Research on Classification and Recognition of Credit Status of Loan Users on the Online Lending Platform

Group 4: Xiaofan Wang (NUID: 001302984), Haoxuan Qiu (NUID: 001054159)

Executive Summary: In this project, we used the dataset from Lending Club and selected features that are highly relevant to our project according to the variables interpretation of LC Directory. After data preprocessing, we implemented several common algorithms to predict the target variable "loan_status" whether belongs to "Fully Paid" or "Charged Off", and then comparing their performances to select the best one, logistic regression model. Thus, we could provide better recommendation for investors and help them reduce the loan risks and maximize their profits.

I. Background and Introduction

- **The Background and The Goal**

  In the financial market, online lending platforms have played an important role in reducing costs, increasing revenue, and further improving profitability. Borrowers can obtain loans within a short period of time with low interest rates through credit verification. However, this mechanism may also bring some adverse consequences, such as delay or non-payment by the borrower. Therefore, if the investor wants to make more profits from the interest of loan, it is necessary to strictly identify and pre-judge the credit status of the loan users through online lending platform, and ensure that the borrower has the ability to repay before providing the loan to the borrower, so as to decide whether to agree borrower's loan application.

- **The Possible Solution**

  This research takes the credit status of loan users as the target variable, and divides them into two categories "Fully Paid" and "Charged Off". Given that this study is a binary classification problem, several different algorithms can be built and applied so as to identify and predict the credit status of loan users more accurately and help online lending platforms reduce loan risk and make more profits.

II. Data Exploration and Visualization

- **Data Source and Basic Information**

  This research uses complete loan dataset for all loans issued through 2008 to 2019, including the loan status and relevant payment information. The original file is a matrix of about 2.26 million observations and 145 variables, which was collected by the Lending Club and a data dictionary is also provided in a separate file, which contains detailed explanation of all variables in the dataset. (https://www.lendingclub.com/loans).

- **Target Variable: Loan Status**

  The loan dataset consists of loans in various statuses so this study started with exploring their counts and percentages of different loan statuses, the table shown as below.

Table 2.1 Counts and Percentages of Different Loan Statuses

Loan Status	Count	Percentage
Charged Off	241206	11.4160%
Current	850637	40.2599%
Default	28	0.0013%
Does not meet the credit policy. Status: Charged Off	745	0.0353%
Does not meet the credit policy. Status: Fully Paid	1965	0.0930%
Fully Paid	986651	46.6972%
In Grace Period	8375	0.3963%
Late (16-30 days)	3305	0.1564%
Late (31-120 days)	19954	0.9444%

From Table 2.1, it is clear that there are nine different loan statuses, but in actual situations, it just needs to be classified into two categories: “Charged Off (0)”, “Fully Paid (1)”, which can identify whether the loan users can pay off the entire loan within the prescribed time, as the figure shown below.

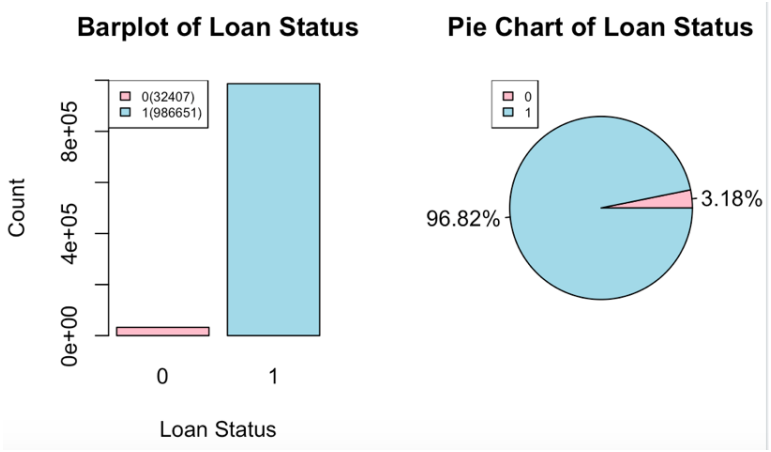

Figure 2.1 Bar Plot and Pie Chart of Loan Status

The Figure 2.1 shows that most of loan users can pay off all the loans and the percentage accounts for 96.86%, which was obviously larger than that of defaulted loan users, only with 3.18%. However, this phenomenon exposes the problem of imbalanced categories extremely distributed in this sample. And this problem will be solved by using "oversample" in the next step.

