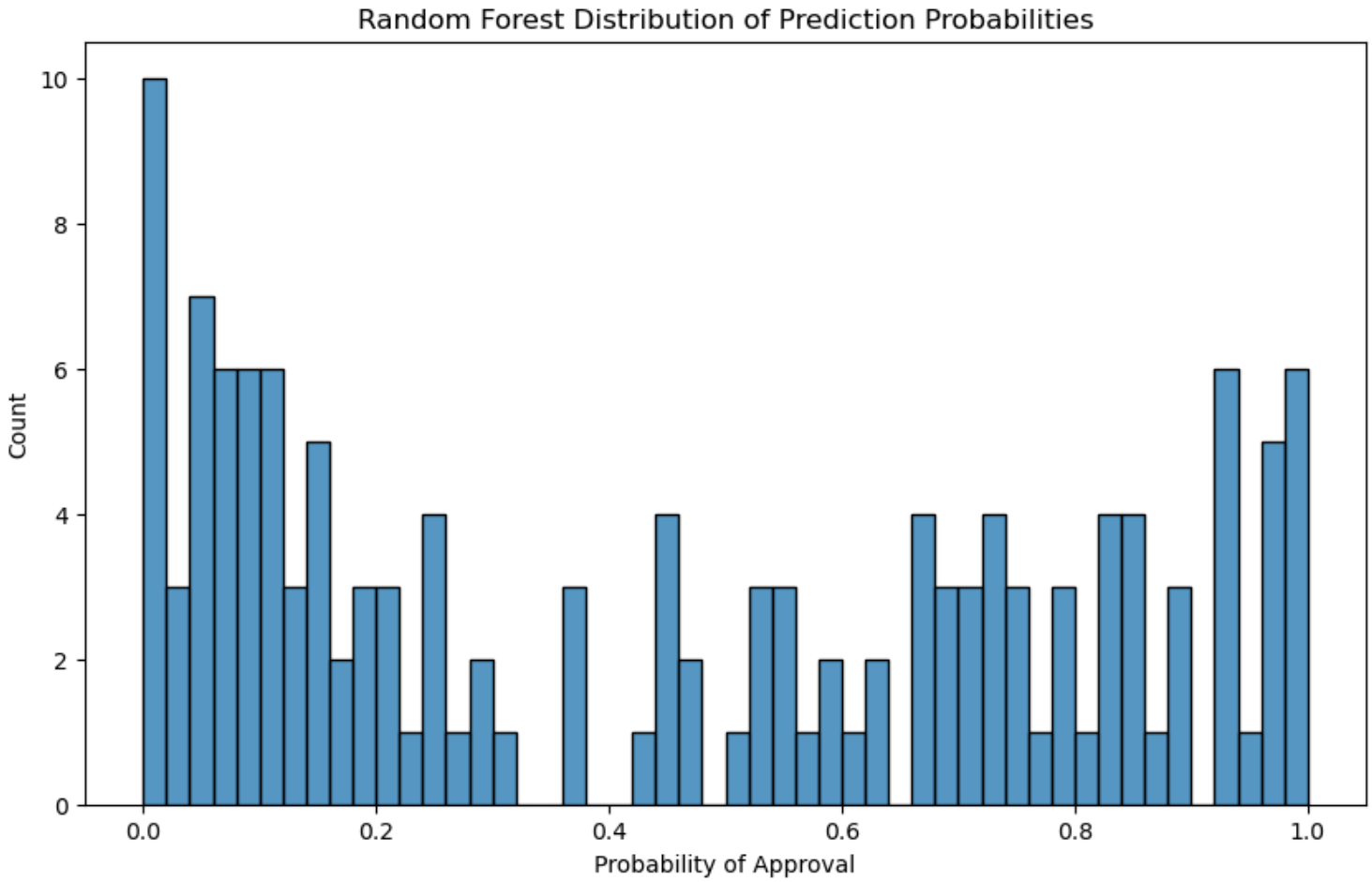

The Random Forest model achieves an impressive AUC of 0.91, indicating excellent discriminative ability. The curve shows consistently strong performance across different threshold values, significantly outperforming the random baseline and demonstrating robust predictive power.

