An Analysis on Classifying Personal Loans Pay-Off Status Using Logistic Regression Modelling

Credit Score, Income Bracket and Loan Term are Not Enough to Predict Default Risk on Debt Consolidation Loans for High Income Borrowers with Low Risk Profile

Group Neon: Ava Grier, Ngoné Lo, Yuxin Yan

01/03/2020

Abstract

We explore the possibility of using logistic regression to classify whether a customer may default on loan payments. This technique is applied to a bank loan dataset containing profiles of high income borrowers who obtained debt consolidation loans. Three variables, credit score category, income bracket, and loan term, which have a relationship with loan status, are used in the model. The model has an overall accuracy of 75%. However, the precision and recall of the model for the default class are 33% and 18% respectively, meaning that our model struggles with classifying loan defaults correctly. Indeed, the model appears to be under-fitted and hence may not be the most effective tool to identify loan defaults.

Introduction

In the debt sector, personal loans are the fastest growing type of consumer lending (Friedman, 2018). Greater demand for loans makes it increasingly important for financial institutions to have the foresight to refuse loans to customers who will most likely default on their loans. Unlike a mortgage, personal loans are "unsecured", which means that consumers do not need to put up any collateral. Hence, when a personal loan is defaulted, even though the lender could file a lawsuit against the consumer, there is a high risk that the loan can never be repaid. As a result, it is crucial to evaluate a borrower's repayment chances beforehand and identify those who are most likely to default on personal loans. In this project, we explored a borrower's payback probability using logistic regression based on his/her credit score, income, and loan term. Although credit score category, income bracket and short loan term have a positive correlation with loan repayment, they are not sufficient when it comes to identifying and classifying true instances of defaulting correctly.Our model does a poor job at classifying true instances of defaulting correctly and classifies true instances of repayments as defaulting at a high rate.

Dataset

The original dataset ***credit_train*** was retrieved from Kaggle. It has 100,000 rows and 17 columns. Each row is a past borrower's profile, while 19 variables are available under each profile, e.g. the borrower's credit score and years in current job. The borrower's classification label can be found on the first column *loan status*, where it shows if his/her loan is either *Fully Paid* or *Charged Off*.

With respect to class labels, it was observed that; approximately 100% of the *number of tax liens* were zero; 98% of debt-to-income values were under 35% (***See appendix for how debt-to-income was calculated***); approximately 90% of the *number of bankruptcies* were zero; greater than 80% of the *number of credit problems* were 0; and approximately 80% of *loan purpose* was for debt consolidation. Due to their strong presence, these class labels were used to select the population of customers to be studied.

The dataset required cleaning prior to being analyzed. Variables such as *number of open accounts* are removed, considering that they could be less consequential. Adjustments are made on incorrect values, for example, we decide to normalize 4-digit credit scores down to 3-digit by dividing them by 10. Lastly, rows with missing values are removed. In the end, around 51000 tuples remain in the cleaned dataset.

Exploratory Data Analysis

The variables in the final dataset are *loan status*, *credit score*, *years in current job*, and *annual income*. Prior to modelling, an exploratory data analysis was conducted to help us get a deeper understanding of the dataset. The distribution of loan status, which is our target variable, is shown in **Table 1**. Close to 80% of loans were fully paid; this represents a class imbalance.

Table 1: Proportion Distribution of Loan Status

loan_status	n	percent
Charged Off	10931	21.1%
Fully Paid	40972	78.9%

As for the potential predictor variables, the distribution of credit score and annual income are shown in **Figure 1** and **Figure 2**. The histogram of credit score is skewed to the left as credit scores trend in the higher ranges with most of the population having scores above average. From **Figure 2**, we learn that the population in this dataset has high income levels. Like in real life, the distribution of income in this dataset is right skewed.

Figure 1: Distribution of Credit Score
With Summary Statistics

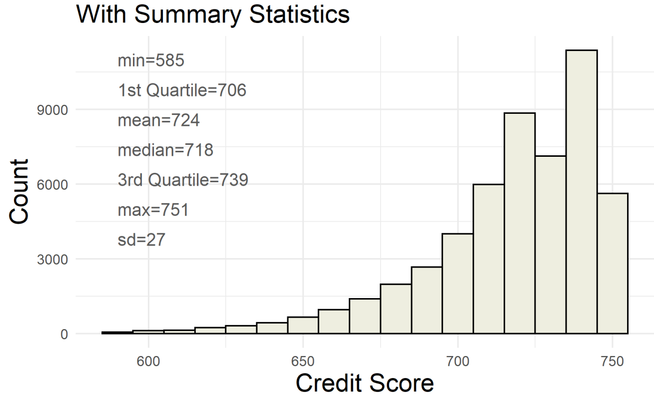
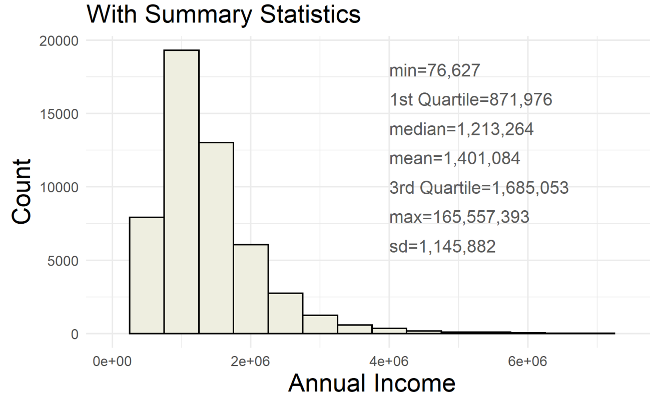