- **Interest Rate and Grade**

In order to observe how useful these variables would be for credit risk modelling. It is necessarily to compare the interest rates of different grades. It is known that the better the grade the lowest the interest rate. We can nicely visualize this with boxplots, as shown in the Figure 2.2.

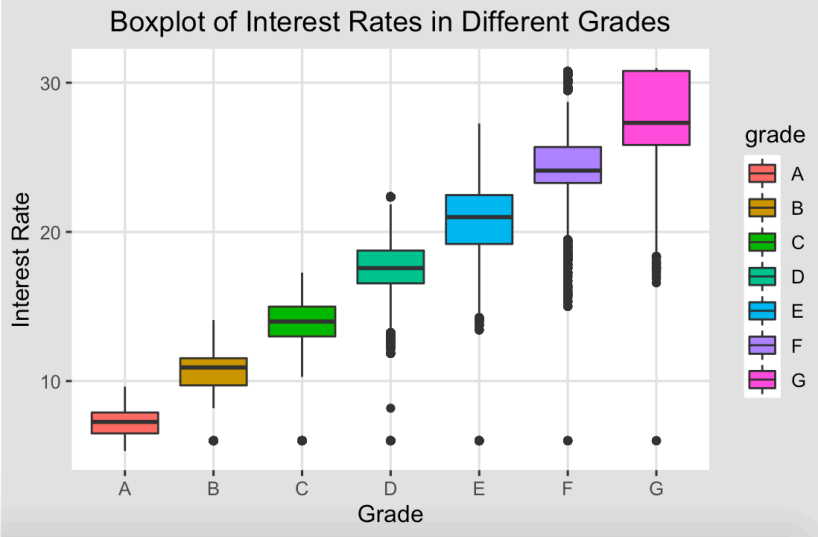

Figure 2.2 Boxplot of Interest Rates in Different Grades

- **Total Loan Amount in Different States of US**

In order to efficiently allocate financial resources to the different places in the United States and build the appropriate links between the borrowers and investors, the distribution of total loan amount could be computed and visualized, as shown in the Figure 2.3.

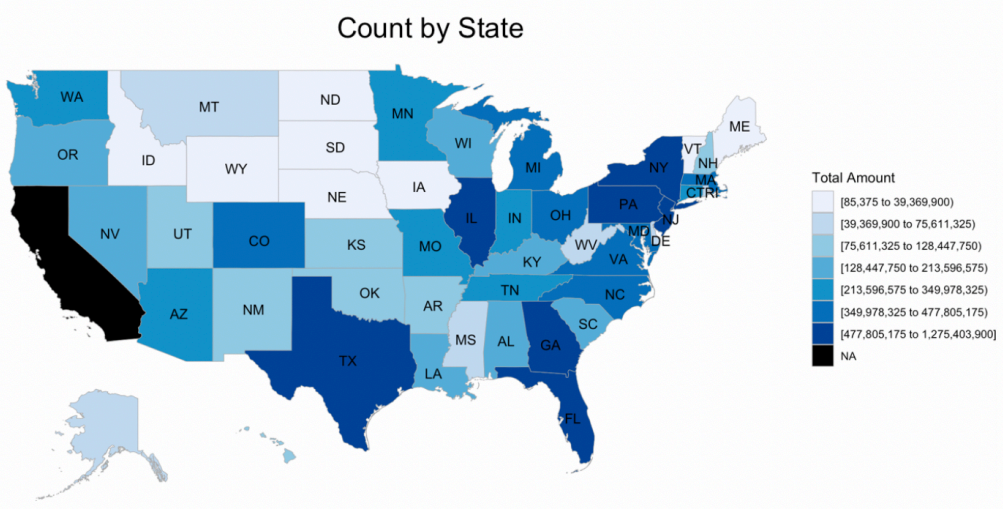

Figure 2.3 Distribution of Total Loan Amount in Each States

The Figure 2.3 shows that total loan amount of those states (TX, GA, FL, NY, PA, IL, NJ) reached the largest with the deepest blue. And the color of each state is different, from deep to light based on the different loan amounts (from high to low). Therefore, it is reasonable for online lending platforms to allocate more resources and build more links between investors and borrowers in the states with higher requirement of loan. In addition, it is also important to improve the supervision of loan risk in these states.

