Credit Classifier Report

Kulindu Cooray
kulindu.coorays@gmail.com

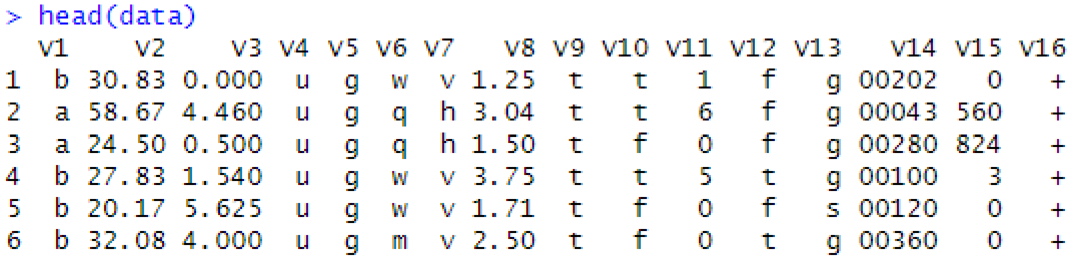
Abstract - Credit companies process numerous applications annually and must assess whether to approve or deny credit to each applicant. This paper investigates various methods and algorithms for classifying applicants as creditworthy or not. Using a credit approval dataset from the University of California, Irvine's Machine Learning Repository, several statistical models employing diverse techniques are applied. The study aims to identify accurate models and pinpoint the key factors influencing the likelihood of credit approval.

I. Introduction

Credit approval relies on evaluating an individual's financial history to determine their risk of default. Lenders gather applicant data to make informed decisions about whether to extend credit. This process requires statistical analysis and the development of predictive models to improve decision-making and ensure efficiency. By analyzing historical data, these models can provide accurate predictions about the likelihood of repayment. In this study, all statistical analyses will be conducted using R and RStudio.

II. Data

This paper uses the Credit Approval dataset from the University of California, Irvine Machine Learning Repository. The dataset originally includes 690 entries and 16 variables.

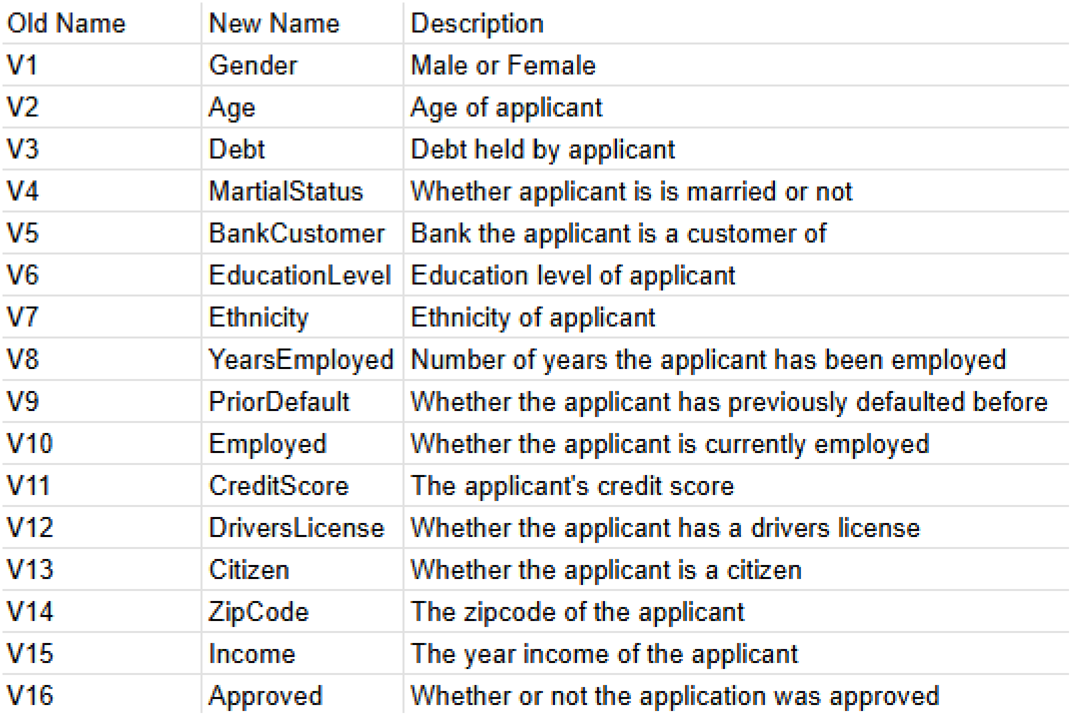


```
> head(data)
  V1    V2    V3 V4 V5 V6 V7   V8 V9 V10 V11 V12 V13   V14 V15 V16
1  b 30.83 0.000  u  g  w  v 1.25  t   t   1   f   g 00202   0   +
2  a 58.67 4.460  u  g  q  h 3.04  t   t   6   f   g 00043 560   +
3  a 24.50 0.500  u  g  q  h 1.50  t   f   0   f   g 00280 824   +
4  b 27.83 1.540  u  g  w  v 3.75  t   t   5   t   g 00100   3   +
5  b 20.17 5.625  u  g  w  v 1.71  t   f   0   f   s 00120   0   +
6  b 32.08 4.000  u  g  m  v 2.50  t   f   0   t   g 00360   0   +
```

However, it can't be analyzed as-is because of issues like multi class variables, differences in the scales of continuous variables, and missing data. Here's how these problems were addressed.