## Random Forest Feature Importance



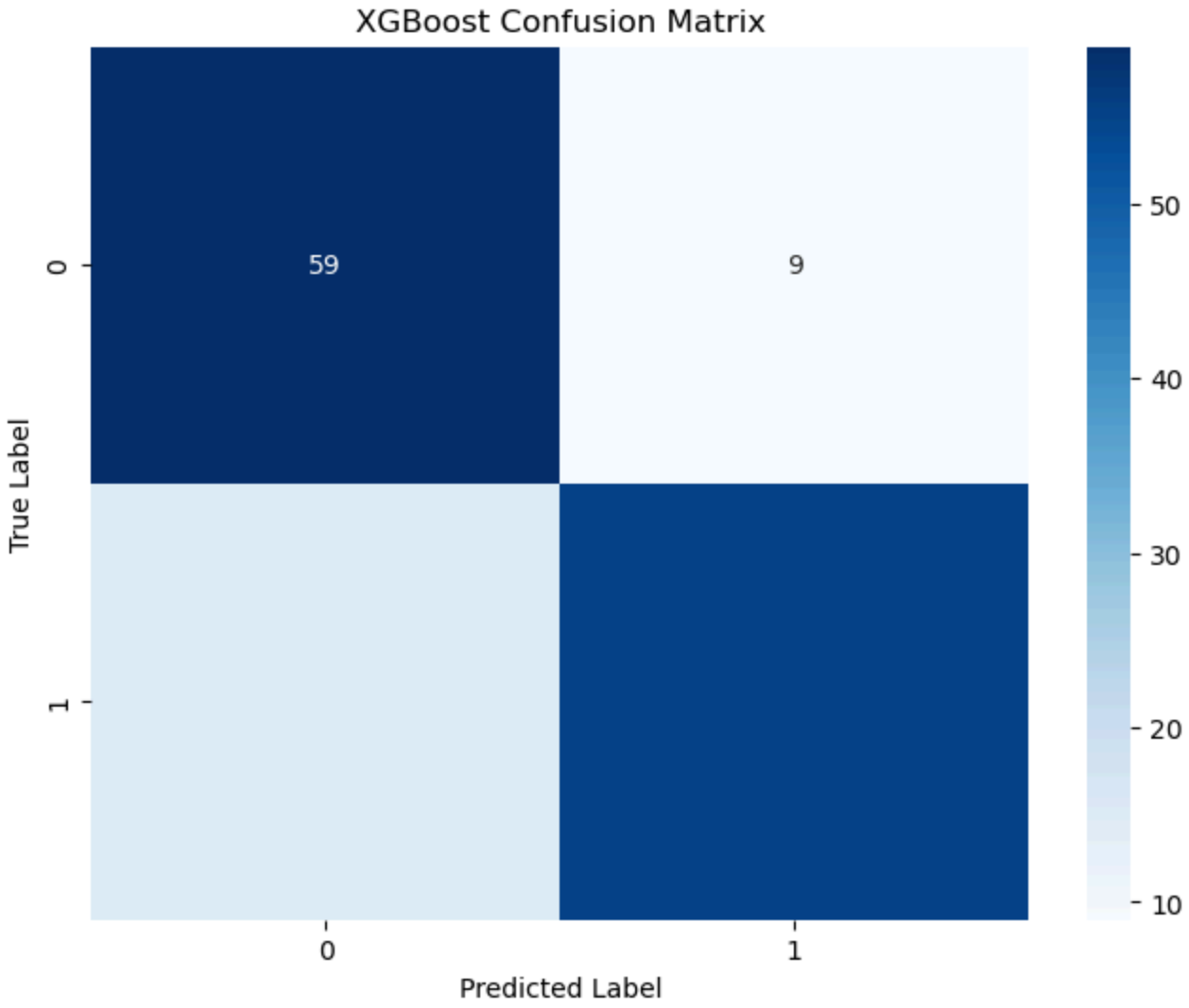

The feature importance graph shows PriorDefault as the most critical factor, followed by CreditScore and YearsEmployed. This aligns with intuitive understanding of credit decisions while providing more nuanced importance rankings than the coefficient-based methods used in logistic regression.