III. Data Preparation and Preprocessing

- **Dimension Reduction and Feature Selection**

According to the detailed explanation of all variables in the data dictionary, this study selected nine key features and omit their missing values. The features can be explained as below:

(1) loan_status
The variable “loan_status” represents current status of loan and was regarded as target variable, which contains nine different levels of credit status of loan users. (e.g. “Charged off”, “Default”, “Fully paid”…)

(2) loan_amnt
The variable “loan_amnt” represents total amount of the loan taken by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.

(3) int_rate
The variable “int_rate” represents interest rate on the loan.

(4) grade
The variable "grade" represents loan grade assigned by the Lending Club.

(5) emp_length
The variable "emp_length" represents employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.

(6) home_ownership
The home ownership status provided by the borrower during registration or obtained from the credit report. The values are: RENT, OWN, MORTGAGE, OTHER. And "NONE" and "ANY" has been removed.

(7) annual_inc
The variable "annual_inc" represents the self-reported annual income provided by the borrower during registration.

(8) term
The variable "term" represents the number of payments on the loan. Values are in months and can be either 36 or 60.

(9) region
The variable "region" represents the loan location of the borrowers.

- **Variable Converting and Normalization**

The variables we selected have different data types (like: character, time date), which should be converted into the numeric type in order for better computation. (1) The data type of "loan_status" is character, we replace the status "Fully Paid" by "1" and "Charged Off" by "0" and discard irrelevant observations. (2) The variable "Grade" is sequential variable, we could replace "A, B, C…" by numeric elements "1, 2, 3…". (3) The variable "Home_ownership" is categorical variable, including four categories: mortgage, rent, own, others. We converted them into four dummy variables: home_ownership_mortgage, home_ownership_rent, home_ownership_own, home_ownership_others. If the record belongs to a certain category among them, R coding set 1 into the blank, if not, set 0. (4) The variable "Term" have two different time periods, 36 months and 60 months, we converted them into two numbers, 36 and 60. After converting variables into numeric type, we normalize them to reduce data redundancy and improve data integrity to achieve more accurate computation.

- **Correlation Analysis and Check multicollinearity**

  Multicollinearity refers to a situation in which two or more explanatory variables in a multiple regression model are highly linearly related. In order for more accurate computation, we check multicollinearity and discard highly-correlated variables. As the Figure 3.1 shown to us, the variables “int_rate” and “grade’ are highly positive correlated and the variables “home_ownership_mortgage” and “home_ownership_rent” are highly negative correlated, therefore, we dropped the variables “home_ownership_rent” and “grade”.