A. *Formatting*

To make the dataset easier to interpret, descriptive names were assigned to the original variable names. For example, V1 was converted into a binary gender variable, and V2 was converted into a age variable. The changes made are as follows:



Old Name	New Name	Description
V1	Gender	Male or Female
V2	Age	Age of applicant
V3	Debt	Debt held by applicant
V4	MartialStatus	Whether applicant is is married or not
V5	BankCustomer	Bank the applicant is a customer of
V6	EducationLevel	Education level of applicant
V7	Ethnicity	Ethnicity of applicant
V8	YearsEmployed	Number of years the applicant has been employed
V9	PriorDefault	Whether the applicant has previously defaulted before
V10	Employed	Whether the applicant is currently employed
V11	CreditScore	The applicant's credit score
V12	DriversLicense	Whether the applicant has a drivers license
V13	Citizen	Whether the applicant is a citizen
V14	ZipCode	The zipcode of the applicant
V15	Income	The year income of the applicant
V16	Approved	Whether or not the application was approved

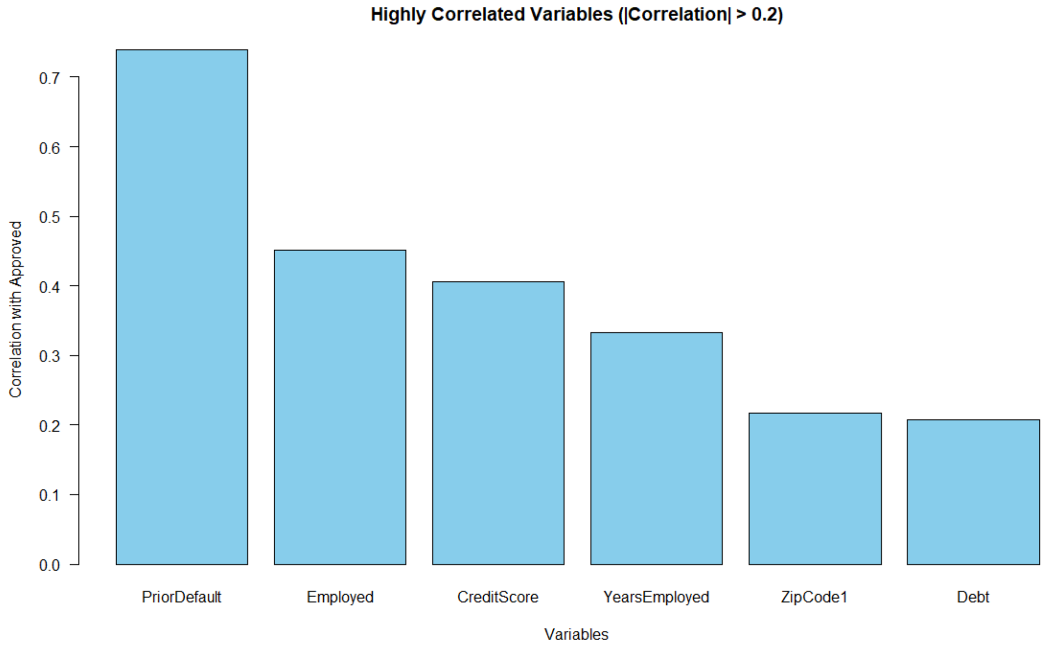
Next, discrete variables with multiple categories, such as education level, were split into separate binary variables to allow analysis in R. This process increased the total number of variables from 16 to 41.Missing data is a common challenge in large datasets, and the Credit Approval dataset contained 37 missing values. To address this, all observations with missing data were removed, reducing the dataset size to 653 complete entries.

III. Significant Variables

With the data preprocessed and ready for analysis, we can begin exploring the dataset. Among the 41 features available for prediction, not all will significantly contribute to determining whether an application is approved. Here, we outline methods to identify which predictors are most relevant. Additional techniques will be applied within specific models.