Figure 2: Distribution of Annual Income
With Summary Statistics



The proportion distribution for loan term and years in current job are shown in **Table 2** and **Table 3** respectively. 69% of the loans are short term loans; and 33% of the borrowers have been in their current job for 10 or more years.

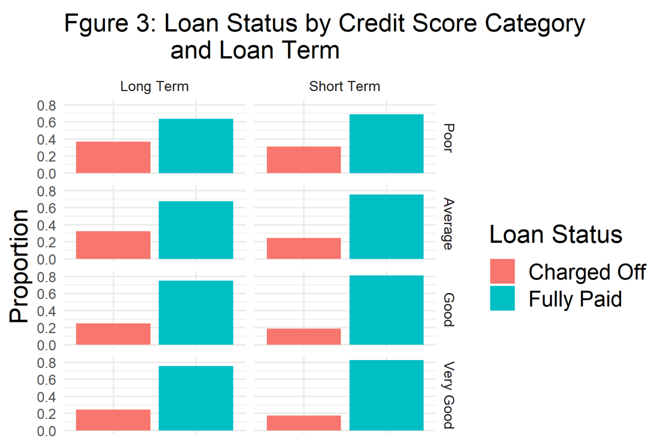
Table 2: Proportion Distribution of Loan Term

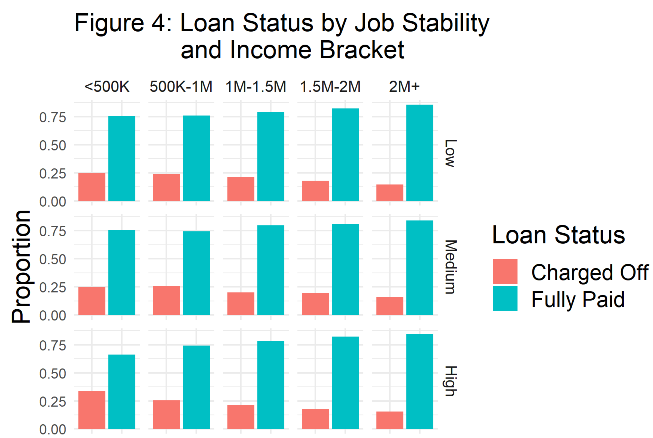
term	n	percent
Long Term	15969	30.8%
Short Term	35934	69.2%

Table 3: Proportion Distribution of Years in Current Job

years_in_current_job	< 1 year	1 year	2 years	3 years	4 years	5 years	6 years	7 years	8 years	9 years	10+ years
n	4408	3452	4925	4416	3168	3634	3059	3065	2462	2180	17134
percent	8.50%	6.70%	9.50%	8.50%	6.10%	7.00%	5.90%	5.90%	4.70%	4.20%	33.00%

To align more with real world practices where people are often assigned to a bracket, we decided to categorize credit score and annual income. Moreover, the variable years in current job was grouped and muted to *job stability*. ***See appendix for categorization and grouping used***. The variation in loan status due to credit score category and loan term is shown in **Figure 3**, while the variation in loan status due to job stability and income bracket is illustrated in **Figure 4**.





Compared to long term loans, short term loans have a higher proportion of loan repayment. Moreover, for both long and short term loans, the better the credit score category, the higher the proportion of loans being fully paid; likewise, there's a positive relationship between proportion of loans being fully paid and income bracket. However, job stability does not seem to have much effect on the proportion of loan status being repaid. We conclude that *credit score category*, *income bracket*, and *loan term* have the most impact on loan repayment. Therefore, we will use these variables to build our final classification model.

