UNIVERSITY OF CALIFORNIA
Los Angeles

Loan Repayment Prediction
Using Machine Learning Algorithms

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics

by

Chang Han

2019

ABSTRACT OF THE THESIS

Loan Repayment Prediction
Using Machine Learning Algorithms

by

Chang Han
Master of Applied Statistics in
University of California, Los Angeles, 2019
Professor Yingnian Wu, Chair

In the lending industry, investors provide loans to borrowers in exchange for the promise of repayment with interest. If the borrower repays the loan, then the lender would make profit from the interest. However, if the borrower fails to repay the loan, then the lender loses money. Therefore, lenders face the problem of predicting the risk of a borrower being unable to repay a loan. In this study, the data from Lending club is used to train several Machine Learning models to determine if the borrower has the ability to repay its loan. In addition, we would analyze the performance of the models (Random Forest, Logistic Regression, Support Vector Machine, and K Nearest Neighbors). As a result, logistic regression model is found as the optimal predictive model and it is expected that Fico Score and annual income significantly influence the forecast.

The thesis of Chang Han is approved.

Nicolas Christou

Hongquan Xu

Frederic R Paik Schoenberg

Yingnian Wu, Committee Chair

University of California, Los Angeles

2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would thank Prof. Wu for his generous support and help with all of the advices. I also thank Nicolas Christou, Prof. Xu, and Prof. Schoenberg for all of the comments that greatly improved my thesis.

CHAPTER 1

Introduction

The loan is one of the most important products of the financial institutes. All the institutes are trying to figure out effective business strategies to persuade more customers to apply their loans. However, there are some customers are not able to pay off the loan after their application are approved. Therefore, many Financial institutions take several variables into account when approving a loan[Hon18]. Determining whether a given borrower will fully pay off the loan or cause it to be charged off (not fully pay off the loan) is difficult. If the lender is too strict, fewer loans get approved, which means there's less interest to collect. But if they're too lax, they end up approving loans that default [Bha18]. In this study, loan behaviors are analyzed with several machine learning models.

The dataset that used in this paper is from Lending Club, a website that connects borrowers and investors over the Internet. It includes 9,578 observations that were funded through the LendingClub.com platform between May 2007 and February 2010.

The logistic regression is widely used to solve the classification problem. In some previously cases, the logistic regression was used to estimate the loan repayment status. However, sometimes this may not be very accurate because logistic regression may only capture linear relationships between variables [PLI02]. Sometimes, the data are not limited to have the linear relationships between variables. Many different models have also been used to deal with this kind of classification problem. Such as Naïve Bayesian model. However, it relies on independence assumption and will perform badly if this assumption is not met.

In this thesis, exploratory data analysis is applied to check and handle the missing values, and necessary data transformations is conducted to process the data in Chapter two. In the third chapter, several machine learning models were trained to predict for the loan repayment.

The machine learning models include: Logistic Regression, Random Forest, KNN (K nearest neighbors), SVM (supporting vector mechanine). In Chapter three, each model will be evaluated via k-fold classification techniques and confusion matrix. Conclusions and further improvements are discussed in the last chapter.

CHAPTER 2

Exploratory Data Analysis

The success of classification learning is heavily dependent on the quality of the data provided for training. In this chapter, we will have an overview of the loan repayment data set and perform a data exploratory analysis in order to determine to preprocess the data and improve the prediction result. The data will also be split into training set (70%), and test set (30%). Training set will be used to fit the model, and test set will be to evaluate the best model to get an estimation of generalization error.

2.1 Feature Analysis

In this study, the publicly available data from Lending Club was used. The data can be accessed from LendingClub.com. The data covers the 9,578 loans funded by the platform between May 2007 and February 2010. Table 2.1 demonstrates the first 5 rows of the dataset.

Table 2.1: First five rows of data set

credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs	pub.rec	not.fully.paid
1	debt_consolidation	0.1189	829.1	11.35040654	19.48	737	5639.958333	28854	52.1	0	0	0	0
1	credit_card	0.1071	228.22	11.08214255	14.29	707	2760	33623	76.7	0	0	0	0
1	debt_consolidation	0.1357	366.86	10.37349118	11.63	682	4710	3511	25.6	1	0	0	0
1	debt_consolidation	0.1008	162.34	11.35040654	8.1	712	2699.958333	33667	73.2	1	0	0	0
1	credit_card	0.1426	102.92	11.29973224	14.97	667	4066	4740	39.5	0	1	0	0

Below is a short description of each feature in the data set:

- credit_policy: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.
- purpose: The purpose of the loan such as: credit card, debt consolidation, etc.

- int_rate: The interest rate of the loan (proportion).
- installment: The monthly installments ($) owed by the borrower if the loan is funded.
- log_annual_inc: The natural log of the annual income of the borrower.
- dti: The debt-to-income ratio of the borrower.
- fico: The FICO credit score of the borrower.
- days_with_cr_line: The number of days the borrower has had a credit line.
- revol_bal: The borrower's revolving balance.
- revol_util: The borrower's revolving line utilization rate.
- inq_last_6mths: The borrower's number of inquiries by creditors in the last 6 months.
- delinq_2yrs: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.
- pub_rec: The borrower's number of derogatory public records.
- not_fully_paid: indicates whether the loan was not paid back in full (the borrower either defaulted or the borrower was deemed unlikely to pay it back).

By checking the data type and definition of each variable, the variables could be classified into two groups: numerical and categorical. Figure 2.1 is the density plot to check distribution of each numerical variable. Figure 2.2 is the count plot to check the distribution of each categorical variable.

2.2 Correlation Check

To understand the relationship between multiple variables and attributes in the dataset. The first step is to check the correlation relationship between each variable, as Correlation is used as a basic quantity for many modelling techniques, as it can also help in predicting

one attribute from another. Figure 2.3 is the correlation plot for all variables in the dataset. According to this plot, only few of the variables seem to be correlated with others. For example, the feature "interest.rate" and "revol.until" are hightly correlated, the feature "credit.policy" and the "fico" are somehow correlated. Therefore, random forest model would be used as baseline model to better understand the dataset, because unlike regression model, it would be immune to multicollinearity by nature. When they decide to split, the tree will choose only one of the perfectly correlated features[Bad19] .

2.3 Strategies to deal with imbalanced data

Classification problems in most real-world applications have imbalanced data sets. In other words, the one class examples (minority class) are a lot less than another (majority class). It is common to see that in spam detection, ads click, customer churn, etc.

From Figure 2.4, it is noticed that our data set is imbalanced. The number of positive examples is 1533 and the number of negative examples is 8045. The positive examples (people who haven't fully paid) are only 19% from the total examples. By which it is likely to cause the problem of accuracy paradox. it is very likely predicting one class regardless of the data it is asked to predict (Jason Brownlee, August 19, 2015, 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset). Therefore, accuracy is no longer a good measure of performance for different models. because if simply predict all examples to belong to the negative class, 81% accuracy would be achieved. Better metrics for imbalanced data sets are AUC (area under the ROC curve) and f1-score. However, since class imbalance influences learning algorithms during the training process by making the decision rule biased towards the majority class, Class-imbalance must be carefully handled to build a good classifier [KZ01].

In this study, the method Balanced Bagging is performed to deal with the imbalanced dataset. The idea behind the method is simple, which is similar to the balanced Random Forests. it resamples each subset of data before to train each estimator of the ensemble. In short, it combines the output of an Easy Ensemble sampler with an ensemble of classifiers

and combine the output of all classifiers [LWZ06].