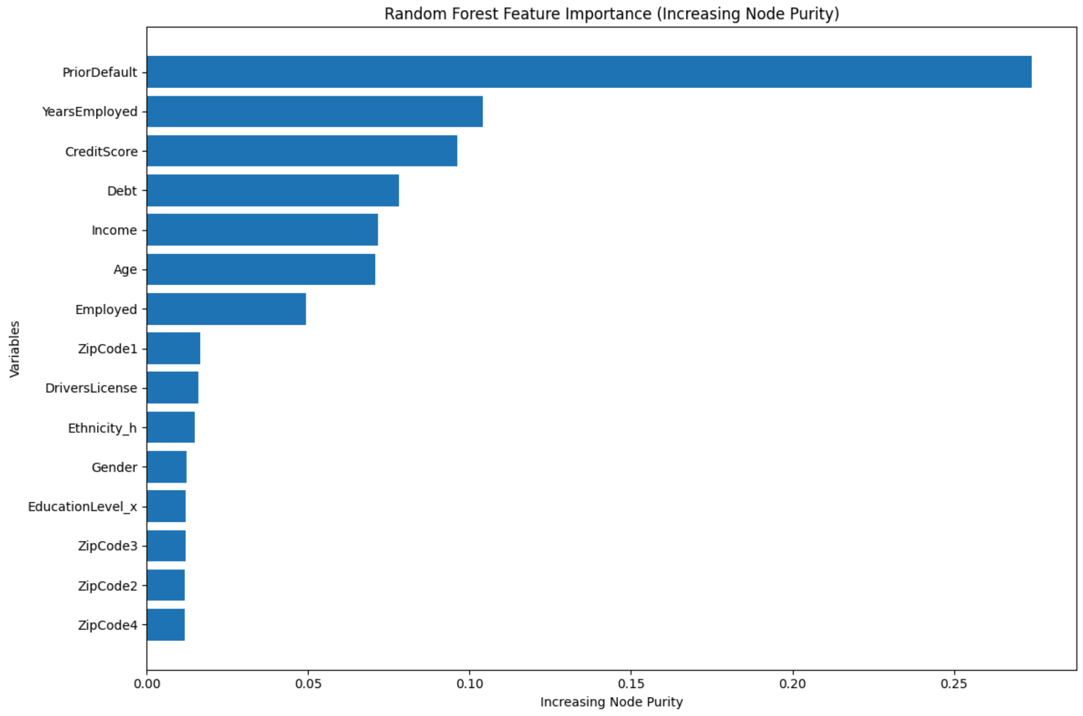
A. Correlated Variables

To start, we compute the correlation matrix for the dataset, focusing on the relationship between each variable and the Approved variable. Variables with a correlation greater than 0.2 are identified as the most relevant predictors. A bar chart visualizing these highly correlated variables is presented:



B. Random Forest

A Random Forest model identifies important predictors by measuring how much each variable reduces node impurity called Gini score. Higher values indicate greater predictive relevance.



The analysis reveals the most significant predictors: PriorDefault, CreditScore, YearsEmployed, Income, Debt, Age, and Employed. These findings align with the correlation analysis, confirming these seven variables as the most influential for loan approval prediction.

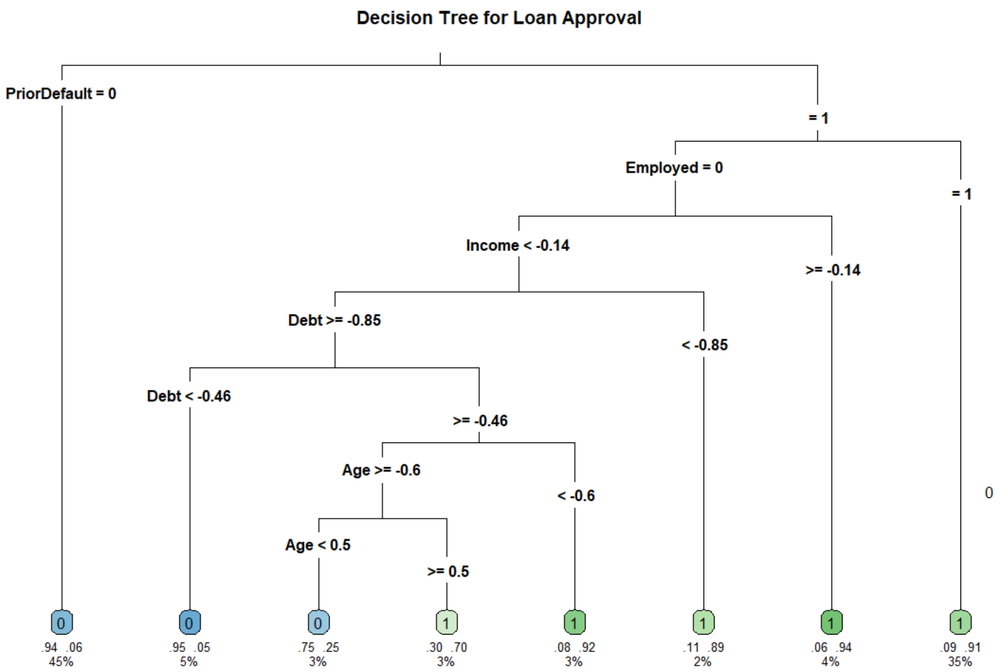
Geographic variables such as ZipCode show minimal importance, indicating location is not a significant factor. Therefore, the following Significant Predictors will be used in all classification models: For the classification models developed in this paper, the following variables will be used as the Significant Predictors:

PriorDefault, CreditScore, YearsEmployed, Income, Debt, Age, Employed

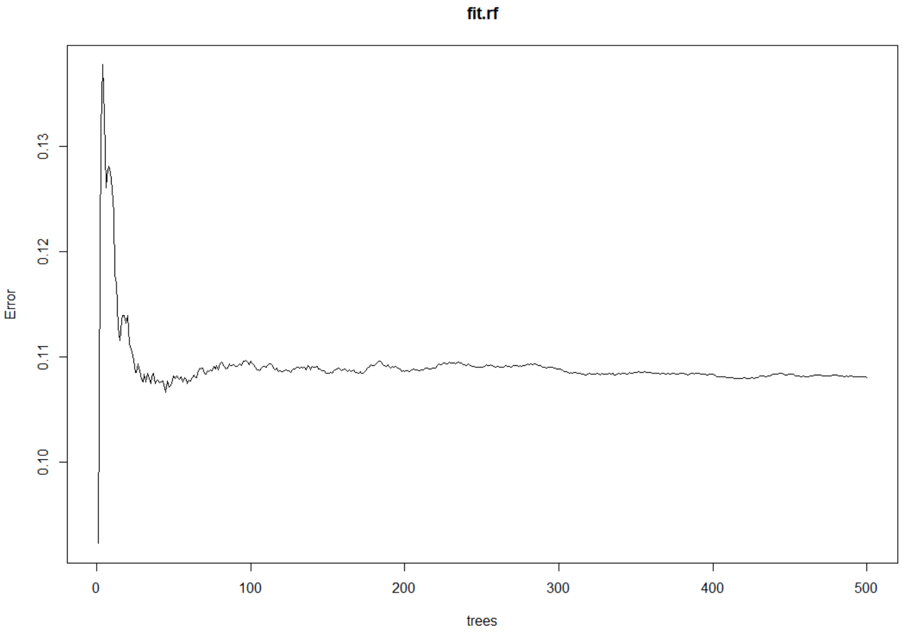
IV. Classification Models

A. Random Forest

Decision trees are a non-parametric method for classification. The tree splits the data by asking a series of yes/no questions. Each split evaluates a specific condition and the process goes further down the tree structure. Below is an example of a decision tree generated using the `rpart` library.



Single decision trees often show high variance, meaning their predictions can differ based on small changes in the data. One method to reduce this variance is to average the predictions of multiple trees. Instead of averaging an arbitrary number different trees, random forests improve this approach by using de-correlation. The predictors are chosen randomly for each split. This randomness helps ensure that individual trees are less similar to one another. Averaging the predictions in large numbers of random trees reduces test error and provides a more accurate model.

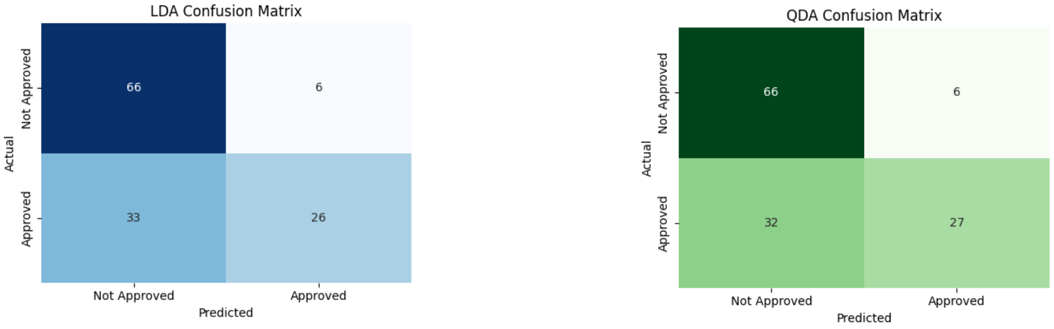


As shown above, increasing the number of trees significantly reduces the test error, particularly up to approximately 100 trees. Beyond this point, additional trees provide little to no improvement in performance. The random forest classification method achieves an accuracy of 90.08% accuracy using significant features and 90.84% accuracy using all features.

B. Discriminant Analysis

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are classification methods that separate classes by modeling the distribution of predictors. LDA assumes that both classes share the same covariance structure, leading to a linear decision boundary. QDA, on the other hand, allows each class to have its own covariance structure, resulting flexible quadratic decision boundaries.

Applied to the data, LDA got 70.23% accuracy and QDA did slightly better at 70.99%. The small gap shows QDA's extra flexibility doesn't help much, and both models perform weaker than other methods.