Logistic Regression Modelling

Logistic regression is a supervised learning classification algorithm that uses one or more independent variables to determine an outcome with only two possibilities (Statistics Solutions), as is the case with this dataset. Creating predictive models on this data can prove to be valuable because its internal rate of default, at 21%, is substantially higher than the industry rate of 3.3% (Kirkham, 2020). To fully learn the predictive effect of the selected variables on loan repayment behaviour, we ran a logistic regression model with *credit score category*, *income bracket*, and *loan term* as predictor variables and *loan status* as the target variable. We used a 75%-25% split for training and test sets. Credit score category and income bracket were numerically encoded and loan term was encoded into a boolean. Binary encoding was used for the target variable loan status with *charged off* as 0 and *fully paid* as 1. Equation 1 shows the equation of the model in terms of the estimated probability of a loan being repaid and equation 2 in terms of the estimated log of the odds.

$$\hat{P}(loan\ repaid) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1(credit\ score\ category)+\hat{\beta}_2(income\ bracket)+\hat{\beta}_3(loan\ term)}}{e^{\hat{\beta}_0+\hat{\beta}_1(credit\ score\ category)+\hat{\beta}_2(income\ bracket)+\hat{\beta}_3(loan\ term)}+1} \tag{1}$$

$$\log\left(\frac{\hat{P}(loan\ repaid)}{1-\hat{P}(loan\ repaid)}\right) = \hat{\beta}_0+\hat{\beta}_1(credit\ score\ category)+\hat{\beta}_2(income\ bracket)+\hat{\beta}_3(loan\ term) \tag{2}$$

Table 4 displays the coefficients output of the model.

Table 4: Model Coefficient Ouptput

term	estimate	std.error	statistic	p.value
(Intercept)	0.7052882	0.0541235	13.031093	0
credit_score_category_encoded	0.1328005	0.0182712	7.268275	0
income_bracket_encoded	0.2334729	0.0119699	19.504934	0
long_term_loanTRUE	-0.5197044	0.0298075	-17.435347	0

For a customer with a short term loan, a poor credit score, and an annual income of less than 500K, the log of the odds of paying back their loan is 0.71. With income and and loan term kept constant, the log of the odds of paying a loan goes up by 0.13 for every unit increase in credit score category (e.g. from average to good). Similarly, with credit score category and loan term kept constant, the log of the odds of paying back a loan goes up by 0.23 for every unit increase in income bracket (e.g. from 1M-1.5M to 1.5M-2M). Finally, compared to a short term loan, with credit score category and income bracket kept constant, the log of the odds of paying back a loan is 0.52 times less for a long term loan. All these coefficients are statistically significant.

The model was initially ran with a fit of 0.5, indicating that if the model predicts the possibility of *fully paid* to be over 0.5, the result is deemed to be *fully paid*. However, this created under-fitting as every customer was predicted to have paid off their loans. When they were fitted to 0.7 instead, a more reasonable classification is obtained.

Results and Discussion

One way to assess the predictive performance of a model is to review its confusion matrix output. In general, good models will have a high proportion of true negatives and true positives and low proportion of false negative and false positives. **Table 5** and **Table 6** shows the confusion matrix and the accuracy metrics of our model on its test set respectively. **Table 7** shows some performance metrics related to the *charged off* class.

We believe it is more risky for financial institutions to lend to customers who won't repay (false negatives) than to refuse loans to customers who will repay (false positives). Hence, we decided to focus on the charged-off class. Our model has an accuracy of 75%. However, the recall of the charged class (negative class or 0) is only at 18% meaning that our model is effective at predicting non-repayments instances only 18% of the time. Furthermore, the precision for the charged-off

Table 5: Confusion Matrix of the Model

X1	actual charged off	actual fully paid
predicted charged off	495	990
predicted fully paid	2219	9271

Table 6: Accuracy Metrics of the Model

accuracy	Lower accuracy	upper accuracy	balanced accuracy
0.752678227	0.7451597	0.7600833	0.5429529

Table 7: Performance Metrics for the Charged Off Class

precision	recall	f1-score
0.333333333	0.1823876	0.2357704

class is only 33% meaning that only 33% of the instances predicted as non-repayment are true non-repayment instances. To better understand how far off the model is, we plot the estimated probabilities of loan being repaid against the true instances of loan status for both the training and test sets. See **Figure 5**. It looks like our model is under-fitted and thus suffers from low variance and high bias. The predictive power of the model can be viewed in **Figure 6** which shows the calculated probability of loan repayment for short term versus long-term loans based on credit score category and income bracket. In accordance with **Table 4**, the probability of repayment increases with credit score category and income bracket going up. Similarly, with credit score category and income bracket kept constant, short loan term have a higher probability of being paid back than long term loans.

Figure 5: Estimated Probability of Loan being Repaid vs. True Status of Loan

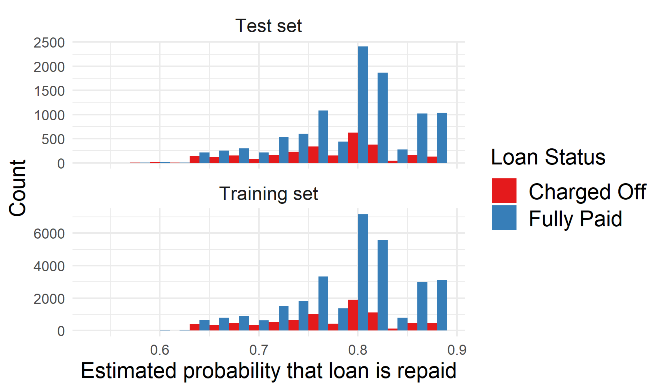


Figure 6: Probability of Loan Being Repaid based on Credit Score Group, Term of Loan, and Income Bracket

Cross Validation

We used k-fold cross validation to assess the performance of our logistic regression model on limited data. Ideally, unseen data should be used for cross validation. However, because we have already used all our data for training and testing, we conducted the cross validation analysis on the training set and set k=10 folds as per the general recommendation. **Table 8** shows the accuracy metrics of our cross validation analysis.

