Loan Default Prediction

Super Rookie

Rongxuan Wang(rw2900), Yan Dong(yd2625), Yinjue Yi(yy3193)
Zhihan Zhou(zz2885), Jiatong Zhang(jz3401)

Abstract

Personal loans are indispensable for the daily life of many people nowadays. And distinguishing whether a client could fully pay their loan on time is the top-drawer question for lending banks. The objectives of our project are to use machine learning models to predict whether the loan could be fully paid on time or not and explore important features that might be useful for future loan decisions.

Our dataset contains the accepted loans between 2007 to 2018 from LendingClub website, with data size of 150 columns and 2 million rows. 150 columns contain 1 predicting variable, loan_status, and 149 relevant variables, such as loan amount, term, interest rate, grade, etc.

Data Preprocessing

Feature Selection

Since we get 149 relevant variables, which are too many to all be our features for analyzing, we first do feature selection by carefully reviewing the description for each variable, and then grab 21 features from them which make perfect sense to us for understanding the loan information.

Examples:

Annual_inc: The self-reported annual income provided by the borrower during registration.

Delinq_2yrs: The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years.

Then, by correlation measurement, we preserve only one feature among features with high correlation. We finally get 18 features. (Appendix 1)

Example:

Interest_rate and *grade* variables have a correlation approximately to 0.98. After field research, we find that the grade is assigned to each loan according to its interest rate. The higher the interest rate is, the higher the grade it would have. So, we preserve *grade* variable and drop *interest_rate.*

Data Cleaning & Feature Engineering

Our original data contains more than two million rows, so we directly use dropna() to drop the rows which contain null values. Then we get a dataset with one million rows that is sufficient for us to build models on.

For predicting variable *loan_status*, it has 7 statuses: ‘Fully Paid’, ‘Current’, ‘Charged Off’, ‘In Grace Period’, ‘Late (31-120 days)’, ‘Late (16-30 days)’, ‘Default’. According to our objective, we regularize the status into two parts: first drop ‘Current’ and ‘In Grace Period’, then set ‘Fully Paid’ as 1, others as 0, which represent not fully paid on time (default, charge off and late).

Finally, we turn three category features – *term, grade and verification_status* – into numerical types, by using cat.codes. For example, *grade*: ABCDEFG -> 0123456; *term*: 36 months -> 0 and 60 months -> 1; *verification_status*: not verified -> 0, source verified -> 1, fully verified -> 2.

Our current data contains 19 columns, 18 features and 1 predicting variable, and 1 million rows, with no null values and all variables in numerical type.

Data Visualization

Variable Correlation

According to the correlation coefficient matrix (Appendix 2), we can see that the relationships between variables are all not strong. In order to check the multicollinearity issue in our models, we apply variance inflation factor analysis on 18 features. As reported by the VIF table (Appendix 3), we can clearly see that the variance inflation factor of features are all below 5, which means that there is no multicollinearity issue in our models. This also means that all features can be included as predictors because they all have small correlations with others.

Categorical Features Analysis

In our features, we have three categorical predictors: *grade, term*, and *verification status*. We analyze these three variables through the default rate distributions and the number distributions in three aspects: the overall data distribution, the default data distribution, and the non-default data distribution.

According to the distribution of default rate on *grade* (Appendix 4), we can see that this predictor has an apparent linear uptrend. With the grade number increasing, the default rate increases. As the higher grade number means that the loan is worse, it makes sense that the loan with higher grade number would be more likely to default. As reported by the number distribution of *grade* (Appendix 5), we can see that the difference between the default number and non-default number decreases with the grade number increasing and the data weighs slightly heavier on top grade levels than the latter grade levels. Overall, the *grade* variable tends to become an important predictor due to its apparent trend on the default rate distribution and not heavy skewed data distribution.

As shown in the distribution of default rate on *term* (Appendix 6), it is obvious to find that the loans with longer term have higher probability to default. This makes sense because longer-term loans tend to have higher risk. This distinguished trend makes it more possible for this feature to positively influence our result. According to the number distribution of *term* (Appendix 7), we can see that the data weighs heavier on the short-term loans than the long-term loans, which might have a negative impact on the predicting accuracy.

As reported by the distribution of default rate on *verification status* (Appendix 8), we can see that the default rate slightly increases with the income verified deeper and deeper. This is opposite to our understanding because we think if the income is verified deeper, the default rate should be lower. This ambiguous trend of default rate distribution tends to make this feature have small contributions in our models. As shown in the number distribution of *verification status* (Appendix 9), we can see the data weigh nearly equally on three kinds of verification status, which make it more meaningful to include this feature.

Modeling and Results

According to our processed dataset and features of different models, like learning rate, accuracy and explainability, we choose the following 6 feasible models: linear regression, logistic regression, decision tree, random forest, neural network and SVM.

After selecting models, we use random under_sampling to balance our data so as to improve the prediction performance. Then we split the total data set into a train set (70%) and a test set (30%).

Basic linear regression: Linear regression fits a linear model with coefficients w = (w1,...,wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Before running the model, we use *StandardScaler().fit_transform()* to standardize the data set. As linear regression returns a numeric value instead of a classification, a threshold is set to get classification from the result of the regression that the *loan_status* is identified as 'Fully Paid' if the prediction from regression is larger than the threshold. Through multiple tests, we choose 0.5 to be the threshold. The basic linear model has test scores with Accuracy: 64.90%, Precision: 64.51%, Recall: 66.50% and AUC of 0.7061 (Appendix 10).

Linear regressions with Lasso: Lasso regression uses a hyperparameter alpha to penalize the sum of absolute values of weights and would shrink some coefficients to 0 if they have low explanatory power. In our dataset, all coefficients are shrunk to 0 when applying Lasso regression, so it is not appropriate to use Lasso for prediction here.

Linear regressions with Ridge: Ridge regression also uses a hyperparameter alpha, but to penalize the sum of squared values of weights and alpha is selected to be 1 to get the lowest error. The ridge regression model has test scores with Accuracy: 64.90%, Precision: 64.51%, Recall: 66.50% and AUC of 0.7061 (Appendix 11), nearly the same with the basic linear regression.