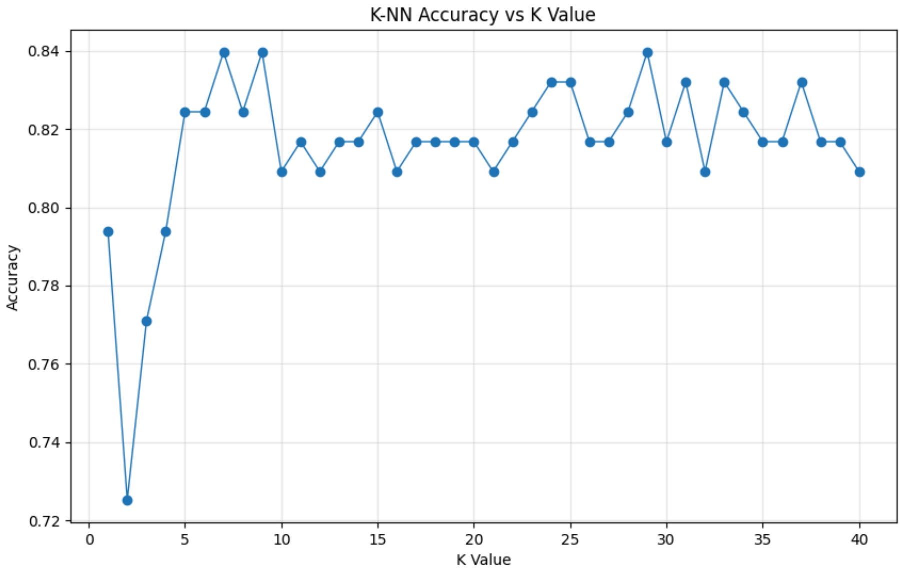
Accuracy: 70.23%

Accuracy: 70.99%

Figure 1: LDA and QDA confusion matrices with respective accuracies.

C. K-Near Neighbors (KNN)

Next we are using K nearest neighbors (KNN) method for classification. This algorithm works by calculating the distance to the k nearest data points and classifying an applicant based on the greatest magnitude among those points. The choice of K heavily influences the performance of the model, as it determines the number of neighbors considered in the decision.



The optimal value of K is 7 with an accuracy of 84% accuracy.

D. Support Vector Machine (SVM)

The SVM models work by finding a hyperplane in the feature space such that it separates it into distinct regions where data from different classes are placed. A model will find an optimum boundary to divide data samples. Here, I used four SVM models, using linear and radial kernel on both the significant features and all features to define the shape of the decision boundary.

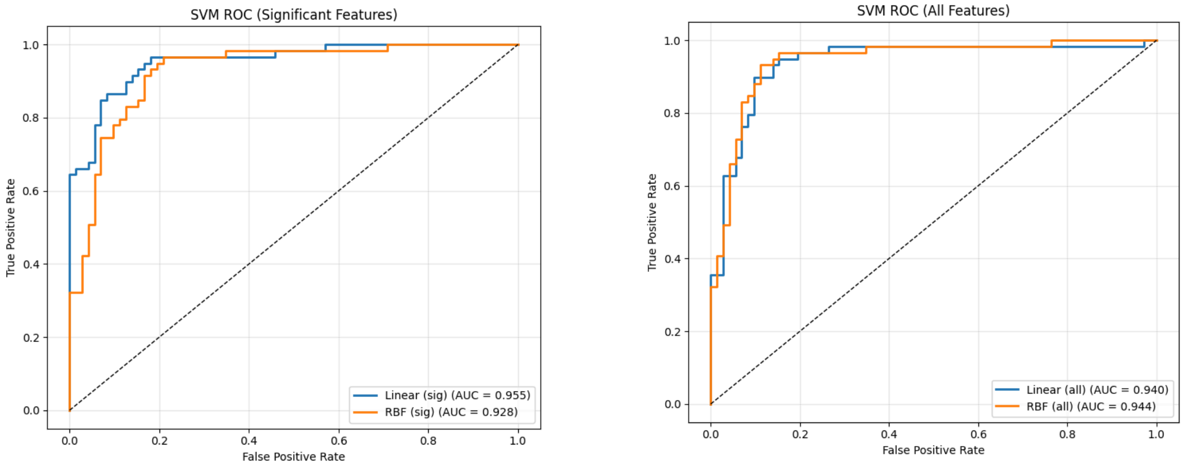
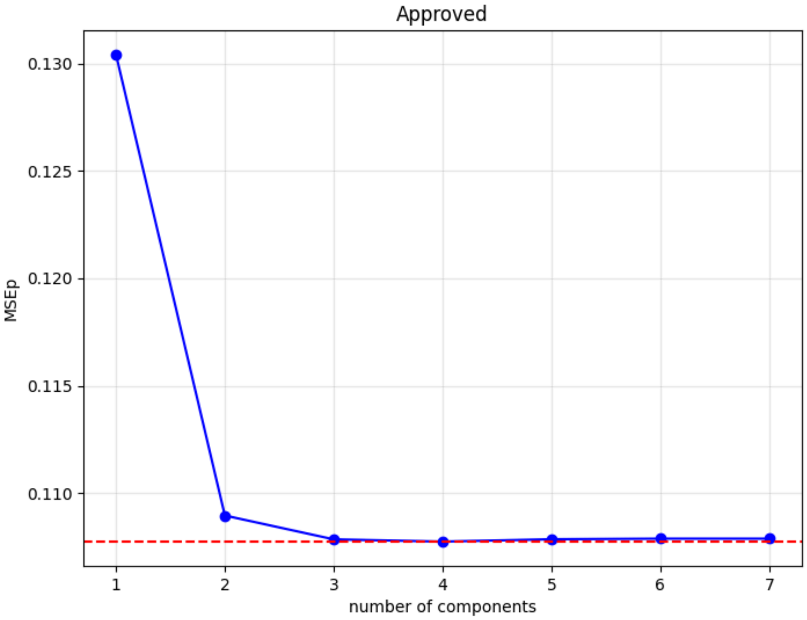