2.4 Strategies to deal with missing values

In the real world, the data sets almost always have missing values. This can be due, for example, users didn't fill some part of the forms or some transformations happened while collecting and cleaning the data before they send it to you. Sometimes missing values are informative and weren't generated randomly. Therefore, it's a good practice to add binary features to check if there are missing values in each row for each feature that has missing values. [Swa18]

In the loan repayment dataset, six features have missing values, so six binary features would be added one for each feature. For example, "days.with.cr.line" feature has missing values, so we would add a feature "is.days.with.cr.line" that takes the values $\in \{0, 1\}$. However, by fitting the dataset into a baseline model and checking the feature importance, it is shown that the later added six features play no important role in the Random Forest model in Figure 2.5. Therefore, in order to keep the feature simplicity, the six later added features are removed from dataset.

Besides, to deal with the missing values in the dataset, two approaches were proposed: the first one is to drop the null values in the dataset, the second one is to impute the missing values using feature medians. By fitting both dataset into the baseline model (random forest model), it is observed in Table 2.2 that using the data with imputing method would result higher AUC score. Therefore, the data imputing with feature medians would be used for the later analysis.

Table 2.2: Baseline Model AUC Score

	AUC Score
Remove missing value	0.6544
Impute null with feature mean	0.67092

2.5 Categorical Feature Transformation

Categorical variables are known to hide and mask lots of interesting information in a data set. It's crucial to develop the methods to deal with such variables. If not, it is likely to miss out on finding the most important variables in a model.

According to the table 2.3, It looks that the dataset has only one categorical feature, "purpose". For this categorical feature, there are 6 possible categories: "credit card", "major purchase", "home improvement", "educational", "debt consolidation". Therefore, dummy variables from the feature "purpose" could be created since it is a nominal (not ordinal) categorical variable. It's also a good practice to drop the first one to avoid linear dependency between the resulted features.

Table 2.3: Data Type Table

credit.policy	int64
purpose	object
int.rate	float64
installment	float64
log.annual.inc	float64
dti	float64
fico	int64
days.with.cr.line	float64
revol.bal	int64
revol.util	float64
inq.last.6mths	float64
delinq.2yrs	float64
pub.rec	float64
not.fully.paid	int64

2.6 Data Scaling

Standardization of datasets is a common requirement for many machine learning models. It is possible that the machine learning algorithm behave badly if the individual features do not more or less look like standard normally distributed data. Therefore, in order to improve the result of the prediction model. We will standardize the data first.

The main advantage of standardization scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation [Ras14].