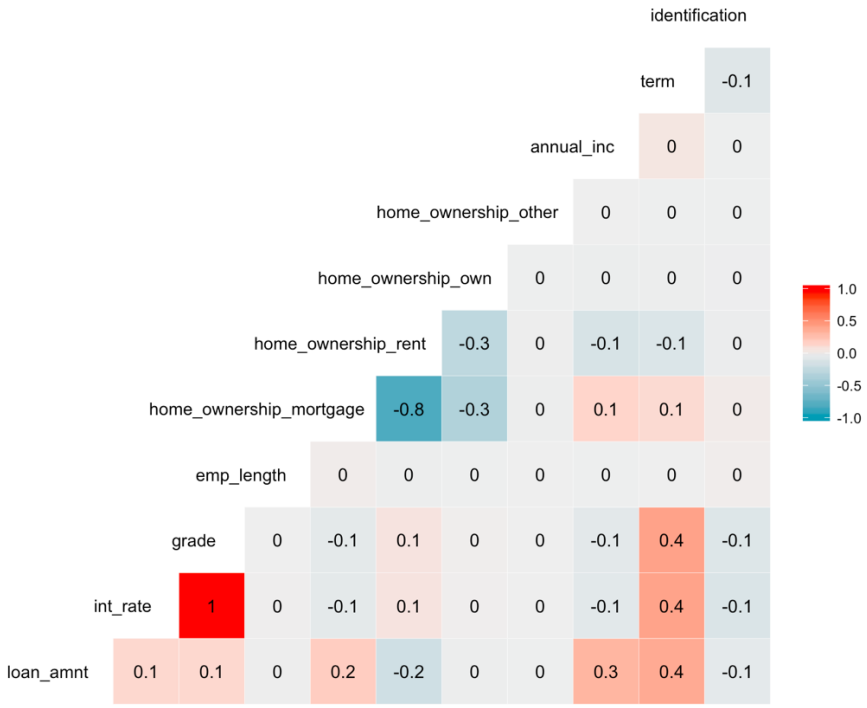

Figure 3.1 The Correlation of Variables Related to Loan Status

- **Imbalance Datasets and Over-sampling**

  Suppose that we have created a model of predicting whether a loan user is default or not. The best classifier can be selected and we train it on the dataset, the accuracy of the model may be very high. However, this model will not find any defaulted loan users if it will be used in real situations due to the imbalanced dataset. One common way to tackle the issue of imbalanced data is over-sampling. Over-sampling refers to various methods that aim to increase the number of instances from the underrepresented class in the data set. In this study, this technique will increase the number of fraudulent transactions in our data and then we obtained the train set (The number of fraud and non-fraud is 109333 and 690667).

IV. Data Mining Techniques and Implementation

- **Classification**

  Classification is one of the most popular and commonly used data mining tools in real life situations. Classification includes two important steps: 1) training the classifier, and 2) testing the classifier. The first step is learning phase where a classifier learns patterns and correlations from training data. In the second step, the classifier is tested with new data where the class label is unknown. Therefore, classification is data mining tool where our model predicts to which class new observation belongs based on previous discovered patterns in training data. In this study, we will use several different classifiers to predict whether the loan users are default or not.

- **Data Mining Techniques**

1. KNN

K-Nearest Neighbors algorithm is a non-parametric approach that classifies new cases based on the similarity measures (with regard to distance functions). In this study, K-Nearest Neighbors will be used to calculate the likelihood if a loan user will be a defaulter or not.

2. Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules. Beginning at the root node, the algorithm splits the data on the feature that results in the greatest information gain (or entropy). Iteratively, we can repeat this splitting at each child node until the leaves are pure.

3. Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Mathematically, logistic regression estimates a multiple linear regression function defined as:

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = (\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)$$

With real constants $\beta_0, \beta_1, \ldots, \beta_n$. The logit model can be estimated via maximum likelihood estimation using numerical methods as we will do in R.

4. SVM

Support vector machines (SVM) use a mechanism called kernels, which essentially calculate distance between two observations. The SVM algorithm then finds a decision boundary that maximizes the distance between the closest members of separate classes.

5. **Neural Network**
   Predicting the class label using neural networks through attribute relevance analysis is presented in this study. This method has the advantage that the number of units required can be reduced so that we can increase the speed of neural network technique for predicting the class label of the new tuples.

6. **Naïve Bayes**
   Naive Bayes (NB) is a very simple algorithm based around conditional probability and counting.

7. **Linear Discriminant Analysis**

   The idea was to find a linear combination of features that are able to separate two or more classes.

- **The Flowchart of Algorithms Implementation**

Figure 4.1 The Flowchart of Algorithms Implementation

V. Performance Evaluation

- **Performance Comparison (Confusion Matrix)**

  After we have analyzed data, we can conclude which features we will include in the model. Therefore, the model will have the following 9 independent variables: "loan_amnt (x_1)", "int_rate (x_2)", "emp_length (x_3)", "home_ownership_mortgage (x_4)", "home_ownership_own (x_5)", "home_ownership_other (x_6)", "annual_inc (x_7)", "term (x_8)", while our dependent variable is "loan_status (y)".

  Before building the model, we split our data into train set (70%) and test set (30%). First, we are going to train the classifier on the training set. Secondly, we are going to test how good our classifier is on the test set where label class is unknown. The best way to split our data into training and test set is by cross-validation method that uses multiple test sets. Then, we use four main performance indexes of confusion Matrix to compare different algorithms, the results as following:

Table 5.1. Performance Comparison of Different Algorithms

Algorithm	Accuracy	Precision	Recall
KNN	85.3%	5.4%	44.4%
Decision Tree	89.7%	9.9%	45.5%
Logistic Regression	96.6%	2.2%	18.7%
SVM	80.1%	6.8%	48.4%
Neural Network	87.4%	7.3%	45.1%
Naïve Bayes	83.2%	8.6%	36.9%
Linear Discriminant Analysis	82.8%	5.2%	47.4%

Based on the above table, even though Decision Tree had shown a relatively good result in training, but it is Logistic Regression that out-performs the other models in test. Therefore, we construct our model by logistic regression algorithm.