Figure 2: ROC curves for SVM Models

The linear model has the best performance and barely outperforms the radial kernel functions. For accuracy SVM linear kernel, it received 87.79% for using both significant and all features. For the radial kernel, it received 86.26% accuracy and 89.31% accuracy for significant and all features respectively.

E. Partial Least Squares (PLS)

Unlike other dimensionality reduction approaches, PLS incorporates the response variable when identifying components. The model assigns greater weight to predictors that are more strongly correlated with the response variable. The results of running the model are as follows:



The plot shows that error drops quickly when increasing from one to two components and levels off by three. Using more than three components gives little improvement, meaning three components are enough to capture the main signal without overfitting. By using three components, the results were 87.79% accuracy with significant features and 90.84% accuracy with all features.

F. Linear

$$Y = X\beta + \varepsilon$$

The linear probability model was fit using ordinary least squares. The histograms below show the predicted values for both the significant feature set and the full set of variables. As expected, predictions are spread continuously rather than restricted to probabilities between 0 and 1, with some falling outside this range. A threshold of 0.5 was applied to convert predictions into class labels.

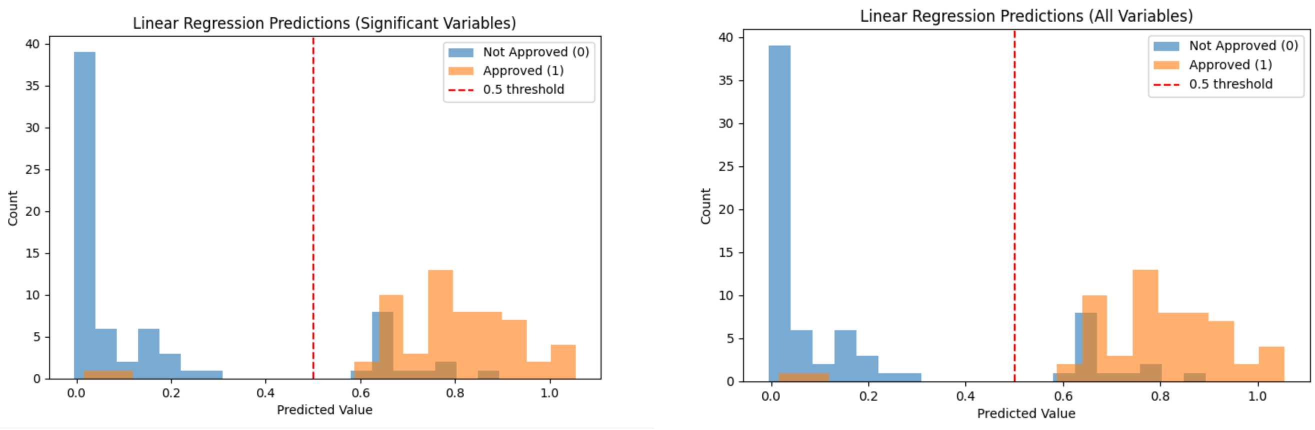


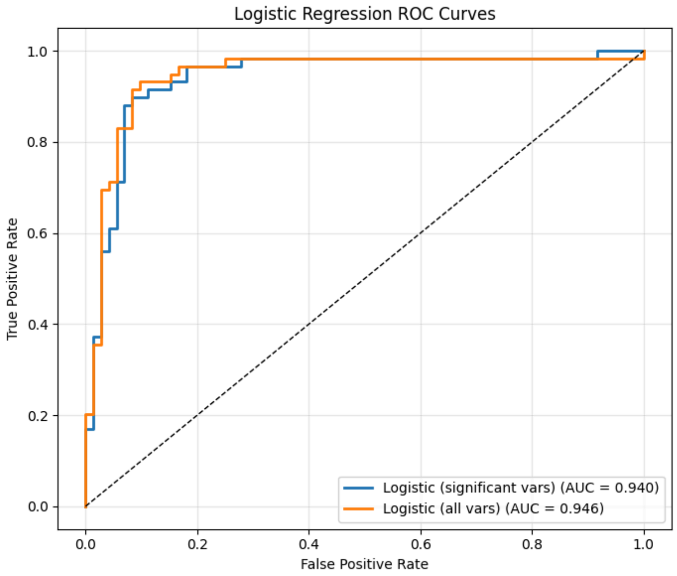
Figure 3: Histogram of Linear Regression Predictions

Linear regression with significant features achieves 87.79% accuracy while all features achieves 90.08% accuracy, and the distributions look very similar across the two specifications. This suggests that adding all features does not substantially improve predictive power compared to using only the significant subset. Instead, the key drivers remain consistent with prior models: prior default, employment status, income, and credit score.