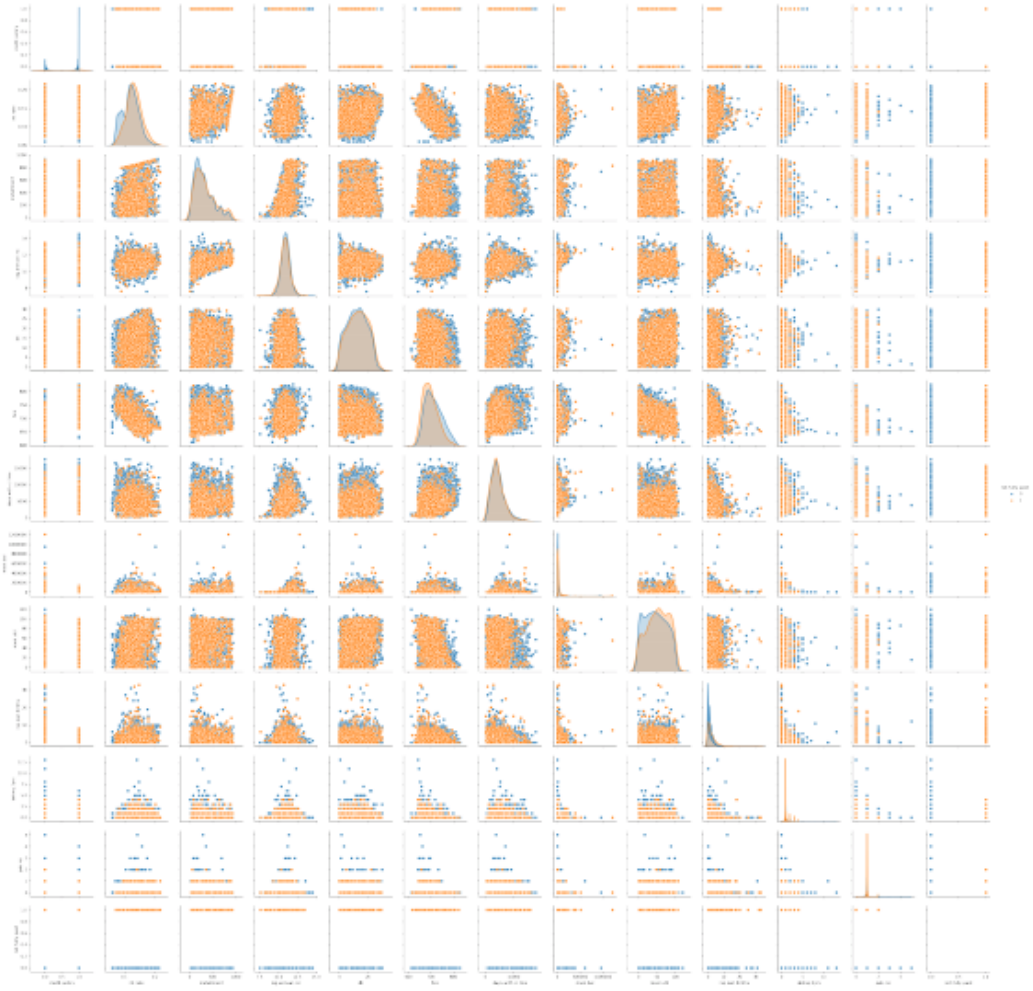

Figure 2.1: Density plot for numerical variables

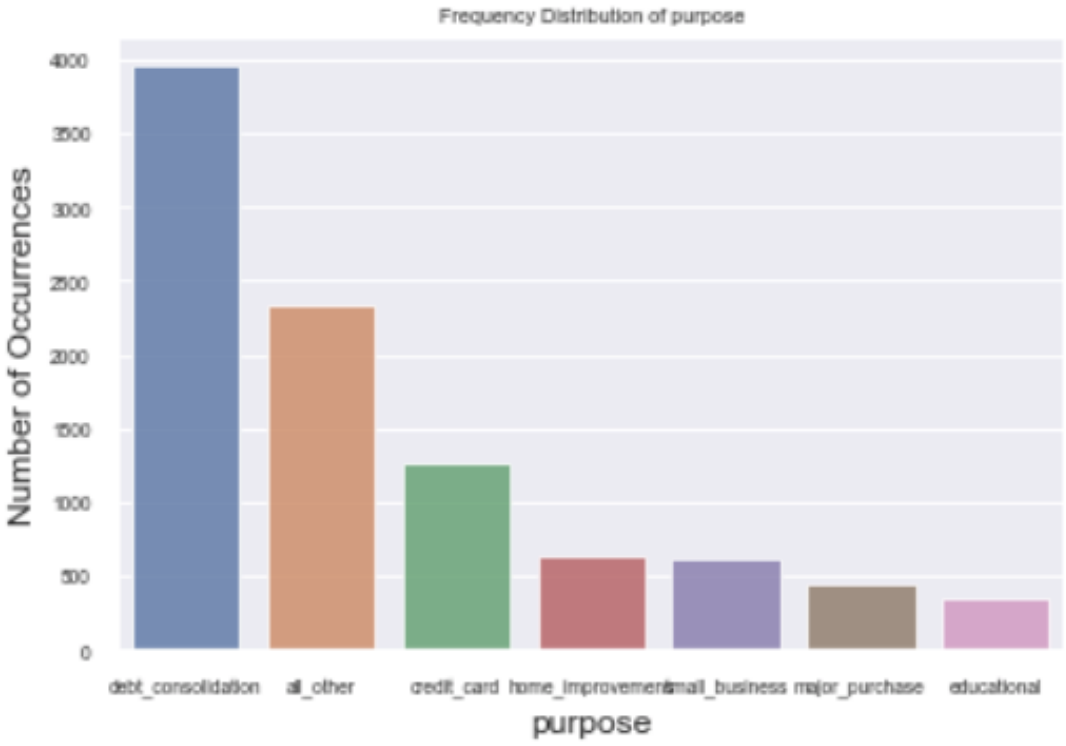

Figure 2.2: Count plot for categorical variables

Figure 2.3: Correlation Plot for all features

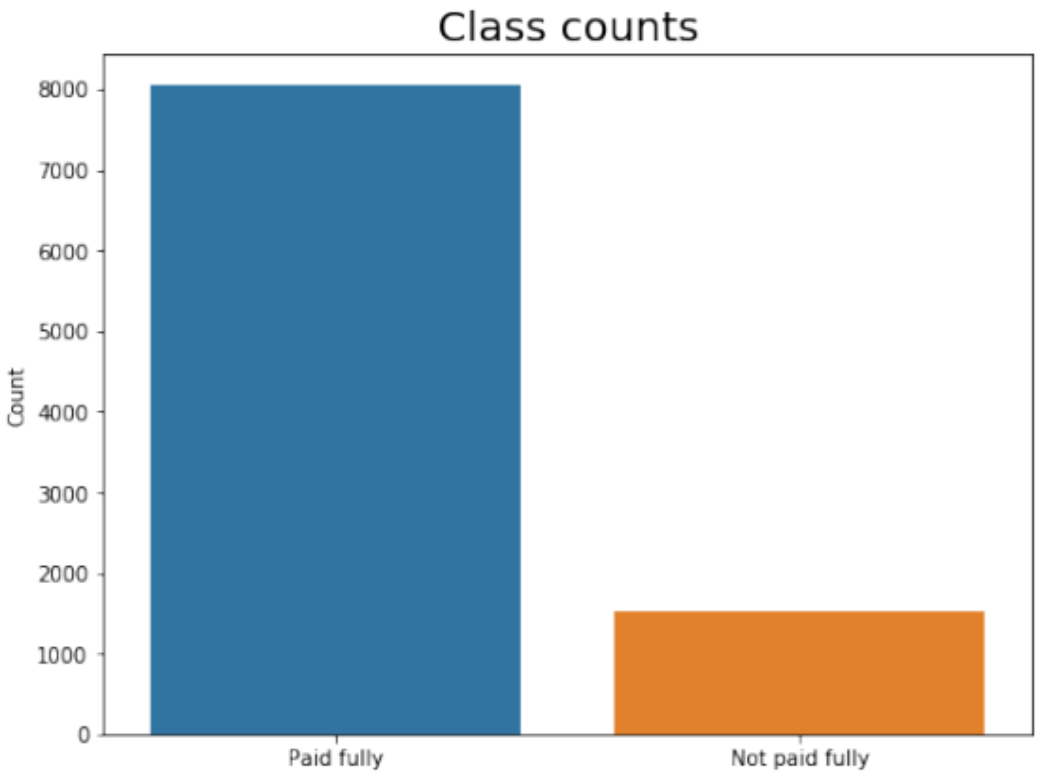

Figure 2.4: Class Counts

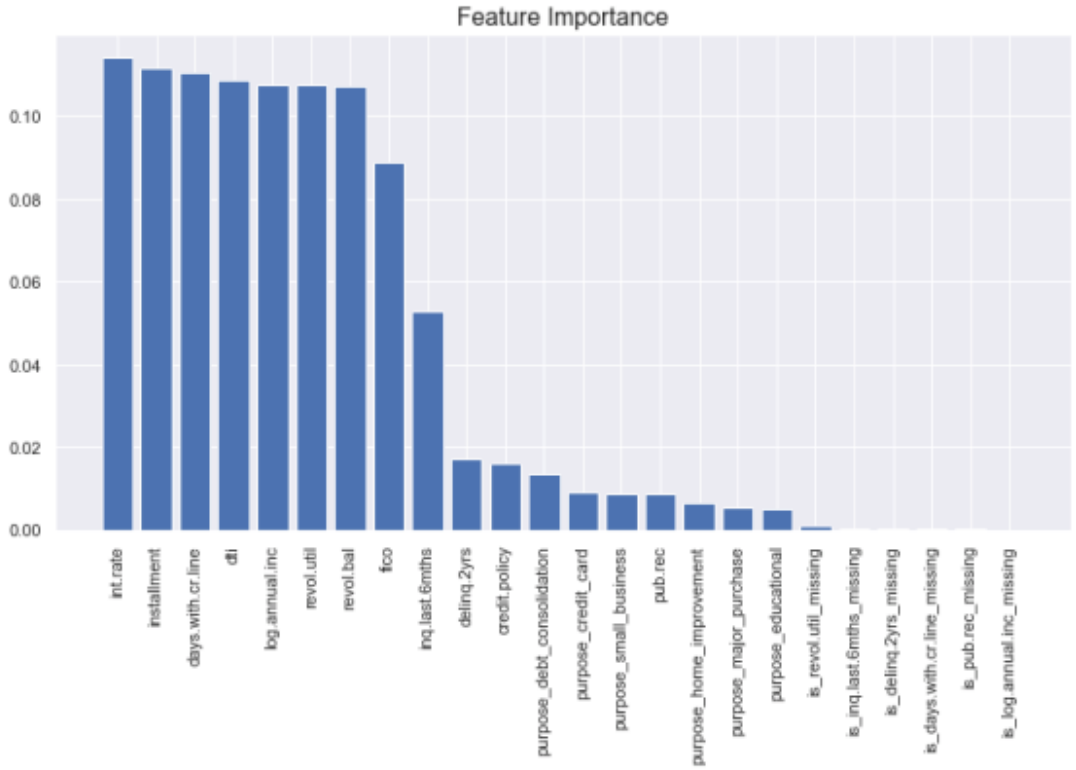

Figure 2.5: Feature Importance based on baseline Model

CHAPTER 3

Modeling

In machine learning, the classification task described above is commonly referred to as supervised learning. In supervised learning there is a specified set of classes, and example objects are labeled with the appropriate class. In this case, the goal is to identify if the lender would be able to repay the loan. In this chapter, several supervised models will be applied on the loan repayment dataset. The models include: Logistic Regression, Random Forest, KNN (K nearest neighbors), SVM (supporting vector mechine). Instead of having validation set to tune hyperparameters and evaluate different models, we'll use 10-folds cross validation because it's more reliable estimate of generalization error [Bro18].

Notice that since we are solving the classification problem with imbalanced data, not only the accuracy score will be used as the criterion to measure the performance of the model, but also, we will consider the AUC score, Precision and Recall as the performance index.

$$Precision = TruePositives/(TruePositives + FalsePositives)$$

$$Recall = TruePositives/(TruePositives + FalseNegatives)$$

3.1 Random Forest

Random Forest is a supervised learning algorithm. It is like an ensemble of decision trees with bagging method. The general idea of the bagging method is that a combination of learning models improves the overall result. The Random Forest algorithm randomly selects observations and features to build several decision trees and then averages the results. [Don18]. However, unlike decision trees, random forest prevents overfitting problem for most

of the time, as it creates several random subsets of the features and only construct smaller subtrees. In this section, we would use loan repayment data to train a random forest model.

Step 1: construct a random forest model by default using Scikit-learn package and conduct a confusion matrix to see how the model performs on the loan repayment dataset.

Figure 3.1 is the confusion matrix for the default model

Precision: 0.2682

Recall: 0.4788

AUC Score: 0.6874

Step 2: Use Randomized Search Cross validation method to determine the optimal combinations of parameter.