Logistic regression: Logistic regression returns a binary variable for classification and we use *LogisticRegressionCV()* to set up the model, with *solver='saga', max_iter=10000, Cs=1.* The model has test scores with Accuracy: 64.96%, Precision: 64.97% and Recall: 66.09% and AUC of 0.6495 (Appendix 12). Sorting the absolute value of the regression coefficients, we could get the feature importance, *grade* and *term* are the top three features. Adding the lasso penalty(*penalty='l1'),* the model has test scores with Accuracy: 63.94%, Precision: 62.90% and Recall: 68.24% and AUC of 0.6393. Adding the ridge penalty(*penalty='l2'),* the model has test scores with Accuracy: 64.95%, Precision: 64.69% and Recall: 66.09% and AUC of 0.6495.

Decision Tree: Decision tree does the classification by learning decision rules from data features. The *max_depth* of our tree model is set to be 4 (Appendix 13). The test scores: Accuracy: 66.43%, Precision: 69.01%, Recall: 60.42%, AUC: 0.6647 (Appendix 14).

Random Forest: Random Forest takes multiple decision trees into consideration. We have used GridSearchCV to find the best parameter combination. Our model returned the majority of classifications out of 1000 trees with test scores: Accuracy: 67.28%, Precision: 67.74% Recall: 66.77% and an AUC of 0.6728 (Appendix 15). Feature importance was also calculated on this model (Appendix 16). *dti* (the ratio of total monthly debt payments to the borrower's income) appears to be the most important feature.

Neural Network: For the neural network, which is flexible but time-consuming, we first standardize independent variables to improve the speed and stability of our model. Then we use Hyperparameter tuning to find the best hyperparameters, maximizing the model's performance, minimizing a predefined loss function to produce better results with fewer errors. Through grid search, we choose epochs 5 and bitch_size 10 for the neural network. The test scores: Accuracy: 65.36%, Precision: 67.17%, Recall: 60.56% and an AUC of 0.7124 (Appendix 17).

SVM: A similar process is performed for the SVM model. Moreover, since SVM needs even more time to train than Neural Network, we randomly choose 100,000 data and use cross validation to improve performance. The test scores: Accuracy: 66.43%, Precision: 66.14%, Recall: 64.21%, AUC: 0.6552 (Appendix 18). *grade* is the most important feature in the SVM model(Appendix 19).

Conclusion

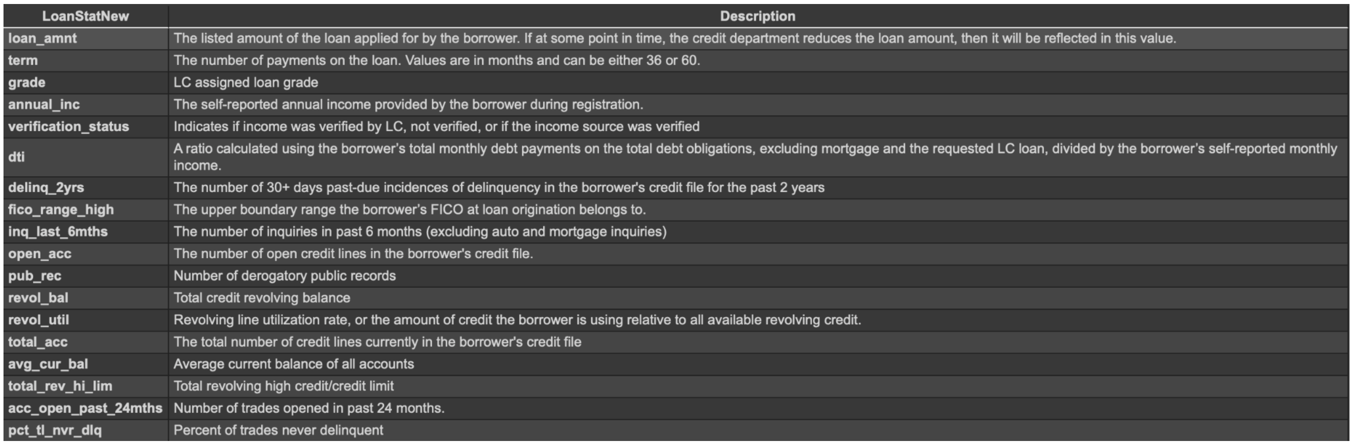
Based on our analysis, we find that the random forest model is the best model considering the four test scores and confusion matrix, followed by decision trees model, neural network model, SVM model, logistic regression model and linear regression model. According to the feature importance analysis, we find *grade* and *dti* (the ratio of total monthly debt payments to the borrower's income) appears to be the most significant features to predict whether the loan will be fully paid on time or default.

Future Improvements

In order to make the model perform better, we can try to improve from the perspectives of algorithms and data sources. On the current dataset, some other classifiers can be considered, such as KNN, XGBoost, etc. In terms of data sources, people's willingness to repay is influenced by many factors, so data sources can also be more diverse. For example, macro data and local environmental data may affect individuals' willingness to repay. To investigate the relationship, we can combine different datasets using the key of geographical locations, and use our model to explore whether introducing new data can improve accuracy. When there are many data dimensions, principal component analysis can be used to evaluate the contribution of each dimension and select the dimensions with higher explanatory power.