Table 8: Accuracy Metrics of the Cross Validation Analysis

accuracy	accuracySD
0.788918	0.000107646

The accuracy of our cross validation analysis is about 79%, which is a liitle bit better than the accuracy of our model.

Conclusion

We apply logistic regression to classify whether a loan would be fully paid or charged off. Based on three selected variables, loan term, borrower's annual income and credit score, we achieve a model with 75% accuracy. However, by looking at the model's confusion matrix, precision of 33%, and recall of 18%, we determine that the model would not serve as a great tool when granting loans, as it may not effectively detect loan defaulters.

Weaknesses

We unfortunately do not have a readme file explaining all the columns in the dataset. As a result, we choose to either drop columns or make assumptions when interpreting the data. One example is that we drop the "Current.Loan.Amount" column, because we are not sure whether it refers to the customer's previously existing loan or the new loan they are granted; Another example of ambiguity, is that we do not know the currency for income in the dataset. If we assume income is in Canadian dollars, that assumption leads to the consideration that the minimum income in this dataset, \$76627, is not at all representative of income for Canadians. In fact, the median household income in Canada in 2015 was \$70336 (Alini, 2017). In this case, our model may not be suitable to classify loan status for customers who earn the median household income or lower .

Ethical Consideration

The classification algorithm could potentially be discriminative. In our model, with the annual income being a predicting variable, it might grant loans to a person who has higher income and reject a person who has lower income, even though they're both applying for the same amount of loan and are both capable of paying back. Whether to grant a loan should be based on risk assessment, but whether income and credit scores can reflect the true risks remain unclear. While the goal of developing the model is to minimize the risk for the financial institutions, it simply filters out the people who are not "good" enough to be a customer, discriminating against them because of their income and credit scores.

At the same time, the algorithm could be manipulated and may not provide enough transparency to customers. In our model, 0.7 fitting is chosen; however, if a different fitting is chosen, or a different dataset is used, the decision for many loan applications could be the opposite. On one hand, it's unfair for the customers; on the other hand, the customers would not get enough information on how the algorithm works exactly, hence they would not know how they could improve their chance to get the loan.

References

RStudio Team (2019). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Firke, Sam (2020). janitor: Simple Tools for Examining and Cleaning Dirty Data. R package version 1.2.1. https://CRAN.R-project.org/package=janitor.

Elin Waring, Michael Quinn, Amelia McNamara, Eduardo Arino de la Rubia, Hao Zhu and Shannon Ellis (2020). skimr: Compact and Flexible Summaries of Data. R package version 2.1. https://CRAN.R-project.org/package=skimr

David Robinson and Alex Hayes (2020). broom: Convert Statistical Analysis Objects into Tidy Tibbles. R package version 0.5.4. https://CRAN.R-project.org/package=broom

Max Kuhn and Hadley Wickham (2020). tidymodels: Easily Install and Load the 'Tidymodels' Packages. R package version 0.1.0. https://CRAN.R-project.org/package=tidymodels

Max Kuhn (2020). caret: Classification and Regression Training. R package version 6.0-85. https://CRAN.R-project.org/package=caret

Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.28.

Hao Zhu (2019). kableExtra: Construct Complex Table with 'kable' and Pipe Syntax. R package version 1.1.0. https://CRAN.R-project.org/package=kableExtra

Begiev, Z. (2017). Bank Loan Status Dataset. Retrieved February 14, 2020, from https://www.kaggle.com/zaurbegiev/my-dataset

Statistics Solutions. What is Logistic Regression? (n.d.). Retrieved February 26, 2020, from https://www.statisticssolutions.com/what-is-logistic-regression/

Friedman, Z. (2018). Personal Loans Now Fastest Growing Consumer Debt. Retrieved from https://www.forbes.com/sites/zackfriedman/2018/07/11/personal-loans-consumer-debt/#122b3acf55ad

Alini, E. (2017). Are you earning a middle-class income? Here's what it takes in Canada, based on where you live. Retrieved from https://globalnews.ca/news/3828447/canada-middle-class-income-inequality/

Kirkham, E. (2020). Personal Loan Statistics for 2020. (2020, January 24). Retrieved from https://www.lendingtree.com/personal/personal-loans-statistics/?ccontent=a1LgFw09t88&s1=a1LgFw09t88&s2=a1LgFw09t88-QNGYZVg__xXXCNScCh1R8eQ&ranMID=41202&ranEAID=a1LgFw09t88&ranSiteID=a1LgFw09t88-QNGYZVg__xXXCNScCh1R8eQ&PUBSID=2126220&PUBNAME=adgoal.net