- **ROC Curve for Logistic Regression Model**

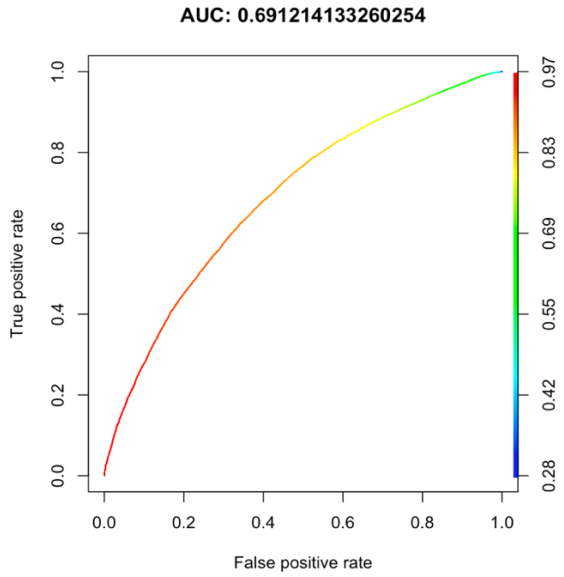

Figure 5.1 ROC Curve for Logistic Regression Model

VI. Discussion and Recommendation

- **Discussion of Algorithms**

1. **KNN**

Advantages: (1) Simple to implement. (2) Flexible to feature choices. (3) Naturally handles multi-class cases. (4) Can do well in practice with enough representative data.

Disadvantages: (1) Large search problem to find nearest neighbors. (2) Storage of data. (3) Must know we have a meaningful distance function.

2. **Decision Tree**

   Advantages: (1) Simple to understand, interpret, visualize. (2) Decision trees implicitly perform variable screening or feature selection. (3) The internal workings are capable of being observed and thus make it possible to reproduce work. (4) Can handle both numerical and categorical data. Performs well on large datasets and extremely fast.

   Disadvantages: (1) Overfitting: Decision-tree learners can create over-complex trees that do not generalize the data well. (2) Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called variance, which needs to be lowered by methods like bagging and boosting. (3) Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.

3. **Logistic regression**

   Advantages: (1) Outputs have a nice probabilistic interpretation, and the algorithm can be regularized to avoid over-fitting. (2) Logistic models can be updated easily with new data using stochastic gradient descent.

   Disadvantages: (1) Logistic regression tends to underperform when there are multiple or non-linear decision boundaries. (2) They are not flexible enough to naturally capture more complex relationships.

4. **SVM**

   Advantages: SVM algorithm can model non-linear decision boundaries, and there are many kernels to choose from. They are also fairly robust against overfitting, especially in high-dimensional space.

   Disadvantages: Memory intensive, trickier to tune due to the importance of picking the right kernel, and don't scale well to larger datasets.

5. **Neural Network**

   Advantages: (1) No need for feature engineering. (2) Best results with unstructured data. (3) No need for labeling of data. (4) Efficient at delivering high-quality results.

   Disadvantages: (1) The need for lots of data. (2) Black Box. (3) Overfitting the model. (4) Lack of flexibility.

6. **Naïve Bayes**

   Advantages: Even though the conditional independence assumption rarely holds true, NB models actually perform surprisingly well in practice, especially for how simple they are. They are easy to implement and can scale with dataset.

Disadvantages: Due to the sheer simplicity, NB models are often beaten by models properly trained and tuned using the previous algorithms listed.

7. **Linear Discriminant Analysis**

Advantages: Classification functions useful for profiling: can order predictors in terms of separating the classes.