G. Logistic

$$Y = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}$$

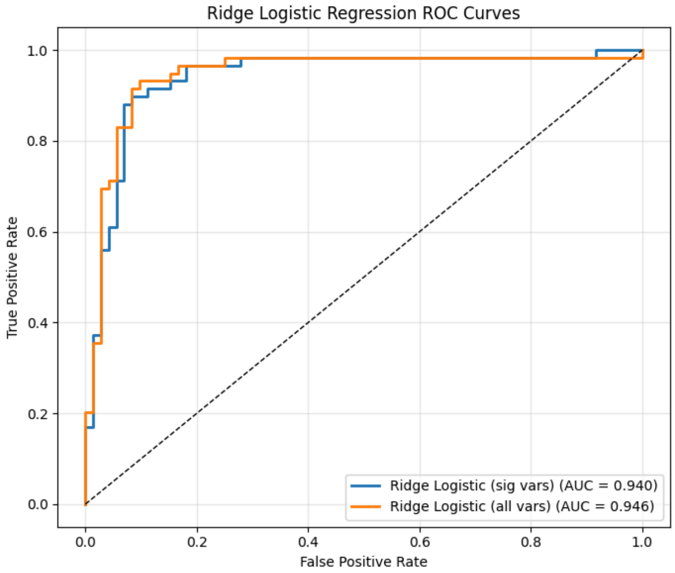
In the logistic regression analysis, I assess the performance of two models. These models differ based on their selection of predictor variables, with one using all features and the other using significant predictors.



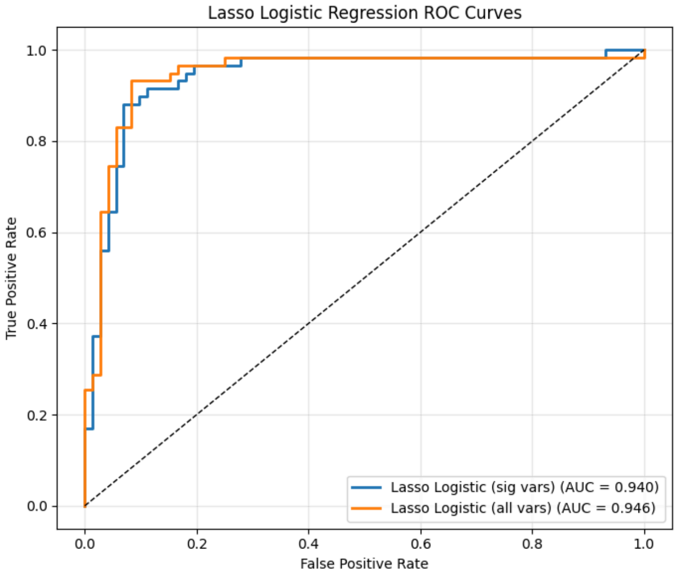
Both logistic regression models show strong performance, with AUC values of 0.940 (significant predictors) and 0.946 (all predictors). The slight gain from using all features is marginal, indicating that the significant subset captures most of the predictive information. The models achieved an accuracy of 87.79% and 90.08% using significant and all features respectively.

I. Ridge

Ridge regularization applies an L2 penalty that shrinks coefficients to prevent over fitting while maintaining all predictors. The ROC curves show that both the significant-variable model and the all-variable model achieve strong predictive power, with AUC values of 0.940 and 0.946, respectively. The small gap suggests that adding all features provides little additional benefit, and most of the predictive signal is already captured by the significant subset. The models achieved an accuracy of 87.79% and 90.84% using significant and all features respectively.