Appendix

Debt-to-Income debt-to-income = (monthly debt * 12)/annual income

Categorization and Grouping

Credit Score/Credit Score Category
<620 ==> Poor
620-689 ==> Average
680-719 ==> Good
>=720 ==> Very Good
Annual Income/Income Bracket
<500,000 ==> <500K
500,000-1,000,000 ==> 500K-1M
1,000,000-1,500,000 ==> 1M-1.5M
1,500,000-2,000,000 ==> 1.5M-2M
>2,000,000 ==> 2M+
Years in Current Job/Job stability
0-3 years ==> Low
4-7 years ==> Medium
8 years+ ==> High

Approval to use External Dataset

We obtained approval to use this dataset on Friday February 21 2020 via email. Here is a screenshot of the email.

Re: Problem Set3 Data Source

RA Rohan Alexander <rohan.alexander@utoronto.ca>
To Ava Grier
Cc Ngone Lo

Reply | Reply All | Forward | ...

Fri 2020-02-21 7:55 PM

Dear Ava,

Thank you very much for your email.

The dataset looks great.

No worries to read from your local drive. I have a Kaggle account, so if I want the data then I'll just sign in and grab it.

Looking forward to reading your submission.

Rohan

On Feb 21, 2020, at 7:53 PM, Ava Grier <a.grier@mail.utoronto.ca> wrote:

Hi Rohan, for problem set 3 one of the requirement is that we use a dataset from Open Data Toronto, however you had given me permission to use a dataset from kaggle.com instead. The dataset can be obtained from the link: https://www.kaggle.com/zaurbegiev/my-dataset however one is required to be signed in to Kaggle to be able to download the files.

I am confirming that you have given additional permission to have the dataset read from a local drive in the R code rather than being pulled from Kaggle's site due to the sign in restriction. The link to the dataset will be commented in the code, as you suggested

Thank you,
Ava

Code

Cleaning Dataset

```r
## @knitr cleaning_data

#### Set up workspace ###
#Importing libraries
library(janitor) # Helps with initial data cleaning and pretty tables
library(skimr) # Helps with initial data visualisation
library(tidyverse)

#Loading dataset
loan_data <- read_csv("inputs/credit_train.csv")

#Cleaning the variables' names using Janitor
loan_data <- clean_names(loan_data)

#Overview
skim(loan_data)


#Drop NA rows for annual_income
loan_data<- loan_data %>% drop_na(annual_income)

#Calculate annual debt_to_income
loan_data$debt_to_income=
  (loan_data$monthly_debt*12)/loan_data$annual_income

loan_data<- loan_data %>%
  #Filter to only have population of interest
  #Low risk population with debt consolidation as purpose for loan
  filter(debt_to_income <= 0.35 &
           purpose == "Debt Consolidation" &
           number_of_credit_problems == 0 &
           bankruptcies == 0 &
           tax_liens == 0)

#Select columns/variables of interest
loan_data <- loan_data %>%
  select(loan_status, credit_score, years_in_current_job,
         term, annual_income)

#Overview of data with selected variables
skim(loan_data)
```

```
#Filter out n/a rows in years_in_current_job
loan_data<- loan_data %>%
  filter(years_in_current_job!="n/a")


#There are credit score values bigger than the maximum of 850.
#Let's get an overview of the unique values to get an idea of the situation
unique(loan_data$credit_score[loan_data$credit_score>850])

#Looks like there was an additional zero added for the credit score values
#greater than 850. We divide the credit score values greater than
#850 by 10 o get them back to normal range
loan_data <- loan_data %>%
  mutate(credit_score = case_when(
    credit_score>850 ~ credit_score/10,
    credit_score<=850 ~ credit_score))

#Overview of the variable credit score
skim(loan_data$credit_score)


#The final cleaned dataset has 51903 observations for 5 variables


#### Save cleaned dataset ####
write_csv(loan_data, "outputs/datasets/loan_data_cleaned.csv")
```