Firstly, we define a grid of hyperparameter ranges based on documentation on the random forest in Scikit-Learn. Secondly, we randomly sample from the grid, and perform K-Fold CV with each combination of values. The parameters include:

- n_estimators = number of trees in the forest
- max_features = max number of features considered for splitting a node
- max_depth = max number of levels in each decision tree
- min_samples_split = min number of data points placed in a node before the node is split
- min_samples_leaf = min number of data points allowed in a leaf node
- bootstrap = method for sampling data points (with or without replacement)

Step 3: Train another random forest model with the optimal parameters combination we found in step2 and evaluate the model.

Figure 3.2 is the confusion matrix for the optimized model

Precision: 0.2711

Recall: 0.5016

AUC Score: 0.6951

Comparing the confusion matrix, we can find that the optimized random forest model would have higher precision, recall and AUC score. The AUC score is increased by around 0.8%.

3.2 Logistic Regression

Logistic regression is another supervised learning algorithm that is appropriate to conduct when the dependent variable binary. It is commonly used to obtain odds ratio in the presence of more than one explanatory variable. The procedure is quite similar to linear regression, but its response variable is binomial. The result is the impact of each variable on the odds ratio of the observed event of interest. Below is the general proof of the logistic regression:

In this section, we will fit the logistic regression into the loan data. Also, we will try to determine the optimal parameter for logistic regression. The method is similar to the method in the last section.

Step 1: construct logistic regression model with regularization to avoid overfitting and conduct a confusion matrix to see how the model performs on the loan repayment dataset.

In this case, we would use ridge regression here because it enforces the β coefficients to be lower, but it does not enforce them to be zero. That is, it will not get rid of irrelevant features but rather minimize their impact on the trained model. By which, the model would tend to have more prediction power. [Cha17] Below is the Cost Function for Logistic Regression with Ridge Penalty:

$$\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

Figure 3.3 is the confusion matrix for logistic regression model by default. Based on the matrix, we could calculate for the index to evaluate the model.

Precision: 0.2477

Recall: 0.6286

AUC Score: 0.7038

Step 2: Now we would test different parameters in order to see how accuracy changes and find the optimized parameter for the ridge regression.

In ridge regression, Lambda (λ) controls the trade-off between bias and variance. In the other words, if λ is 0 or close to 0, the model will have enough power to increase its complexity by assigning big values to the weights for each parameter which will lead to overfitting problem. if we increase the value of λ, the model will tend to underfit, as the model will become too simple. In this case, we use parameter C as our regularization parameter. (Where $C = 1/\lambda$)

Figure 3.4 is the validation curve which indicates how each C value affects the accuracy of the training set and the testing set. In this case, the parameter C is not as important as we expected. As we observe form the validation curve, the change of C value won't affect the training and testing accuracy so much. Besides, the accuracy of training data and the accuracy of the testing data are very close no matter what value C chooses. The difference is only within 0.1%. Therefore, we will just consider the default model as the optimized Logistic regression model for loan repayment prediction.

Table 3.1 is the coefficient for the Logistic Regression with L2 penalty.

3.3 Support Vector Machine

A Support Vector Mechine(SVM) is also a supervised learning algorithm which used to separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which classifies new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. In multi-dimensional space, the separation of the class is a hyperplane [HCL03].

In this section, we will fit SVM model into the loan data. Also, we will try to determine

Table 3.1: Log-Reg Coefficient Table

```
Logistic Regression (L2) Coefficients
fico: -0.3513
installment: 0.2435
log.annual.inc: -0.2182
purpose_credit_card: -0.1604
not.fully.paid: -0.1596
credit.policy: -0.1457
inq.last.6mths: 0.1427
purpose_major_purchase: 0.1365
purpose_home_improvement: -0.0991
pub.rec: 0.0707
revol.bal: 0.0704
delinq.2yrs: -0.0703
revol.util: 0.0432
int.rate: 0.0417
days.with.cr.line: 0.0308
purpose_educational: 0.0222
purpose_debt_consolidation: 0.0091
dti: -0.0029
```

the optimal parameter of the model and evaluate the SVM model. The method is similar with the method that we use for the random forest model.

One thing we need to notice is that we are using RBF kernel for the SVM model in this case because, unlike linear kernel, it can handle the situation when the non-linear relationship between the class labels and attributes. In addition, contrast to poly kernel, the number of hyperparameters in RBF kernel is easier to control which reduces the model complexity.

To optimize the SVC model, we are using Grid-Search Cross validation. There are parameters for RBF kernel: C and γ

- C: it is like a regularization parameter, which tells the SVM optimization how much you want to avoid misclassifying each training example. For example, large values of C will result the optimization choosing a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.

- Gamma (γ): tells how far the influence of a single training example would reach. High

Gamma only consider the nearby points of the separation line; and low Gamma would also consider the far away points of the separation line.

Table 3.2 is the result of Grid-Search Cross Validation, the table shows the optimal combination of the parameters for SVM model.

Table 3.2: Best Parameter for Optimized SVM

Parameter	Value
C	10
γ	0.01

Figure 3.5 is the confusion matrix for SVM model by default.

Precision = 0.2609

Recall = 0.5635

AUC Score = 0.6736

Figure 3.6 is the confusion matrix for the optimized SVM model.

Precision = 0.2577

Recall = 0.5700

AUC Score = 0.6799

As we observed from the result of two models, we noticed that the precision rate drops in the exchange of the increase of recall and AUC score. In this case, since the cost that we miss classify ineligible loaner is much higher than the cost we miss classify the eligible loaner. Therefore, we would prefer the model with higher recall score.

3.4 K-Nearest-Neighbors Model

The k-nearest neighbors algorithm (KNN) is a non-parametric method that can be used for classification and regression problems. In classification problems, an object is classified by a vote of its neighbors, with the object being assigned to the class most common among its k

nearest neighbors [Alt92].

The prediction accuracy based on the k-NN model is highly contingent on the value of K. The best choice of K depends upon the data. Usually, larger values of K would reduce the effect of the noise on the classification but make boundaries between each category less distinct. Smaller value of K would reduce the error rate for the training sample, but it would cause the overfitting problem [ELL11]. In this section, we attempt to identify the optimal K value (from 1 to 30) and evaluate the model.

Figure 3.7 is the Error rate vs K value for the loan repayment data set. as we observe from the plot, we may notice that when $K = 1$, the error rate tends to be zero. As K value increases, the error rate would experience a significant increase first, and then it tends to have a slight drop and becomes relatively stable when $K > 10$.

Figure 3.8 is validation curve vs K value for the data set. It indicates that as K increases, the train accuracy will decrease, and the testing accuracy will increase. When $K > 10$, the training and testing accuracy tends to be very close, and they are converging as K increases. Therefore, in this case we would choose $K = 10$ to train the KNN model.

Figure 3.9 is the confusion matrix for the K-nearest-neighbors model when K = 10.

Precision: 0.2261

Recall: 0.5928

AUC Score: 0.6492

3.5 Feature Importance Analysis

Feature Selection methods is a way to efficient way to reduce the dimensions without much loss of the total information. It also helps to make sense of the features and its importance. In this section, we will check feature importance by constructing a gradient boosting model[Asa18].

Figure 3.10 is the partial dependence plots to see what the most important features are and their relationships with whether the borrower will most likely pay the loan in full before

mature data. Note that only the top 8 features were plotted to make it easier to read. According to Figure 3.10, we noticed that fico score significantly affects the status whether the loaner will be able to repay the loan. The borrowers with higher fico score are much less likely to fail to fully repay the loan.

The partial dependence plot indicates that the "log.annual.inc", "installment", "credit.policy", "inq.last.6mths", "revol.bal" as important features which is consistent with findings in logistic regression model(Table 3.1). The "int.rate" is the factor that seems to be more important in the gradient boosting model than that in the logistic regression. This happened because the borrowers with higher interest rates are considered risker, which are less likely to repay the loan.