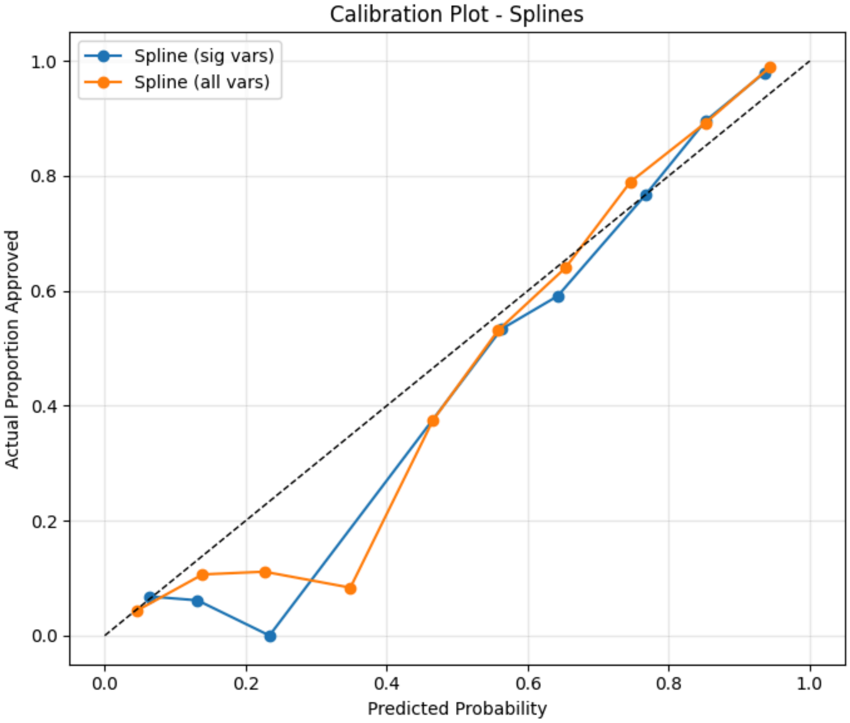
J. LASSO



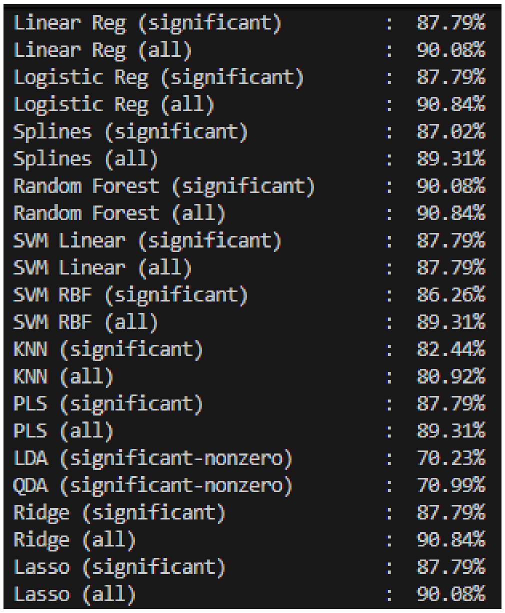
Lasso regularization applies an L1 penalty that shrinks less important coefficients to zero, performing feature selection while maintaining predictive accuracy. The ROC curves show that both the significant-variable and all-variable models achieve strongw performance, with AUC values of 0.940 and 0.946. The similarity of results indicates that most predictive power lies within the significant predictors, while Lasso helps confirm which variables contribute meaningfully by excluding weaker ones. The models achieved an accuracy of 87.79% and 90.08% using significant and all features respectively.

K. Splines

By applying spline transformations to the numeric predictors, the model can flexibly capture non-linear relationships in the data. The models achieve strong performance with accuracies of 87.02% (significant variables) and 89.31% (all variables). The calibration plot shows that predicted probabilities track the actual approval rates reasonably well, staying close to the diagonal reference line. The small difference in accuracy suggests that adding all features only provides a marginal gain over the significant subset.



V. Conclusion



```
Linear Reg (significant)         :  87.79%
Linear Reg (all)                 :  90.08%
Logistic Reg (significant)       :  87.79%
Logistic Reg (all)               :  90.84%
Splines (significant)            :  87.02%
Splines (all)                    :  89.31%
Random Forest (significant)      :  90.08%
Random Forest (all)              :  90.84%
SVM Linear (significant)         :  87.79%
SVM Linear (all)                 :  87.79%
SVM RBF (significant)            :  86.26%
SVM RBF (all)                    :  89.31%
KNN (significant)                :  82.44%
KNN (all)                        :  80.92%
PLS (significant)                :  87.79%
PLS (all)                        :  89.31%
LDA (significant-nonzero)        :  70.23%
QDA (significant-nonzero)        :  70.99%
Ridge (significant)              :  87.79%
Ridge (all)                      :  90.84%
Lasso (significant)              :  87.79%
Lasso (all)                      :  90.08%
```

The table above highlights that among all the models tested, the best subset linear model achieves the highest test accuracy.

The analysis indicates that PriorDefault is the most relevant variable in terms of acceptance or rejection of credit applications. Further, income and credit score are the other major variables affecting application outcomes. However, education, race, and marital status are not significant from a statistical perspective.

Many of the regression techniques returned very similar test performances, indicating that an 87% accuracy might be close to the best possible performance for this size dataset. More flexible models would likely continue to improve in their accuracy with more data points, allowing them to capture more nuanced patterns in the data.

VI. References

1. Deepesh Khaneja, “Credit Approval Analysis using R”, Technical Report, November 2017
2. Ms.D.Jayanthi,“Credit Approval Data Analysis Using Classification and Regression Models”, International Journal of Research and Analytical Reviews, vol5 (3), 2018
3. Fu, Z and Z, Liu, “Classifier Comparisons On Credit Approval Prediction”, Final project of Course CS229 in 2014 at Stanford University