Exploratory Dataset Analysis

```
## @knitr eda

#### Set up workspace ###
#Importing libraries
library(tidyverse)
library(janitor) # Helps with initial data cleaning and pretty tables

#Loading dataset
loan_data <- read_csv("outputs/datasets/loan_data_cleaned.csv")

#First we take a look at the prediction variable: loan status
#Overview of the variable term
status_prop <- loan_data %>%
  tabyl(loan_status) %>%
  adorn_pct_formatting()
```

```r
#### Save term_proportion table####
write_csv(status_prop, "outputs/tables/loan_status.csv")

#The data is highly biased toward Fully Paid Loan Status



#Second, we take a look at credit score

#Summary statistics of the variable credit score
summary(loan_data$credit_score)

#Histogram of credit score
ggplot(data = loan_data, mapping = aes(x=credit_score)) +
  geom_histogram(binwidth=10, color="black", fill="ivory2" )+ #plot histogram
  #annotate summary statistics to plot
  annotate("text", label = "min=585", x = 590, y = 11000, color = "gray35",
           hjust = 0, size=4)+
  annotate("text", label = "1st Quartile=706", x = 590, y = 9800, color = "gray35",
           hjust = 0, size=4)+
  annotate("text", label = "mean=724", x = 590, y = 8600, color = "gray35",
           hjust = 0, size=4)+
  annotate("text", label = "median=718", x = 590, y = 7400, color = "gray35",
           hjust = 0, size=4)+
  annotate("text", label = "3rd Quartile=739", x = 590, y = 6200, color = "gray35",
           hjust = 0, size=4)+
  annotate("text", label = "max=751", x = 590, y = 5000, color = "gray35",
           hjust = 0, size=4)+
  annotate("text", label = "sd=27", x = 590, y = 3800, color = "gray35",
           hjust = 0, size=4)+
  theme_minimal() + # Make the theme neater
  #Define title, subtile, and axis size
  theme(plot.title =element_text(size = 16),
        plot.subtitle =element_text(size = 16),
        axis.title = element_text(size=16))+
  #Define title, subtitle, and axis labels
  labs(title= "Figure 1: Distribution of Credit Score",
       subtitle="With Summary Statistics",
       x="Credit Score",
       y="Count")

#### Save the graph ####
ggsave("outputs/figures/credit_score_distribution.png",
       width = 15, height = 10, units = "cm")
```

```r
#Third, we take a look at annual income

#Summary statistics of the variable annual_income
summary(loan_data$annual_income)

#Looks like we are dealing with millionaires here

#Histogram of annual income
ggplot(loan_data) +
  #plot histogram
  geom_histogram(aes(x=annual_income), binwidth=500000,
                 color="black", fill="ivory2")+
  #annotate summary statistics to plot
  annotate("text", label = "min=76,627", x = 4000000, y = 18000,
           color = "gray35", hjust = 0, size=4)+
  annotate("text", label = "1st Quartile=871,976", x = 4000000, y = 16000,
           color = "gray35", hjust = 0, size=4)+
  annotate("text", label = "median=1,213,264", x = 4000000, y = 14000,
           color = "gray35", hjust = 0, size=4)+
  annotate("text", label = "mean=1,401,084", x = 4000000, y = 12000,
           color = "gray35", hjust = 0, size=4)+
  annotate("text", label = "3rd Quartile=1,685,053", x = 4000000, y = 10000,
           color = "gray35", hjust = 0, size=4)+
  annotate("text", label = "max=165,557,393", x = 4000000, y = 8000,
           color = "gray35", hjust = 0, size=4)+
  annotate("text", label = "sd=1,145,882", x = 4000000, y = 6000,
           color = "gray35", hjust = 0, size=4)+
  theme_minimal() + # Make the theme neater
  #Define title, subtitle, and axis size
  theme(plot.title =element_text(size = 16),
        plot.subtitle =element_text(size = 16),
        axis.title = element_text(size=16))+
  #Define title, subtitle, and axis labels
  labs(title= "Figure 2: Distribution of Annual Income",
       subtitle= "With Summary Statistics",
       x="Annual Income",
       y="Count") +
  xlim(c(0, 7500000)) #set x axis limit

#### Save the graph ####
ggsave("outputs/figures/annual_income_distribution.png",
       width = 15, height = 10, units = "cm")


#The other variables are categorical and are:
# 1 Term of Loan (Long term, Short Term) and
```

```r
# 2. Years in Current Job (> 1 year, 1 Year, 2 years,..., 10+ years )

#Overview of the variable term
term_prop <- loan_data %>%
  tabyl(term) %>%
  adorn_pct_formatting()

#### Save term_proportion table####
write_csv(term_prop, "outputs/tables/term_proportion.csv")

#Re-leveling years in current job
loan_data$years_in_current_job<-factor(loan_data$years_in_current_job,
                                 levels=c("< 1 year", "1 year", "2 years", "3 years",
                                          "4 years", "5 years", "6 years", "7 years",
                                          "8 years", "9 years", "10+ years",))
#Overview of the variable years in current in job
current_job_years_prop <- loan_data %>%
  tabyl(years_in_current_job) %>%
  adorn_pct_formatting()

#### Save term_proportion table####
write_csv(current_job_years_prop, "outputs/tables/current_job_years_proportion.csv")

#Because of skewdness reasons and to align more with real world practices
#where people are often assigned to a income bracket, we decided to categorize