Disadvantages: (1) Not suitable for large datasets. (2) Sensitive to outliers

- **Recommendation**

It is clear that these models are not perfect and they have fairly poor performance. There are other methods that can be used for classification. In addition, a new popular method is survival analysis where probabilities of default change over time, and features that change over time can be included. These new methods would be interesting to conduct in future projects. In addition, the selection of cut-off value (discrimination threshold) is very important because if investors are going to use this model to decide on which loan to invest or not, the choice of cut-off value will determine which applicants will get a loan and which not.

VII. Summary

In this project, we have developed several common algorithms: KNN, Decision Tree, Logistic Regression, SVM, Neural Network, Naïve Bayes, and Linear Discriminant Analysis. After comparing their performance, it can be concluded that their values of accuracy are almost similar. Best model has been chosen based on the results of confusion matrix, it is logistic regression model. It would help the investor have a better understanding of the credit of loan users, which help them to reduce the risk of loaning and increase their interests and inform their future investment decisions on who to give a credit to and what credit limit to provide.

Appendix: R Code for use case study

```
loan<-read.csv(file='~/Desktop/loan/loan.csv', header=TRUE)
dim(loan)
colnames(loan)

install.packages("dplyr")
library(dplyr)
library(ggthemes)
library(ggplot2)

# feature selection
loan_df<-loan %>%
  select(loan_status, loan_amnt, int_rate, grade, emp_length,
         home_ownership, annual_inc, term, addr_state, issue_d)

# missing values
sapply(loan, function(x) sum(is.na(x)))

# remove 4 rows with missing annual income,
# remove 49 rows where home ownership is 'NONE' or 'ANY'
# remove rows where emp_length is 'n/a'.
loan_df<-loan_df %>%
  filter(!is.na(annual_inc),
         !(home_ownership %in% c('NONE','ANY')),
         emp_length!='n/a')

# loan_status
loan_status<-loan_df %>%
  count(loan_status)
ggplot(data = loan_status, aes(x=reorder(loan_status, -desc(n)), y=n, fill=n))+
  geom_col()+labs(x='Loan Status', y='Count')+coord_flip()

loan_df %>%
  group_by(loan_status) %>%
  summarise(count=n(), percentage=count/nrow(loan_df)*100) %>%
  knitr::kable()

Defaulted<-c("Charged_Off","Default",
             "Does not meet the credit policy. Status:Charged Off",
             "In Grace Period","Late (16-30 days)","Late (31-120 days)")
Paid<-c("Fully Paid")
loan_df1<-na.omit(loan_df$loan_status)
loan_1<-loan_df %>%
  mutate(identification = ifelse(loan_status %in% Defaulted, 0,
         ifelse(loan_status %in% Paid, 1, "Other")))
unique(loan_1$identification)
loan_1_1<-loan_1 %>%
  select(-loan_status) %>%
  filter(identification %in% c(0,1))

iden<-loan_1_1 %>%
  group_by(identification) %>%
  summarise(count=n(), proportion=count/nrow(loan_1_1))


par(mfrow=c(1,2))

bar<-c("0(32407)","1(986651)")
barplot(iden$count, main = "Barplot of Loan Status", col=c("pink", "lightblue"),
        xlab = "Loan Status", ylab = "Count",
        ylim = c(0,1000000), names.arg = c('0','1'))
legend('topleft', bar, cex=0.6, fill = c("pink", "lightblue"))
```

```
# int & grade
ggplot(loan_1_1, aes(x=grade, y=int_rate, fill=grade))+geom_boxplot()+
  theme_igray()+
  labs(y='Interest Rate', x='Grade',
       title = 'Boxplot of Interest Rates in Different Grades')+
  theme(plot.title=element_text(hjust=0.5))


# map - loan amount
library(choroplethrMaps)
library(choroplethr)
data("state.regions")
state_count<-loan_1_1 %>%
  group_by(addr_state) %>%
  summarise(value = sum(loan_amnt, na.rm=TRUE))
state<-cbind(state.regions, state_count)
state_map<-data.frame(region=state$region, value=state$value)