3.6 Model Comparison

In this section, the result of the selected models will be analyzed and compared. Figure 3.11 is the chart for the ROC Curve and PR Curve For selected models. As we observed from the ROC curve chart, we would notice that the ROC curves of Logistic Regression, Random Forest, and SVM models are relatively closes to the left-top of the plot than the roc curve of KNN model. And based on the results from previous analysis, we would notice that Logistic Regression model has the highest AUC Score of 0.70185 and the highest Recall rate of 0.6286.

In addition, Random Forest Model has the highest accuracy score of 0.70407 and the highest Precision rate of 0.2711. In this case, the cost that we misclassify the loaner who is ineligible to repay is much higher, so we would prefer to use the model that would result higher recall rate. Therefore, in this analysis, Logistic Regression model is preferred among the selected models.

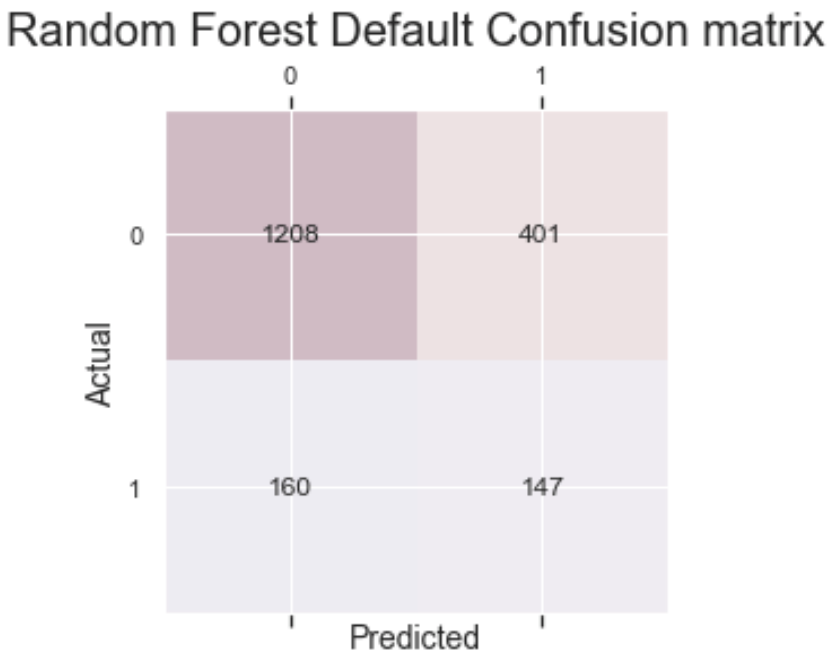

Figure 3.1: Confusion matrix for default Random Forest Model

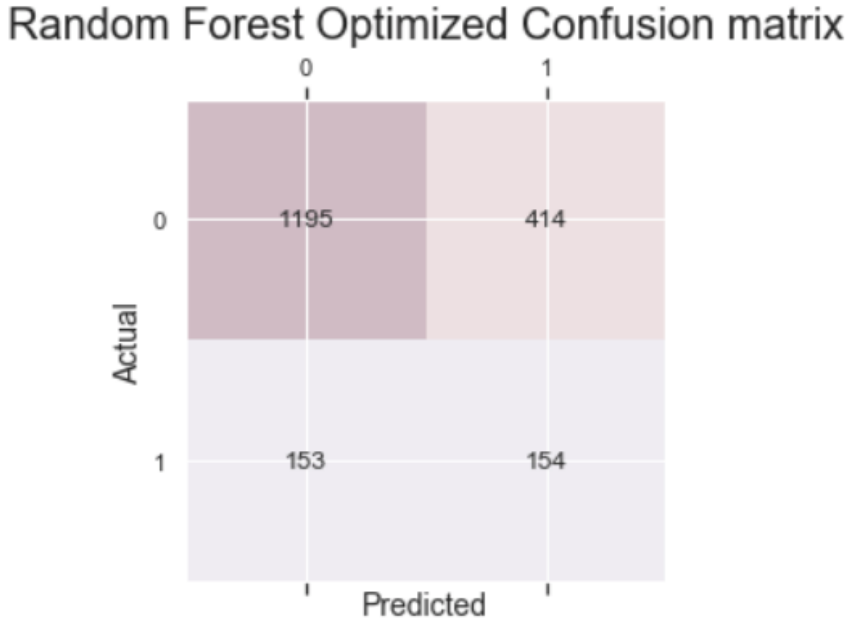

Figure 3.2: Confusion matrix for optimized Random Forest Model

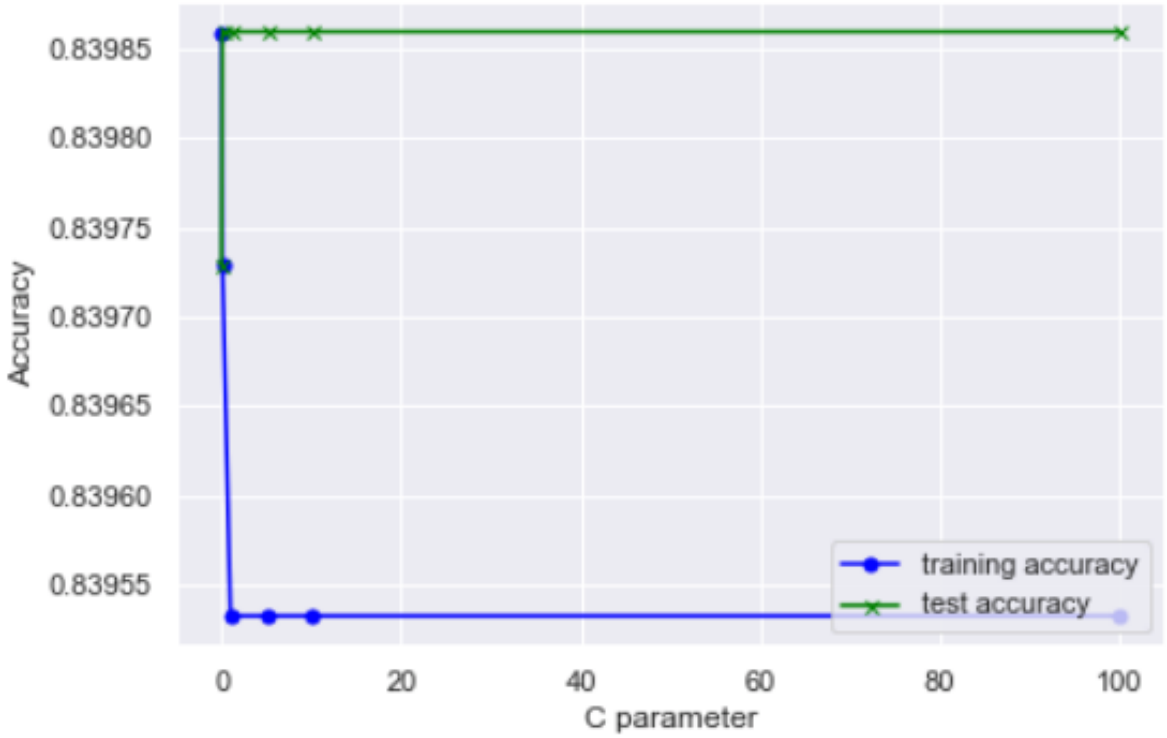

Figure 3.3: Validation Curve with Logistic Regression With L2 penalty

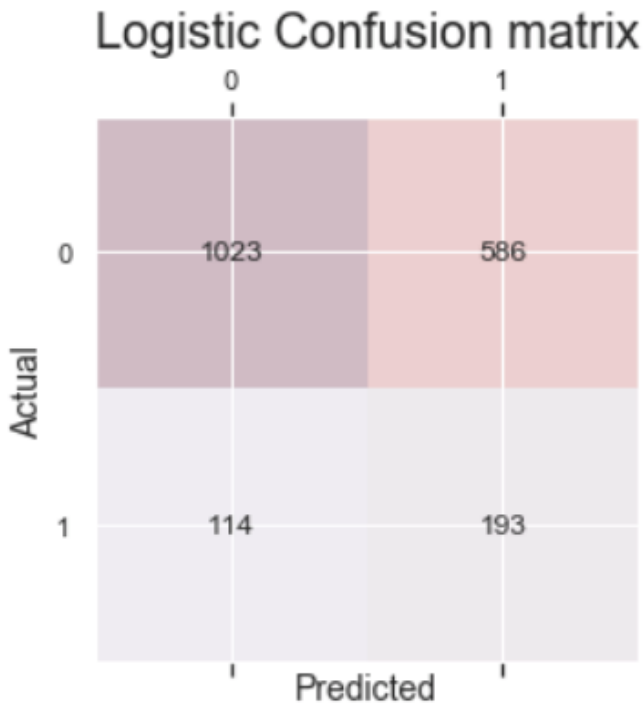

Figure 3.4: Confusion Matrix for Logistic Regression

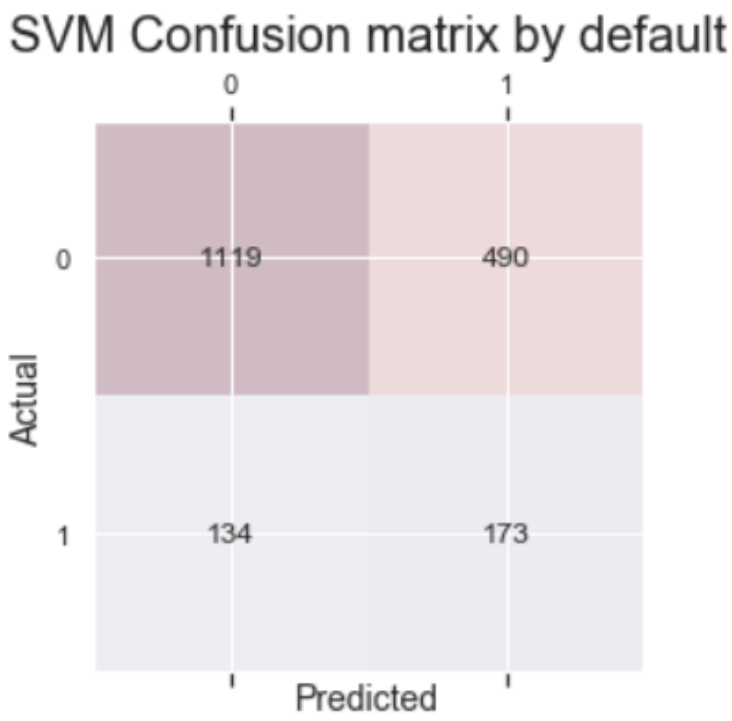

Figure 3.5: Confusion Matrix for Default SVM model