#credit score and annual income.

#Categorizing credit score
loan_data <- loan_data %>%
  mutate(credit_score_category = case_when(
    credit_score < 620 ~ "Poor",
    credit_score >= 620 & credit_score < 690  ~ "Average",
    credit_score >= 690 & credit_score < 720  ~ "Good",
    credit_score >= 720  ~ "Very Good"), credit_score = NULL)

#Overview of the grouped credit_score
table(loan_data$credit_score_category)


#Categorizing annual_income
loan_data <- loan_data %>%
  mutate(income_bracket = case_when(
    annual_income <500000 ~ "<500K",
    annual_income >= 500000 & annual_income < 1000000  ~ "500K-1M",
    annual_income >= 1000000 & annual_income < 1500000  ~ "1M-1.5M",
    annual_income >= 1500000 & annual_income < 2000000  ~ "1.5M-2M",
```

```r
    annual_income >= 2000000 ~ "2M+"), annual_income = NULL)

#Overview of the grouped income
table(loan_data$income_bracket)


#For simplicity reasons, we re-grouped years in current job
#in 3 groups instead of 11 groups

#Categorizing job stability based on years_in_current_job
#Store possible categories in vectors
low <- c("< 1 year", "1 year", "2 years", "3 years")
medium <- c("4 years", "5 years", "6 years", "7 years")
high <- c("8 years", "9 years", "10+ years")

#Create job_stability column by grouping years_in_current_job:
loan_data <- loan_data %>%
  mutate(job_stability = case_when(
    years_in_current_job %in% low ~ "Low", #Assign Low job stability
    years_in_current_job %in% medium ~ "Medium",  #Assign Medium job stability
    years_in_current_job %in% high ~ "High"  #Assign High job stability
  ), years_in_current_job=NULL)

#Overview of the grouped job_stability
table(loan_data$job_stability)


#We suspect credit score and term of loan to be
#the best predictors of loan status
#Re-leveling credit score
loan_data$credit_score_category<-factor(
  loan_data$credit_score_category,levels=
    c("Poor", "Average", "Good", "Very Good"))

loan_summary1 <- loan_data %>%
  group_by(credit_score_category, term, loan_status) %>%
  summarise(n = n()) %>% # Count the number in each group and response
  group_by(credit_score_category, term) %>%
  mutate(prop = n/sum(n)) # Calculate proportions within each group

ggplot(loan_summary1) +
  #Specify a barplot of loan status
  geom_col(aes(x = loan_status, y = prop, fill=loan_status)) +
  #Facet by credit category score and term
  facet_grid(credit_score_category~term) +
  theme_minimal()+  #Make the theme neater
```

```r
  #Define size of title, axis and legend
  theme(plot.title =element_text(size = 16),
        axis.title.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 14),
        strip.text = element_text(size = 9))+
  #Define title, y_axis, and legend labels
  labs(title= "Fgure 3: Loan Status by Credit Score Category
               and Loan Term",
       y="Proportion", fill="Loan Status")+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank() #Delete x_labels
        )


#### Save the graph ####
ggsave("outputs/figures/loan_status_by_credit_score_term.png",
       width = 15, height = 10, units = "cm")

#Loan status vary with both credit score category and term loan


#Next we look at job stability and income bracket
#Re-leveling job stability
loan_data$job_stability<-factor(loan_data$job_stability,
                                levels=c("Low", "Medium", "High"))
#Re-leveling income bracket
loan_data$income_bracket<-factor(loan_data$income_bracket,
                                 levels=c("<500K", "500K-1M", "1M-1.5M",
                                          "1.5M-2M", "2M+"))

#Group by job stability  and income bracket. Calculate proportions
loan_summary2 <- loan_data %>%
  group_by(job_stability, income_bracket, loan_status) %>%
  summarise(n = n()) %>% # Count the number in each group and response
  group_by(job_stability, income_bracket) %>%
  mutate(prop = n/sum(n)) # Calculate proportions within each group

ggplot(loan_summary2) +
  #Specify a barplot of loan status
  geom_col(aes(x = loan_status, y = prop, fill=loan_status)) +
  #Facet by job stability and income bracket
  facet_grid(job_stability~income_bracket) +
  theme_minimal()+  #Make the theme neater
  #Define size of title, axix and legend
```

```r
  theme(plot.title =element_text(size = 16),
        axis.title.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 14),
        strip.text = element_text(size = 10))+
  #Define title, y_axis, and legend labels
  labs(title= "Figure 4: Loan Status by Job Stability
             and Income Bracket",
       y="Proportion", fill="Loan Status")+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank() #Delete x_labels
  )

#### Save the graph ####
ggsave("outputs/figures/loan_status_by_job_stability_income.png",
       width = 15, height = 10, units = "cm")

#While income bracket seems to cause significant difference, job stability
#does not. Hence credit score category, term loan, and income bracket
#will be used as our predictors in our logistic model


#Select columns/variables of interest
loan_data <- loan_data %>%
  select(loan_status, credit_score_category, term, income_bracket)

#### Save selected data for model####
write_csv(loan_data, "outputs/datasets/loan_data_selected.csv")
```