state_choropleth(state_map, title = "Count by State", legend = "Total Amount")
```

```
library(gmodels)
library(lubridate)
library(plyr)
library(caTools)
library(e1071)
library(ROCR)
library(caret)
library(ROSE)

#------Variables Converting ---------------------
loan.model<-subset(loan_1_1, select = c(1:10))
anyNA(loan.model)   # no missing value
dim(loan.model)     # 9 features + 1 response, 1019058 obs

loan.model<-subset(loan_1_1, select=c(1:4, 5, 5, 5, 5, 6, 7, 10))
loan.model$loan_amnt<-as.numeric(loan.model$loan_amnt)
loan.model$int_rate<-as.numeric(loan.model$int_rate)
loan.model$grade=c("A"=1, "B"=2, "C"=3, "D"=4, "E"=5, "F"=6, "G"=7)[as.numeric(loan.model$grade)]
loan.model$emp_length=c("< 1 year"=0, "1 year"=1, "2 years"=2, "3 years"=3, "4 years"=4,
                        "5 years"=5, "6 years"=6, "7 years"=7, "8 years"=8, "9 years"=9,
                        "10+ years"=10)[as.numeric(loan.model$emp_length)]
loan.model<-rename(loan.model,c(home_ownership="home_ownership_mortgage",
                                home_ownership.1="home_ownership_rent",
                                home_ownership.2="home_ownership_own",
                                home_ownership.3="home_ownership_other"))
loan.model$home_ownership_mortgage=c("MORTGAGE"="1", "RENT"="0", "OWN"="0", "OTHER"="0")[as.character(loan.model$home_ownership_mortgage)]
loan.model$home_ownership_rent=c("MORTGAGE"="0", "RENT"="1", "OWN"="0", "OTHER"="0")[as.character(loan.model$home_ownership_rent)]
loan.model$home_ownership_own=c("MORTGAGE"="0", "RENT"="0", "OWN"="1", "OTHER"="0")[as.character(loan.model$home_ownership_own)]
loan.model$home_ownership_other=c("MORTGAGE"="0", "RENT"="0", "OWN"="0", "OTHER"="1")[as.character(loan.model$home_ownership_other)]
loan.model$home_ownership_mortgage<-as.numeric(loan.model$home_ownership_mortgage)
loan.model$home_ownership_rent<-as.numeric(loan.model$home_ownership_rent)
loan.model$home_ownership_own<-as.numeric(loan.model$home_ownership_own)
loan.model$home_ownership_other<-as.numeric(loan.model$home_ownership_other)
loan.model$annual_inc<-as.numeric(loan.model$annual_inc)
loan.model$term=c("36 months"=36, "60 months"=60)[as.numeric(loan.model$term)]
loan.model$identification<-as.numeric(loan.model$identification)
```

```
# Data Normalization
normalize<-function(x){
  return((x-min(x))/(max(x)-min(x)))
}
loan_norm<-as.data.frame(lapply(loan.model, normalize))

# check multicollinearity
library(GGally)
ggcorr(loan_norm, hjust=0.9, label = T, layout.exp = 1)

library(magrittr)
library(dplyr)
loan_norm_model<-loan_norm %>%
  select(loan_amnt, int_rate, emp_length, home_ownership_mortgage, home_ownership_own,
         home_ownership_other, annual_inc, term, identification)

as.character(loan_norm_model$identification)
table(loan_norm_model$identification)
```

```
#split dataset into training and testing set
set.seed(123)  # make results reproduciable
index<-sample(nrow(loan_norm_model), nrow(loan_norm_model)*0.7)
train.df<-loan_norm_model[index,]
test.df<-loan_norm_model[-index,]
train.df$identification<-factor(train.df$identification)
test.df$identification<-factor(test.df$identification)

library(ROSE)
loan.oversample<-ovun.sample(identification~., data = train.df, method = "over", N=800000,seed = 13)$data
table(loan.oversample$identification)


# Logistic Regression Model
lr.model<-glm(formula = identification~.,family = "binomial", data = loan.oversample)
summary(lr.model)
library(lattice)
library(ggplot2)
library(caret)
lr<-train(identification~., data = loan.oversample, method = "glm")
summary(lr)
test.pred<-predict(lr,test.df)
library(dplyr)
pred_label<-as.factor(if_else(test.pred<0.5,0,1))
confusionMatrix(predict(lr, test.df), test.df$identification)

# ROC
library(ROCR)
lr.prediction<-prediction(predict(lr, newdata = test.df, type = "prob")[,"1"], test.df$identification)