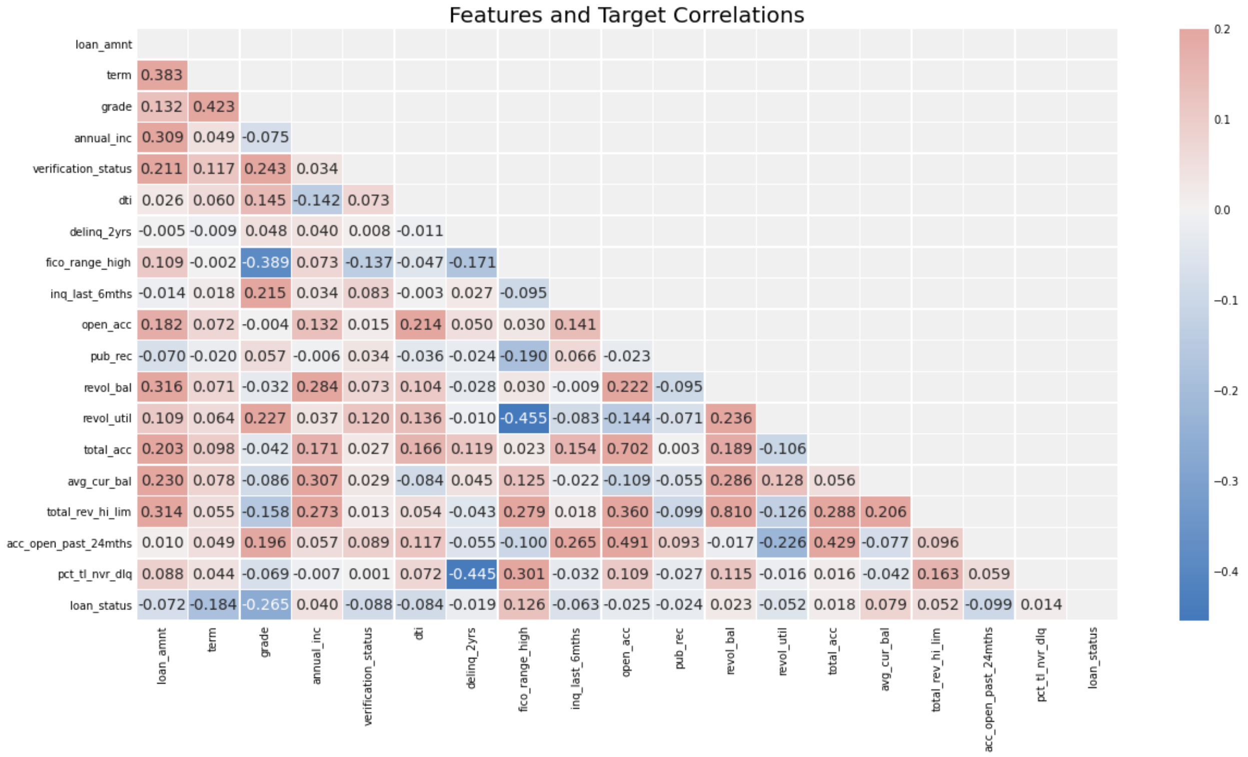
Appendix

Appendix 1: Variable Description



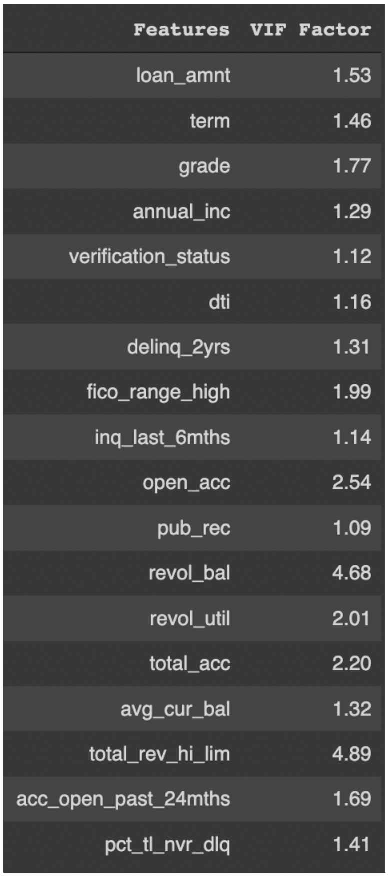
LoanStatNew	Description
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
grade	LC assigned loan grade
annual_inc	The self-reported annual income provided by the borrower during registration.
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
fico_range_high	The upper boundary range the borrower's FICO at loan origination belongs to.
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
open_acc	The number of open credit lines in the borrower's credit file.
pub_rec	Number of derogatory public records
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
total_acc	The total number of credit lines currently in the borrower's credit file
avg_cur_bal	Average current balance of all accounts
total_rev_hi_lim	Total revolving high credit/credit limit
acc_open_past_24mths	Number of trades opened in past 24 months.
pct_tl_nvr_dlq	Percent of trades never delinquent

Appendix 2: Correlation Coefficient Matrix



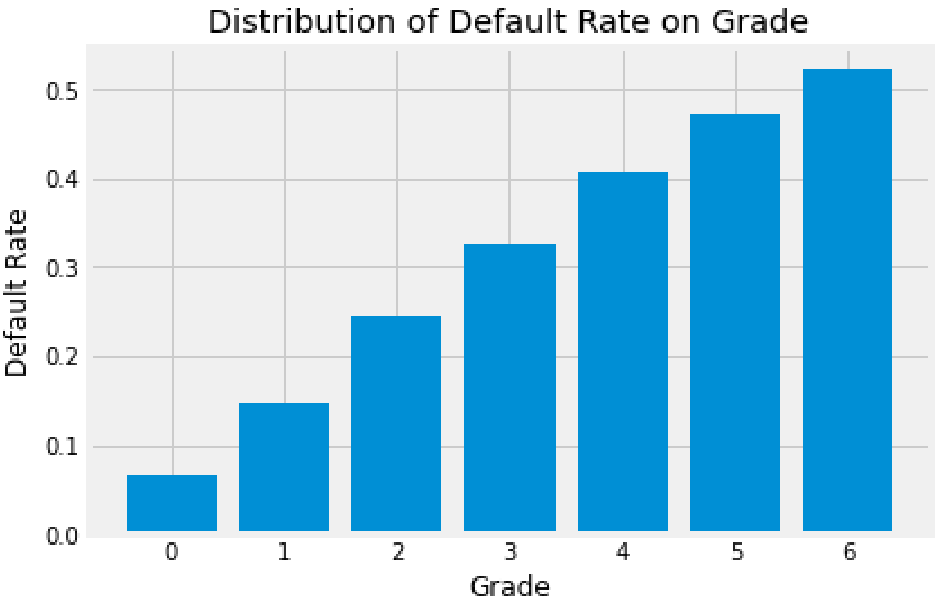
	loan_amnt	term	grade	annual_inc	verification_status	dti	delinq_2yrs	fico_range_high	inq_last_6mths	open_acc	pub_rec	revol_bal	revol_util	total_acc	avg_cur_bal	total_rev_hi_lim	acc_open_past_24mths	pct_tl_nvr_dlq	loan_status
loan_amnt																			
term	0.383																		
grade	0.132	0.423																	
annual_inc	0.309	0.049	-0.075																
verification_status	0.211	0.117	0.243	0.034															
dti	0.026	0.060	0.145	-0.142	0.073														
delinq_2yrs	-0.005	-0.009	0.048	0.040	0.008	-0.011													
fico_range_high	0.109	-0.002	-0.389	0.073	-0.137	-0.047	-0.171												
inq_last_6mths	-0.014	0.018	0.215	0.034	0.083	-0.003	0.027	-0.095											
open_acc	0.182	0.072	-0.004	0.132	0.015	0.214	0.050	0.030	0.141										
pub_rec	-0.070	-0.020	0.057	-0.006	0.034	-0.036	-0.024	-0.190	0.066	-0.023									
revol_bal	0.316	0.071	-0.032	0.284	0.073	0.104	-0.028	0.030	-0.009	0.222	-0.095								
revol_util	0.109	0.064	0.227	0.037	0.120	0.136	-0.010	-0.455	-0.083	-0.144	-0.071	0.236							
total_acc	0.203	0.098	-0.042	0.171	0.027	0.166	0.119	0.023	0.154	0.702	0.003	0.189	-0.106						
avg_cur_bal	0.230	0.078	-0.086	0.307	0.029	-0.084	0.045	0.125	-0.022	-0.109	-0.055	0.286	0.128	0.056					
total_rev_hi_lim	0.314	0.055	-0.158	0.273	0.013	0.054	-0.043	0.279	0.018	0.360	-0.099	0.810	-0.126	0.288	0.206				
acc_open_past_24mths	0.010	0.049	0.196	0.057	0.089	0.117	-0.055	-0.100	0.265	0.491	0.093	-0.017	-0.226	0.429	-0.077	0.096			
pct_tl_nvr_dlq	0.088	0.044	-0.069	-0.007	0.001	0.072	-0.445	0.301	-0.032	0.109	-0.027	0.115	-0.016	0.016	-0.042	0.163	0.059		
loan_status	-0.072	-0.184	-0.265	0.040	-0.088	-0.084	-0.019	0.126	-0.063	-0.025	-0.024	0.023	-0.052	0.018	0.079	0.052	-0.099	0.014	