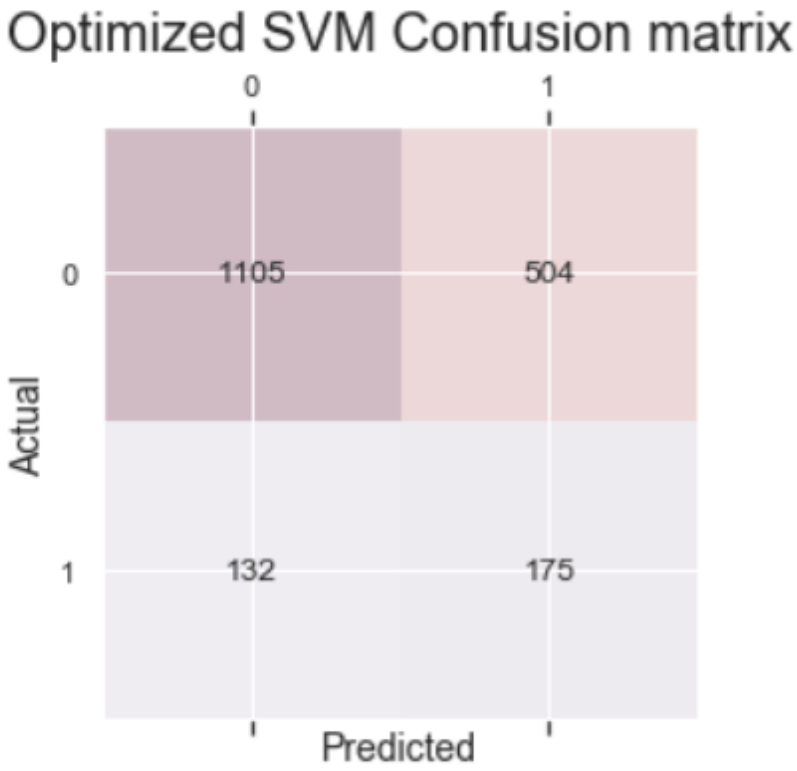

Figure 3.6: Confusion Matrix for Optimized SVM model

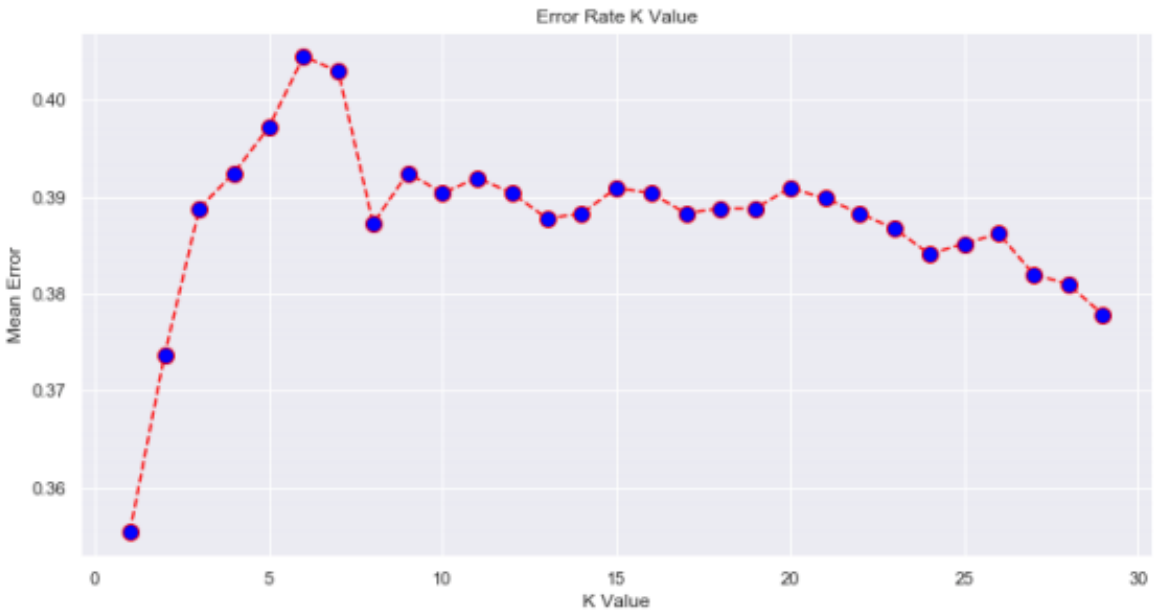

Figure 3.7: Error Rate vs K-value

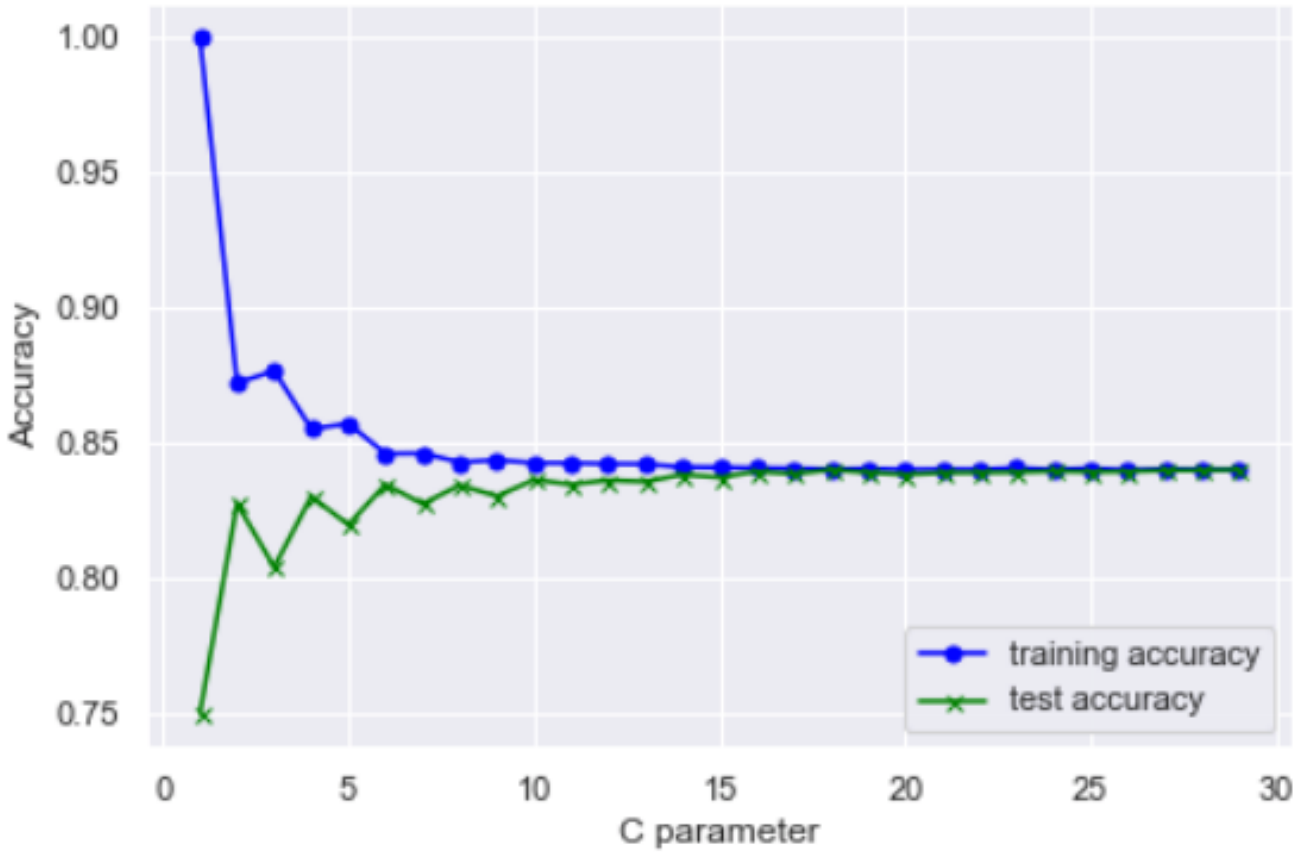

Figure 3.8: Validation Curve with KNN

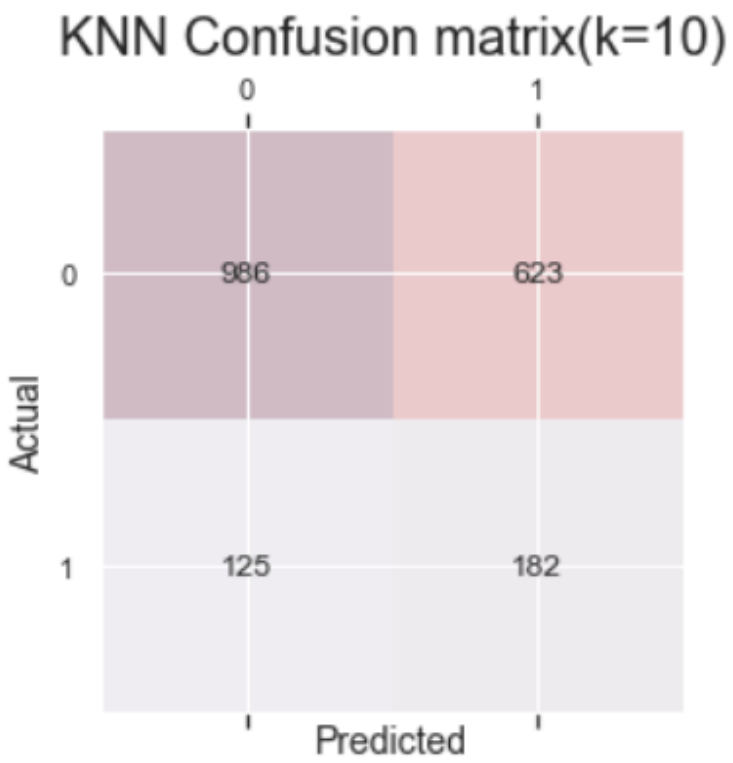

Figure 3.9: Confusion Matrix for KNN model

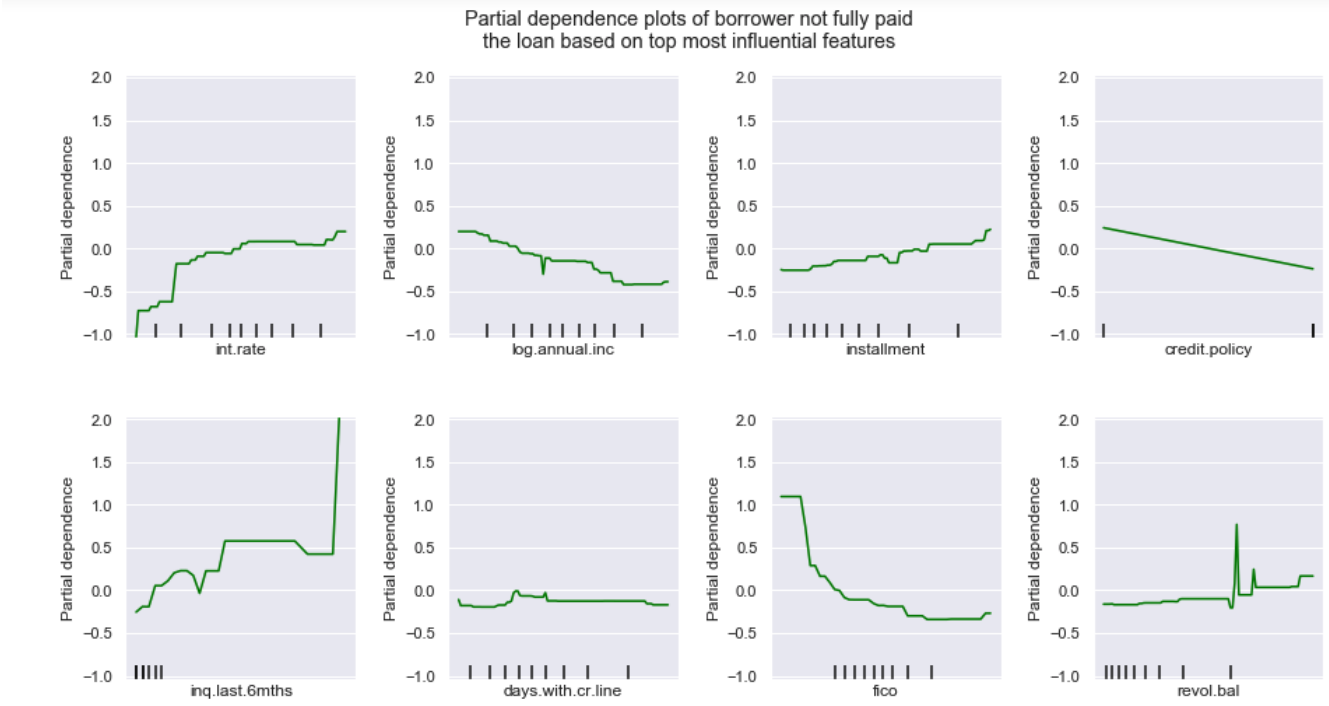

Figure 3.10: partial dependence plots

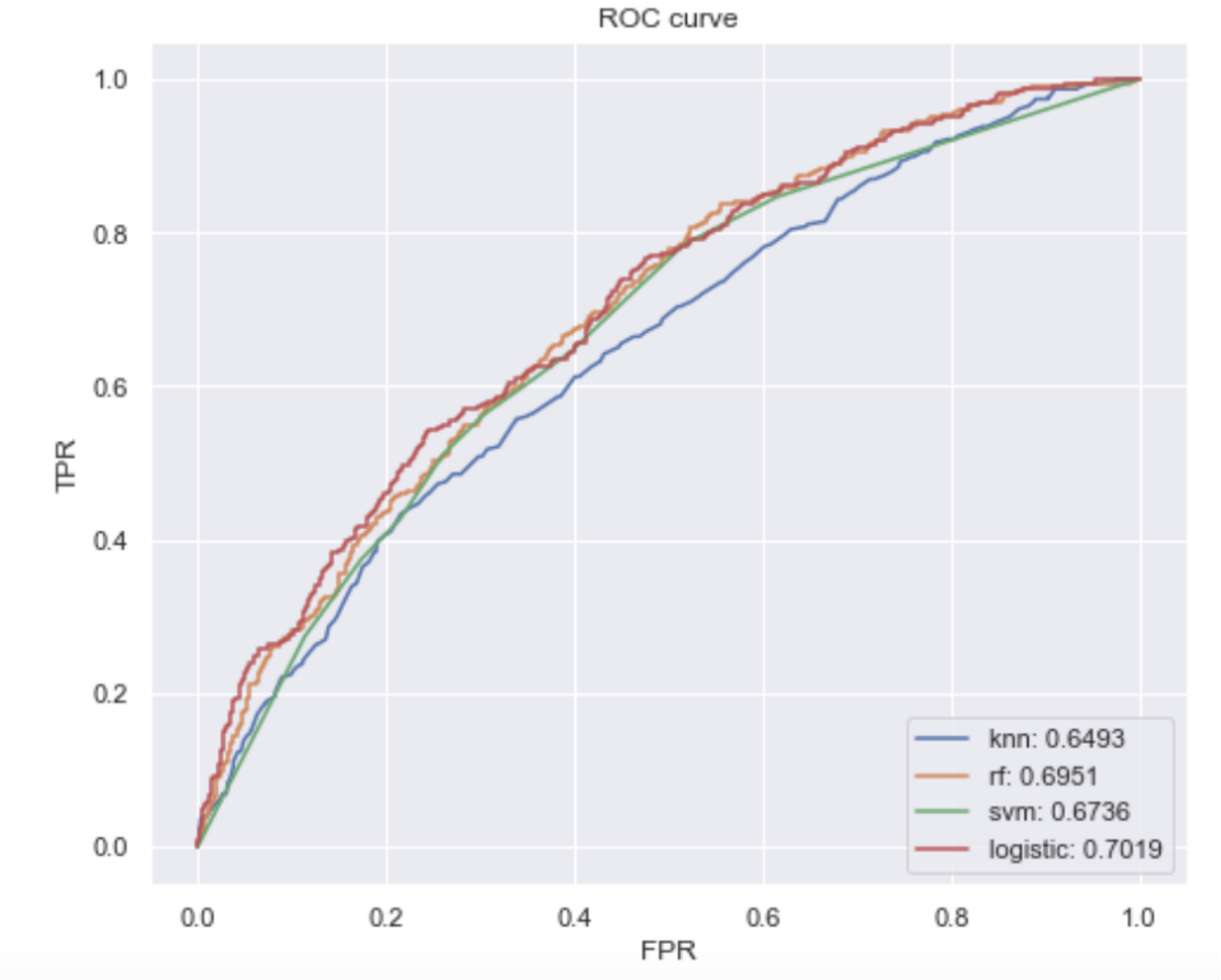

Figure 3.11: ROC Curve For selected models

CHAPTER 4

Enhancement and Conclusion

4.1 Future enhancement

In this study, there are several enhancements that we could make in the future. For example, outlier problem is not considered in the exploratory data analysis. if there are outliers in the dataset, the results of the predictive model will not be as valid as they are. In addition, the deep learning algorithm method should also be implemented when predicting for the loan the repayment status. Besides, if we would have a larger dataset, we would have more training samples. By which, it might help fix the high variance problem and make our analysis more valid.

4.2 Conclusion

Nowadays, the loan business becomes more and popular, and many people apply for loans for various reasons. However, there are cases where people do not repay the bulk of the loan amount to the bank which results in huge financial loss. Hence, if there is a way that can efficiently classify the loaners in advance, it would greatly prevent the financial loss.