## Random Forest Prediction Probabilities



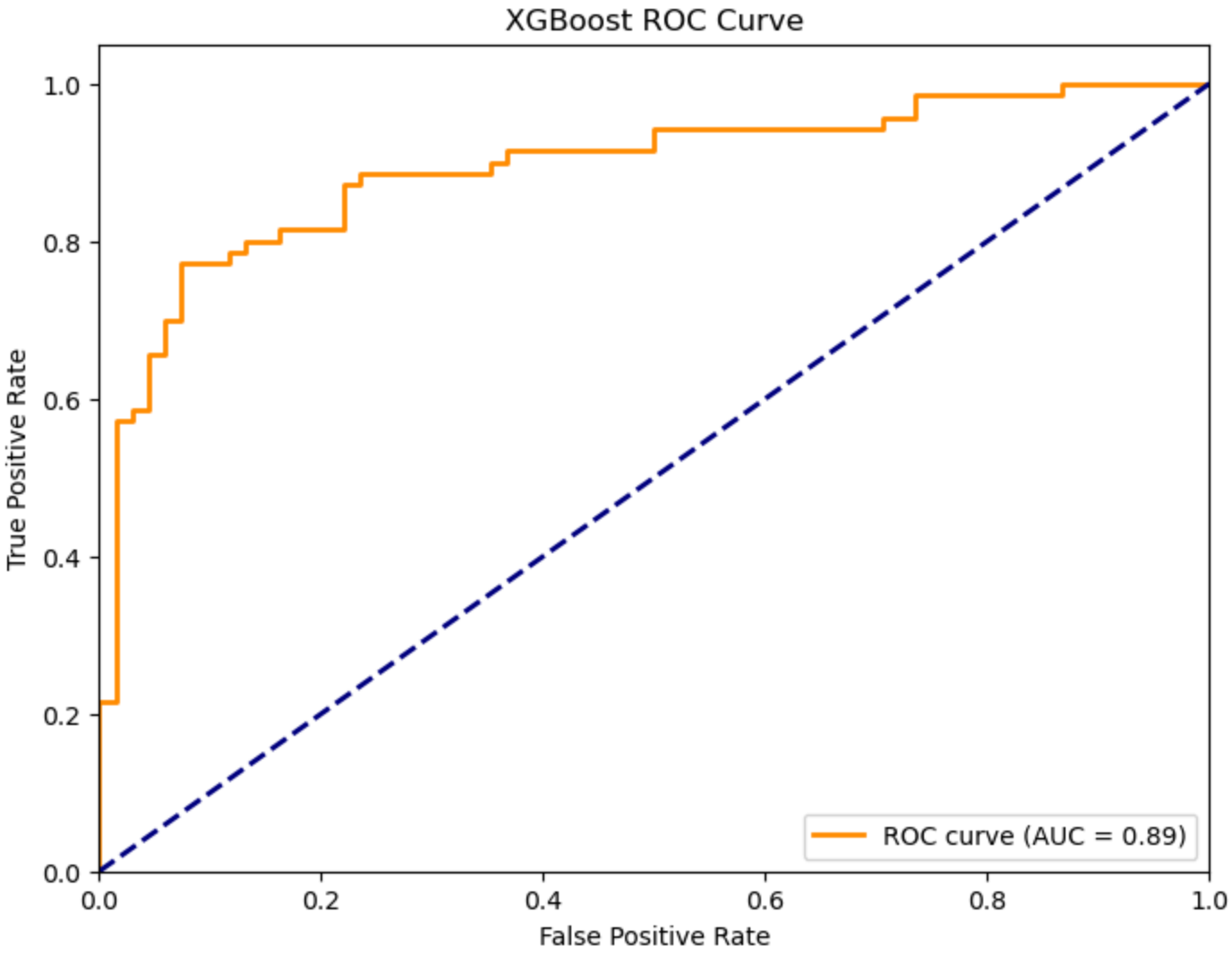

The prediction distribution shows clear separation between approval and rejection probabilities, with a bimodal pattern indicating confident predictions. This suggests the model has learned clear decision boundaries and makes predictions with high certainty.