Appendix 3: Variance Inflation Factor (VIF)

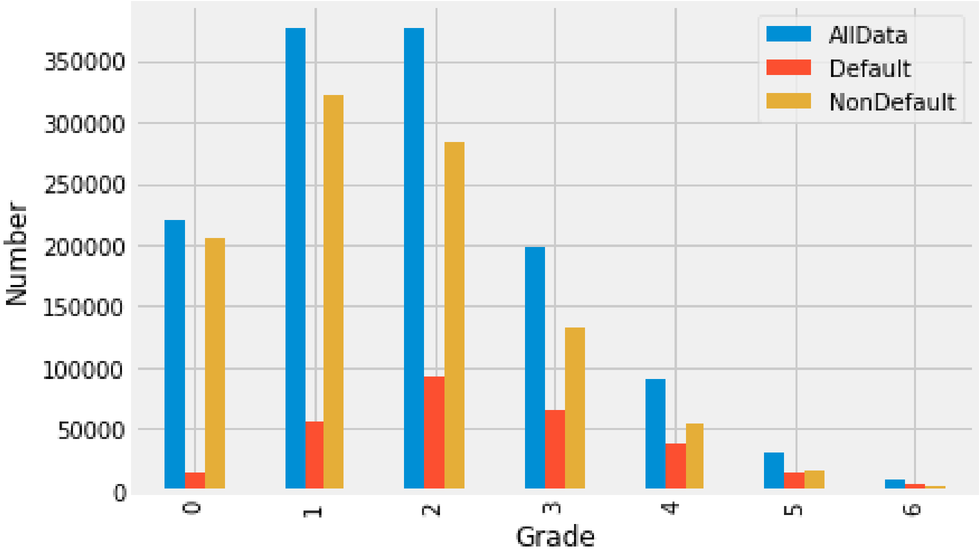


Features	VIF Factor
loan_amnt	1.53
term	1.46
grade	1.77
annual_inc	1.29
verification_status	1.12
dti	1.16
delinq_2yrs	1.31
fico_range_high	1.99
inq_last_6mths	1.14
open_acc	2.54
pub_rec	1.09
revol_bal	4.68
revol_util	2.01
total_acc	2.20
avg_cur_bal	1.32
total_rev_hi_lim	4.89
acc_open_past_24mths	1.69
pct_tl_nvr_dlq	1.41

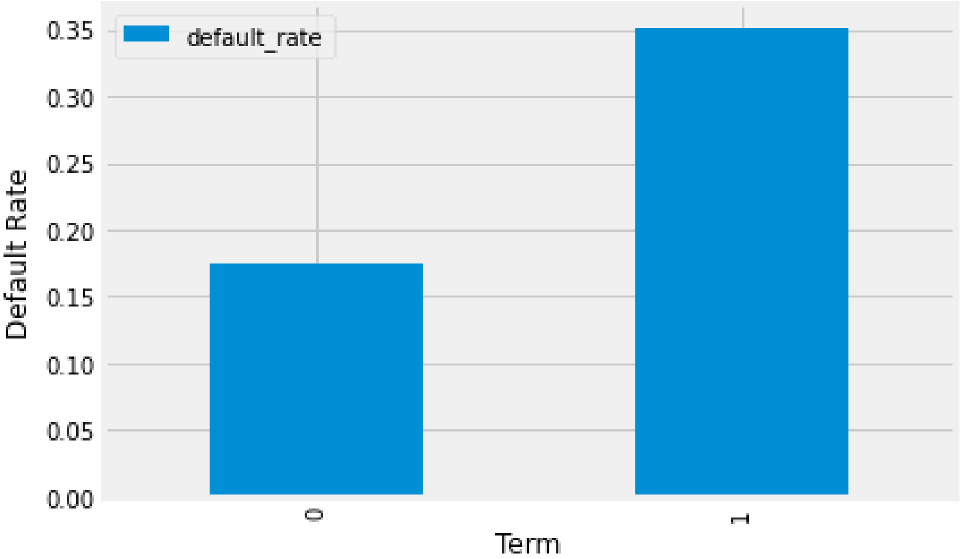
Appendix 4: Distribution of Default Rate on Grade