In this study, the dataset was cleaned first, and the exploratory data analysis and feature engineering were performed. The strategies to deal with both missing values and imbalanced data sets were covered. Then we propose four machine learning models to predict if the applicant could repay the loan, which are Random Forest, Logistic Regression, Support Vector Machine, and K-Nearest Neighbors. When tuning parameters, both Randomized Search Cross Validation and Grid Search Cross Validation methods are applied in different

situations. Through experiments, it is noticed that the model was found which best fits the dataset with highest accuracy is the random forest model, and the model with highest AUC score is Logistic Regression with L2 penalty.

As we expected, borrowers with higher annual income and higher FICO scores are more likely to repay the loan fully; In addition, borrowers with lower interest rates and smaller installments are more likely to pay the loan fully.

REFERENCES

[Alt92] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression." *The American Statistician*, **46**(3):175–185, 1992.

[Asa18] (Sudharsan Asaithambi. "Why, How and When to apply Feature Selection." Jan 2018.

[Bad19] Will Badr. "Why Feature Correlation Matters .... A Lot!" *Towards Data Science*, Jan 18, 2019.

[Bha18] Abhishek Bhagat et al. *Predicting Loan Defaults using Machine Learning Techniques*. PhD thesis, California State University, Northridge, 2018.

[Bro18] Jason Brownlee. "A gentle introduction to k-fold cross-validation." *Accessed October*, **7**:2018, 2018.

[Cha17] Ofir Chakon. "PRACTICAL MACHINE LEARNING: RIDGE REGRESSION VS.LASSO." August 2017.

[Don18] Niklas Donges. "The Random Forest Algorithm." *Statistical Methods*, Feb 2018.

[ELL11] Brian S Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. "Miscellaneous clustering methods." *Cluster Analysis, 5th Edition, John Wiley & Sons, Ltd, Chichester, UK*, 2011.

[HCL03] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. "A practical guide to support vector classification." 2003.

[Hon18] Hongri. "Jia Bank Loan Default Prediction with Machine Learning." pp. 137–163, Apr 10, 2018.

[KZ01] Gary King and Langche Zeng. "Logistic regression in rare events data." *Political analysis*, **9**(2):137–163, 2001.

[LWZ06] Xu Ying Liu, Jianxin Wu, and Zhi Hua Zhou. "Exploratory Under-Sampling for Class-Imbalance Learning." In *International Conference on Data Mining*, 2006.

[PLI02] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. "An introduction to logistic regression analysis and reporting." *The journal of educational research*, **96**(1):3–14, 2002.

[Ras14] Sebastian Raschka. "About feature scaling and normalization." *Sebastian Racha. Disques, nd Web. Dec*, 2014.

[Swa18] Alvira Swalin. "How to handle missing value." *Towards Data Science*, 2018.