performance(lr.prediction, measure = "auc")@y.values
perf<-performance(prediction.obj = lr.prediction, measure = "tpr","fpr")
plot(perf, colorize=TRUE, main=paste("AUC:", performance(lr.prediction, measure = "auc")@y.values))

# Decision Trees
library(rpart)
tune<-data.frame(0.001)
colnames(tune)<-"cp"
tr_control<-trainControl(method = "cv", number=10, verboseIter = TRUE)
loan.rpart.oversampled<-train(identification~., data = loan.oversample, method="rpart",
                              trControl=tr_control, tuneGrid=tune,
                              control=rpart.control(minsplit = 10, minbucket=3))
library(lattice)
library(ggplot2)
library(caret)
confusionMatrix(predict(loan.rpart.oversampled, test.df), test.df$identification)
loan.rpart.pred<-prediction(predict(loan.rpart.oversampled, newdata=test.df,
                                    type="prob")[,"1"], test.df$identification)
performance(loan.rpart.pred, measure="auc")@y.values


# KNN
train_label<-loan.oversample$identification
test_label<-test.df$identification

library(magrittr)
library(dplyr)
train_knn<-loan.oversample %>%
  select_if(is.numeric) %>%
  scale

test.knn<-test.df %>%
  select_if(is.numeric) %>%
  scale(center = attr(train_knn, "scaled:center"),
        scale = attr(train_knn, "scaled:scale"))

# determine k value
sqrt(nrow(train_knn))
library(class)
model_knn<-knn(train=train_knn, test=test.knn, cl=train_label, k=80)
confusionMatrix(model_knn, test_label)
```

```
# svm
run_svm <- function(data = loan.oversample, cost = 1, kernel = "linear", newdata = test.df, degree = 3, gamma = NA){
  if(is.na(gamma)){
    model <- svm(identification ~ ., data = loan.oversample, cost = cost, kernel = kernel, degree = degree)
  } else {
    model <- svm(identification ~ ., data = loan.oversample, cost = cost, kernel = kernel, degree = degree, gamma = gamma)
  }
  predictions <- as.ordered(predict(model, newdata, type = "response"))
  print(table(newdata[["identification"]], predictions))
  print(paste("Misclassification Rate =",
              mean(newdata[["identification"]] != predictions)))
  print(auc(roc(newdata[["identification"]], predictions)))
}

# Varying cost values
costs <- c(0.1, 0.5, 1, 2, 3)
# Training Set
for (i in costs){
  print(paste("Predicting Training Set with Cost =", i))
  run_svm(cost = i, newdata = loan.oversample)
}
# Test Set
for (i in costs){
  print(paste("Predicting Test Set with Cost =", i))
  run_svm(cost = i)
}
# Varying Polynomial Kernels across Degrees
degrees <- c(2, 3, 4)
# Training Set
for (i in degrees){
  print(paste("Predicting Training Set with Degree =", i))
  run_svm(kernel = "polynomial", newdata = loan.oversample, degree = i)
}
## Test Set
for (i in degrees){
  print(paste("Predicting Test Set with Degree =", i))
  run_svm(kernel = "polynomial", degree = i)
}
# Varying Radial Kernel across Gamma values
gammas <- c(0.1, 1, 3)
## Training Set
for (i in gammas){
  print(paste("Predicting Training Set with Gamma =", i))
  run_svm(kernel = "radial", newdata = loan.oversample, gamma = i)
}
## Test Set
for (i in gammas){
  print(paste("Predicting Test Set with Gamma =", i))
  run_svm(kernel = "radial", gamma = i)
}


# naive bayes
library(e1071)
classifier <- naiveBayes(identification~., loan.oversample)
classifier
prediction <- predict(classifier, select(test.df), type="raw")
summary(prediction)
test.df$identification <- ifelse(prediction[,"1"] > 0.75, "1", "0")
table(prediction, test.df$identification)
library(caret)
confusionMatrix(prediction, test.df$identification)

# linear discriminant analysis
model.LDA <- lda(identification~., data=loan.oversample, na.action="na.omit")
model.LDA
pre.LDA <- predict(model.LDA, na.roughfix(test.df))
summary(pre.LDA$identification)
LDA <- table(pre.LDA, test.df$identification)
caret::confusionMatrix(pre.LDA, test.df$identification)
```