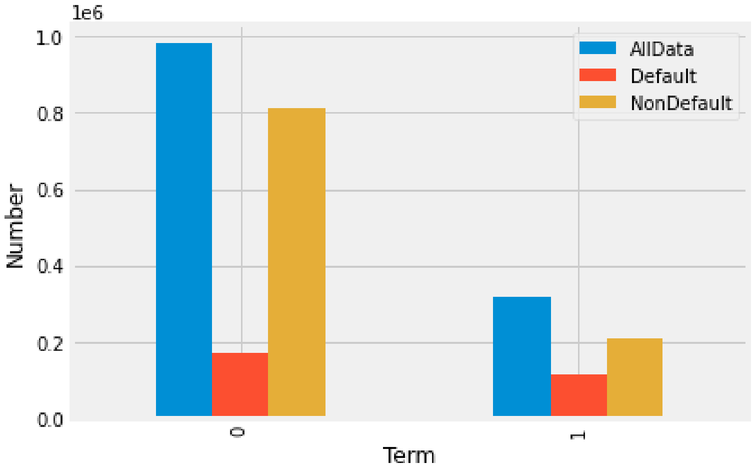
Appendix 5: Number Distribution of Grade



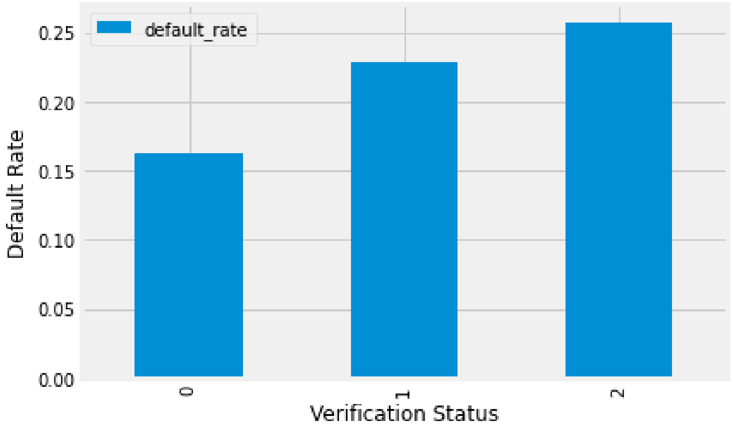
Appendix 6: Distribution of Default Rate on Term



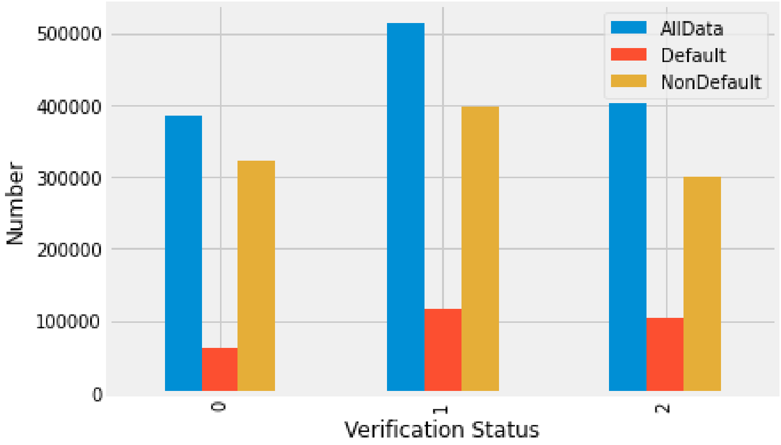
Appendix 7: Number Distribution of Term



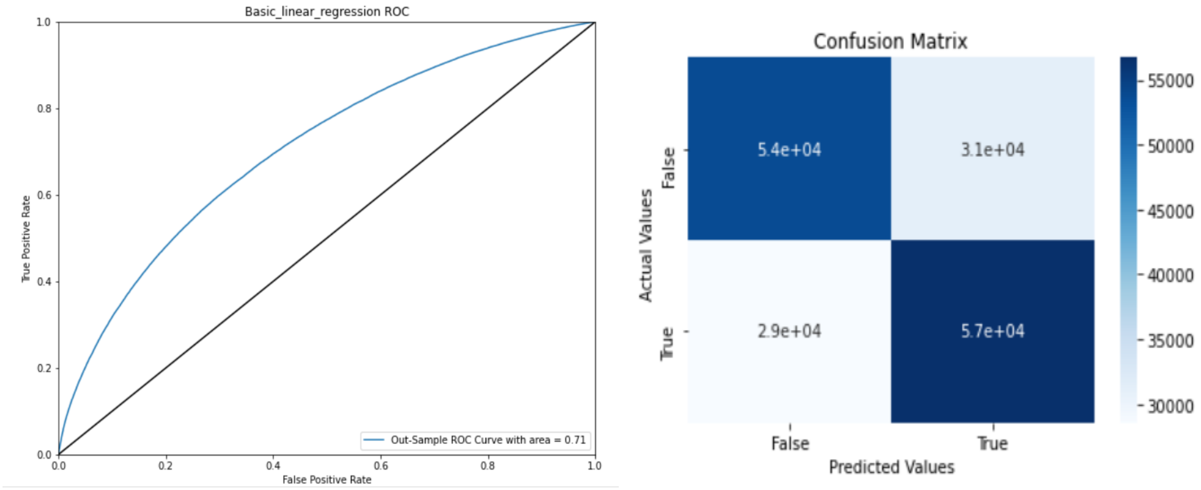
Appendix 8: Distribution of Default Rate on Verification Status



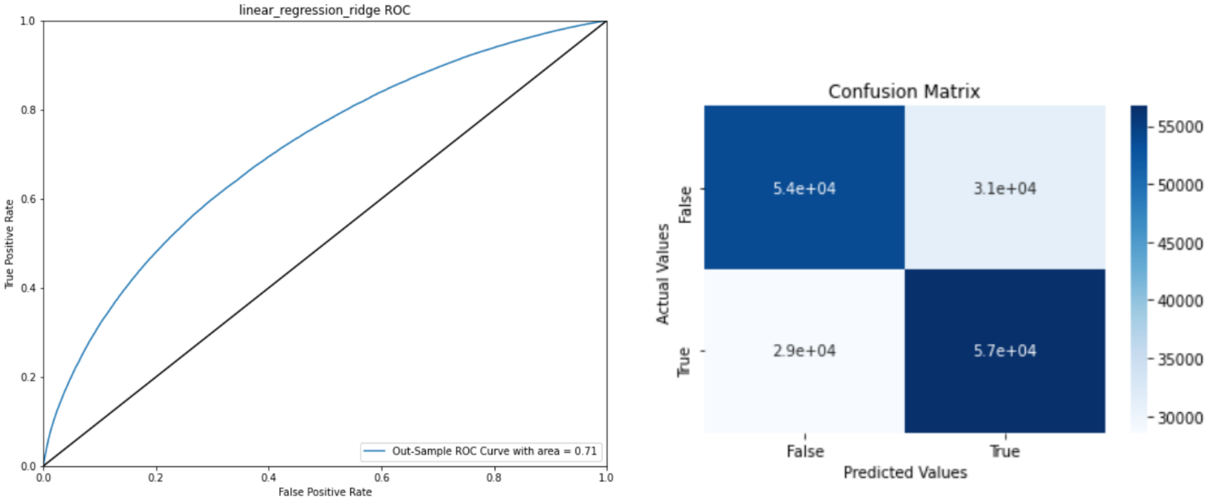
Appendix 9: Number Distribution of Verification Status