## XGBoost Confusion Matrix



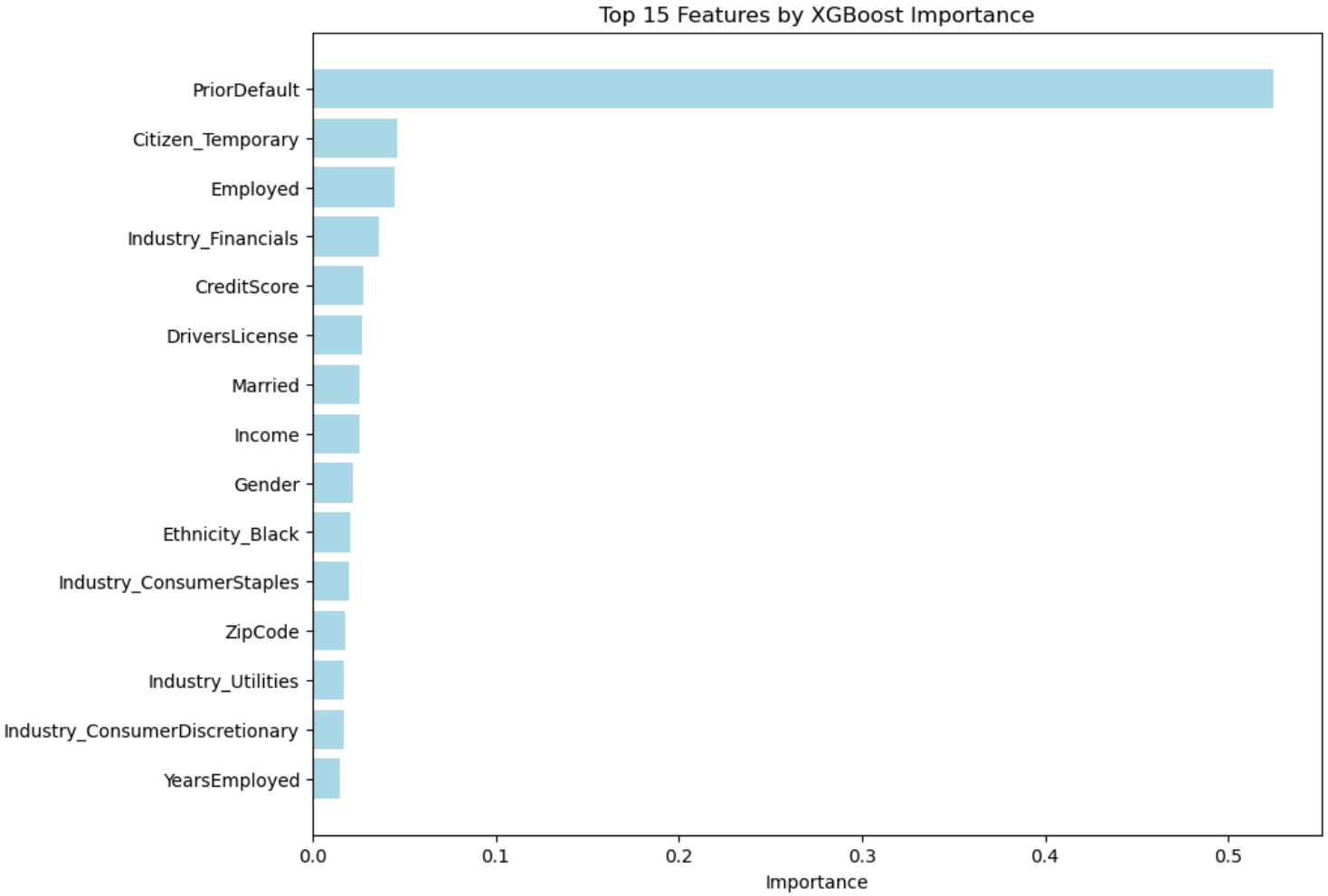

The XGBoost confusion matrix shows strong classification performance with 59 true positives and minimal misclassifications (9 false positives). This indicates that the model is particularly good at identifying applications that should be approved while maintaining a low rate of incorrect approvals.