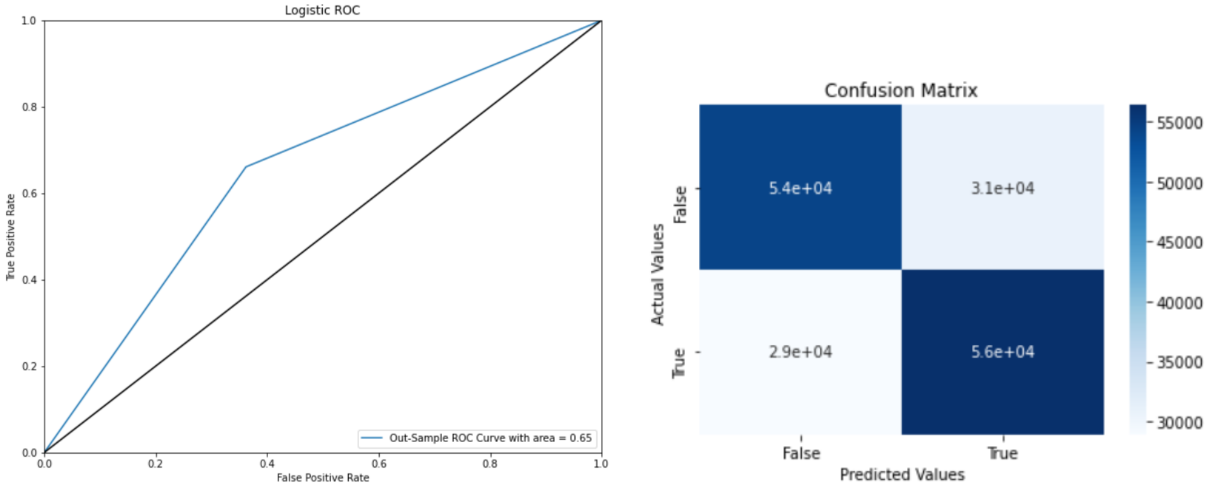
Appendix 10: Basic Linear Regression ROC and Confusion Matrix



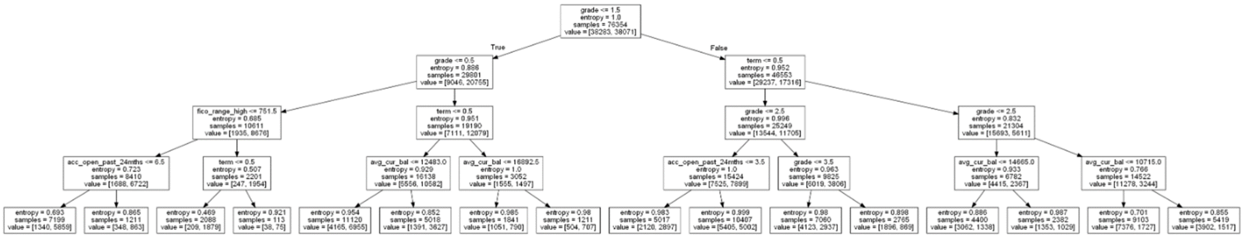
Appendix 11: Ridge Regression ROC and Confusion Matrix



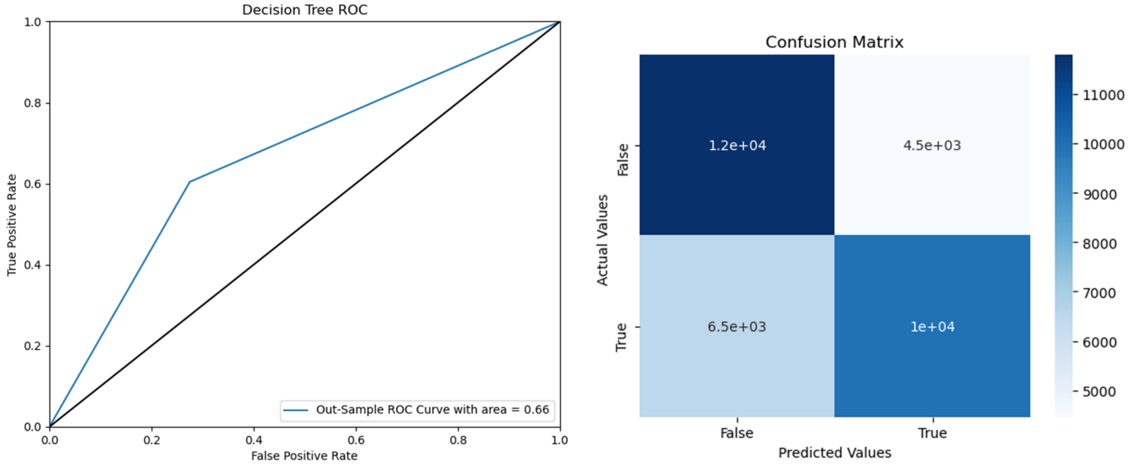
Appendix 12: Logistic Regression ROC and Confusion Matrix