## XGBoost ROC Curve



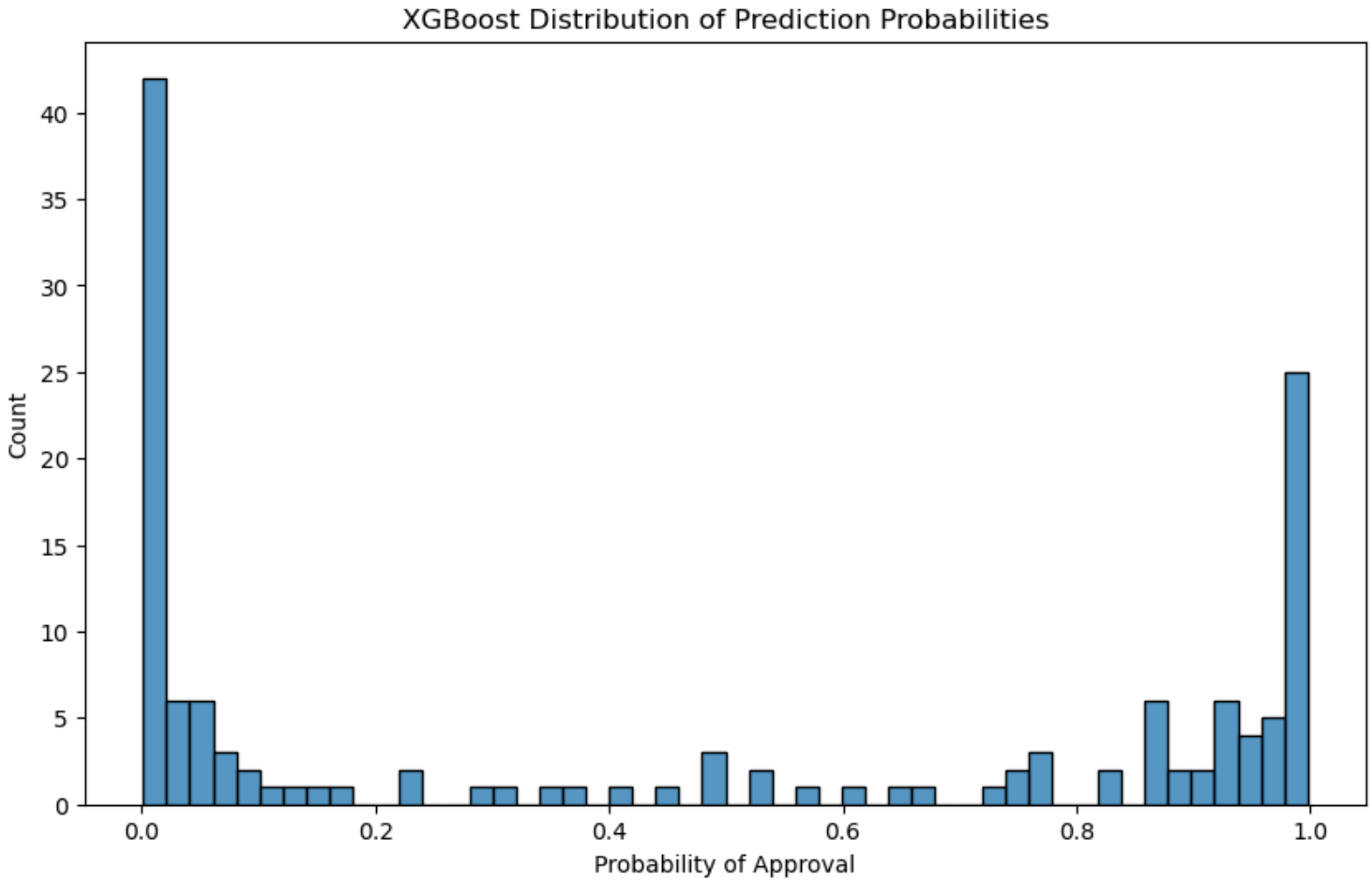

The XGBoost model achieves an impressive AUC of 0.89, very close to Random Forest's performance. The curve demonstrates excellent discrimination ability, with a sharp rise in the true positive rate while maintaining a low false positive rate. This suggests the model is highly effective at distinguishing between approval-worthy and non-approval-worthy applications.

## XGBoost Feature Importance




The feature importance analysis from XGBoost shows PriorDefault as the dominant feature, similar to Random Forest. However, it differs in subsequent important features, highlighting Citizen_Temporary, Employed, and Industry_Financials as key factors. This provides an interesting alternative perspective on feature relevance compared to other models.

## XGBoost Prediction Probabilities




The prediction probability distribution shows a strong bimodal pattern with clear peaks at 0 and 1, indicating high confidence in predictions. This suggests the model makes decisive predictions with approximately 40 cases receiving very low probability and 25 cases receiving very high probability of approval.

### Comparative Analysis and Recommendations

The four models show varying strengths:

- Logistic Regression: Best interpretability with strong performance (84.1%)
- Naive Bayes: Fastest training but lower accuracy (73.9%)
- Random Forest: Excellent feature insights and robust performance (84.0%)