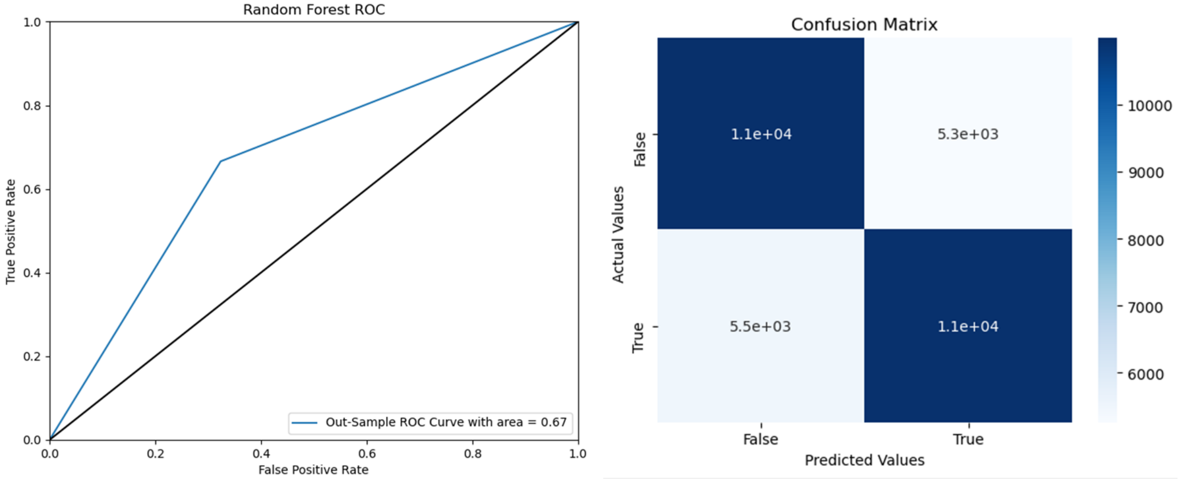
Appendix 13: Structure of Decision Tree


grade <= 1.5 entropy = 1.0 samples = 76354 value = [38283, 38071]
True
False
grade <= 0.5 entropy = 0.886 samples = 29801 value = [9046, 20755]
term <= 0.5 entropy = 0.952 samples = 46553 value = [29237, 17316]
fico_range_high <= 751.5 entropy = 0.685 samples = 10611 value = [1935, 8676]
term <= 0.5 entropy = 0.951 samples = 19190 value = [7111, 12079]
grade <= 2.5 entropy = 0.996 samples = 25249 value = [13544, 11705]
grade <= 2.5 entropy = 0.832 samples = 21304 value = [15693, 5611]
acc_open_past_24mths <= 6.5 entropy = 0.723 samples = 8410 value = [1688, 6722]
term <= 0.5 entropy = 0.507 samples = 2201 value = [247, 1954]
avg_cur_bal <= 12483.0 entropy = 0.929 samples = 16138 value = [5556, 10582]
avg_cur_bal <= 16892.5 entropy = 1.0 samples = 3052 value = [1555, 1497]
acc_open_past_24mths <= 3.5 entropy = 1.0 samples = 15424 value = [7525, 7899]
grade <= 3.5 entropy = 0.963 samples = 9825 value = [6019, 3806]
avg_cur_bal <= 14665.0 entropy = 0.933 samples = 6782 value = [4415, 2367]
avg_cur_bal <= 10715.0 entropy = 0.766 samples = 14522 value = [11278, 3244]
entropy = 0.693 samples = 7199 value = [1340, 5859]
entropy = 0.865 samples = 1211 value = [348, 863]
entropy = 0.469 samples = 2088 value = [209, 1879]
entropy = 0.921 samples = 113 value = [38, 75]
entropy = 0.954 samples = 11120 value = [4165, 6955]
entropy = 0.852 samples = 5018 value = [1391, 3627]
entropy = 0.985 samples = 1841 value = [1051, 790]
entropy = 0.98 samples = 1211 value = [504, 707]
entropy = 0.983 samples = 5017 value = [2120, 2897]
entropy = 0.999 samples = 10407 value = [5405, 5002]
entropy = 0.98 samples = 7060 value = [4123, 2937]
entropy = 0.898 samples = 2765 value = [1896, 869]
entropy = 0.886 samples = 4400 value = [3062, 1338]
entropy = 0.987 samples = 2382 value = [1353, 1029]
entropy = 0.701 samples = 9103 value = [7376, 1727]
entropy = 0.855 samples = 5419 value = [3902, 1517]

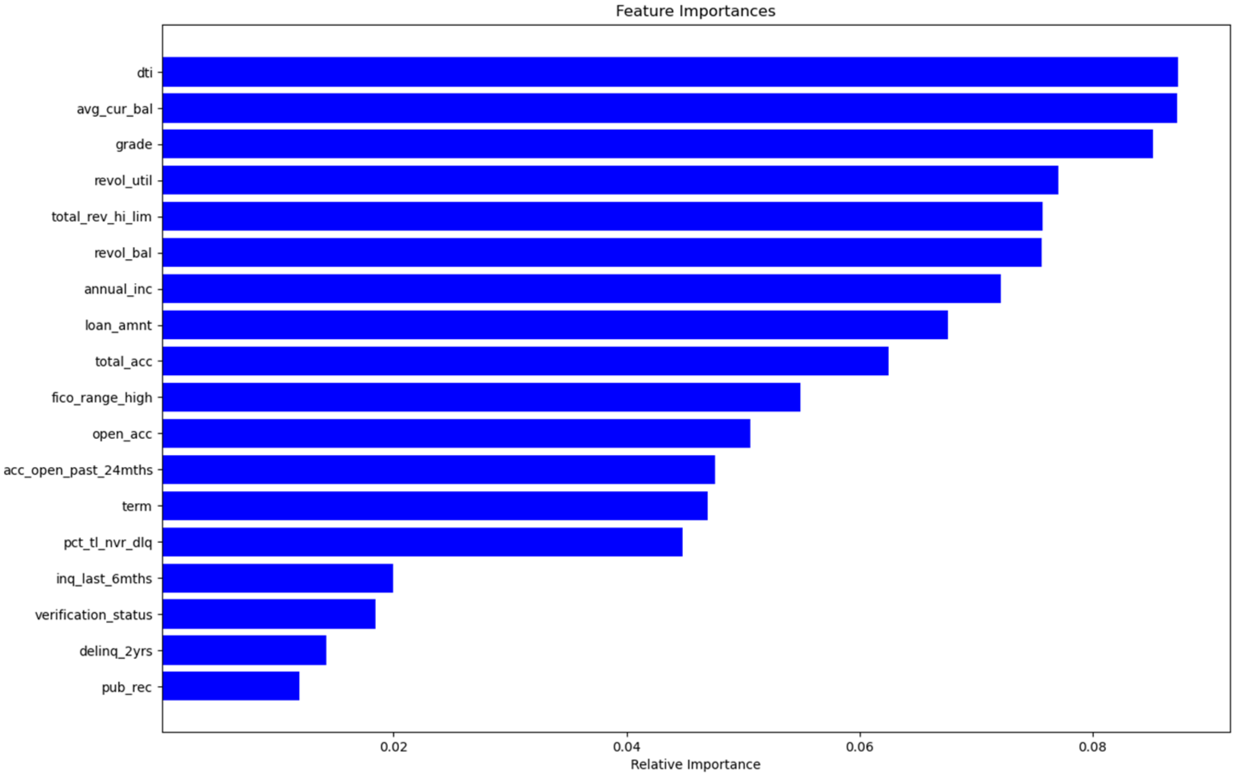
Appendix 14: Decision Tree ROC and Confusion Matrix



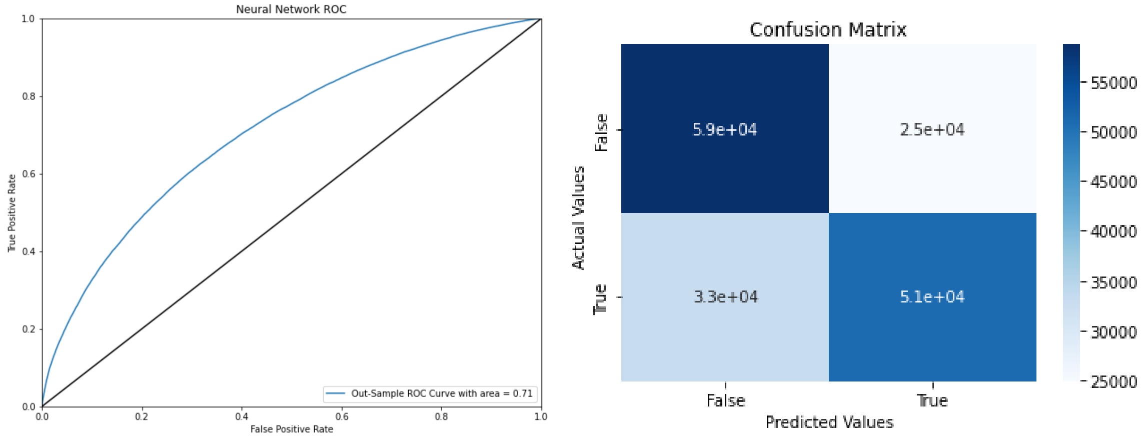
Appendix 15: Random Forest ROC and Confusion Matrix



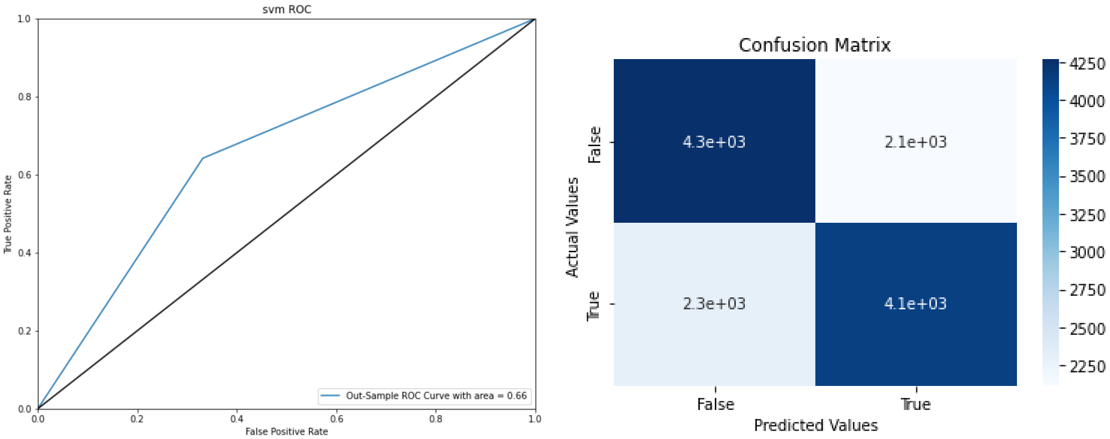
Appendix 16: Feature Importance of Random Forest



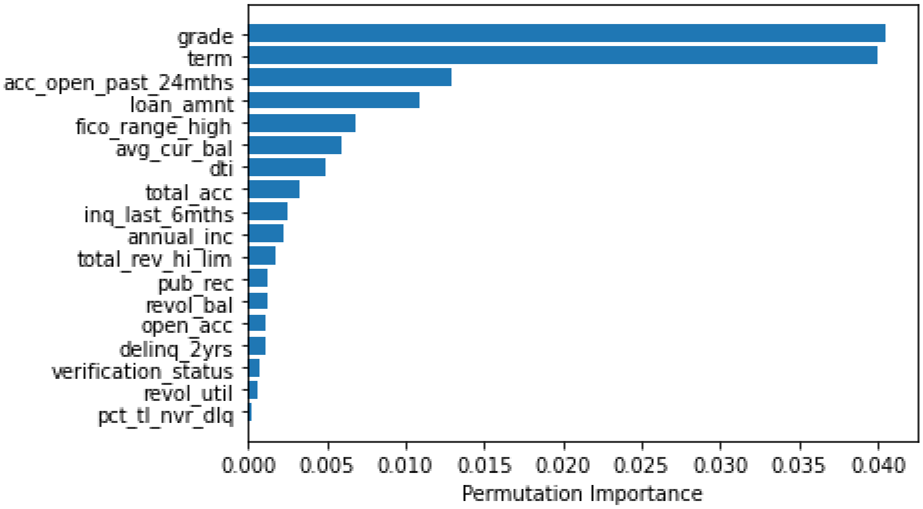
Appendix 17: Neural Network ROC and Confusion Matrix



Appendix 18: SVM ROC and Confusion Matrix



Appendix 19: Feature Importance of SVM



Appendix 20: Model Results

Modeling	Accuracy	Precision	Recall	AUC
Basic linear regression	64.90%	64.51%	66.50%	0.7061
Linear regressions with Ridge	64.90%	64.51%	66.50%	0.7061
Logistic regression	64.96%	64.97%	66.09%	0.6495
Logistic-Ridge Penalty	64.95%	64.69%	66.09%	0.6495
Logistic-Lasso Penalty	63.94%	62.90%	68.24%	0.6393
Decision Tree	66.43%	69.01%	60.42%	0.6647
Random Forest	67.28%	67.74%	66.77%	0.6728
Neural Network	65.36%	67.17%	60.56%	0.7124
SVM	65.52%	66.14%	64.21%	0.6552