- XGBoost: Balanced metrics and good handling of complex patterns (83.0%)

All ensemble methods outperform Naive Bayes, while Logistic Regression and Random Forest show the strongest overall performance.

## Gantt Chart

View Gantt Chart

# Next Steps

1. **Model Enhancement**
   - Explore model stacking techniques
   - Implement automated hyperparameter optimization
   - Develop prediction confidence metrics
2. **Feature Development**
   - Create interaction terms between key features
   - Investigate additional financial indicators
   - Implement feature selection optimization
3. **System Implementation**
   - Develop real-time prediction API
   - Create model interpretation dashboard
   - Establish performance monitoring system

# Contribution Table

| Name | Final Report Contributions |
|---|---|
| Vedanth Sathwik Toduru Madabushi | Ensemble Model Implementation, Final Analysis, Github Pages |
| Sathvik Vangavolu | Results Analysis, Model Comparison, Final Report |
| Siddharth Kolichala | Feature Analysis, Performance Evaluation, Final Report |
| Naman Solan | Visualization Analysis, Model Evaluation, Final Report |

Home Page

1. https://onlinelibrary.wiley.com/doi/abs/10.1002/isaf.1437?casa_token=qKjQzeDzlhwAAAAA%3ABeEwcn2C7OEdWFHUmnxgrr8E9dTA74PQtmmILJ6Rzf2XZdk7gFOUz89y9ejeFHk_mUGwXABUhotrCg ↩

2. https://www.ijsce.org/wp-content/uploads/papers/v11i2/B35350111222.pdf ↩

3. https://arxiv.org/abs/2409.16676 ↩

---