Modelling

```r
## @knitr model


#### Set up workspace ###
#Importing libraries
library(broom) # Helps with model outputs
library(tidymodels) # Help with modelling
library(caret) #Helps with classification regression models and cross validation
library(tidyverse)

#Loading dataset
loan_data <- read_csv("outputs/datasets/loan_data_selected.csv")
```

```r
#Re-leveling income bracket
loan_data$income_bracket<-factor(loan_data$income_bracket,
                                levels=c("<500K", "500K-1M", "1M-1.5M",
                                         "1.5M-2M", "2M+"))

loan_data <- loan_data %>%
  mutate(credit_score_category_encoded = case_when(
    credit_score_category == "Poor" ~ 0,
    credit_score_category == "Average" ~ 1,
    credit_score_category == "Good" ~ 2,
    credit_score_category == "Very Good" ~ 3
  ))

#convert loan term to boolean
loan_data <- loan_data %>%
  mutate(long_term_loan = case_when(
    term == "Long Term" ~ TRUE,
    term == "Short Term" ~ FALSE
  ))

#encode income bracket to numeric
loan_data <- loan_data %>%
  mutate(income_bracket_encoded = case_when(
    income_bracket == "<500K" ~ 0,
    income_bracket == "500K-1M" ~ 1,
    income_bracket == "1M-1.5M" ~ 2,
    income_bracket == "1.5M-2M" ~ 3,
    income_bracket == "2M+" ~ 4
  ))

#convert loan status to binary
loan_data <- loan_data %>%
  mutate(loan_paid = case_when(
    loan_status == "Fully Paid" ~ 1,
    loan_status == "Charged Off" ~ 0
  ))


#Set seed for reproducibility
set.seed(1203)

# Split the data into test/training sets
loan_data_split <-
  loan_data %>%
  initial_split(prop = 3/4)
```

```r
loan_train <- training(loan_data_split)
loan_test <- testing(loan_data_split)

rm(loan_data_split)


#### Model ####
#predict loan_paid using credit score, income_bracket, long_term
model <- glm(loan_paid ~
               credit_score_category_encoded + income_bracket_encoded +
               long_term_loan,
             data = loan_train, family="binomial")


# model output: coeffcicients
coeff_output<-tidy(model)
coeff_output
#### Save coeff_output table####
write_csv(coeff_output, "outputs/tables/coeff_output.csv")


#Look at what the model predicts, compared with the actual
#Cut-off probability = 0.7
loan_model1_fit_train <-
  augment(model,
          data = loan_train,
          type.predict = "response") %>% #
  select(-.hat, -.sigma, -.cooksd, -.std.resid) %>%
  mutate(predict_loan_paid = if_else(.fitted > 0.7, 1, 0))


#How many loans were predicted as charged off (not paid) "0" and
#how many were predicted as Fully paid "1" in the training set
table(loan_model1_fit_train$predict_loan_paid)


#Look at the distribution of how far off the model is
loan_model1_fit_train %>%
  ggplot(aes(x = .fitted, fill = loan_status)) +
  #Specify a histogram of loan status
  geom_histogram(binwidth = 0.02, position = "dodge") +
  theme_minimal()+  #Make the theme neater
  #Define size of title, axix and legend
  theme(plot.title =element_text(size = 14),
        axis.title = element_text(size = 14),
        legend.title = element_text(size = 14),
```

```
        legend.text = element_text(size = 14))+
  #Define title, y_axis, and legend labels
  labs(title = "Estimated Probability of Loan being Repaid vs.
               True Status of Loans",
       x = "Estimated probability that loan is repaid",
       y = "Count",
       fill = "Loan Status") +
  #Choose color palette
  scale_fill_brewer(palette = "Set1")



#### Save the graph ####
ggsave("outputs/figures/model_distribution.png",
       width = 15, height = 10, units = "cm")



#How the model probabilities change based on credit score, income bracket,
#and term of loan
ggplot(loan_model1_fit_train,
       aes(x = credit_score_category_encoded,
           y = .fitted,
           color = income_bracket)) +
  geom_line() +
  geom_point() +
  facet_wrap(term~.)+ #Facet by loan term
  theme_minimal()+  #Make the theme neater
  #Define size of title, axix and legend
  theme(plot.title =element_text(size = 14),
        axis.title = element_text(size = 14),
        legend.title = element_text(size = 14),
        legend.text = element_text(size = 14),
        strip.text = element_text(size = 12))+
  #Define title, y_axis, and legend labels
  labs(title = "Figure 6: Probability of Loan Being Repaid based on Credit
               Score Group, Term of Loan, and Income Bracket",
       x = "Credit Score Group",
       y = "Predicted probability that loan is repaid",
       color = "Income Bracket") +
  #choose color palette
  scale_color_brewer(palette = "Set1")


#### Save the graph ####
ggsave("outputs/figures/predictors_model_change.png",
```

```r
      width = 15, height = 10, units = "cm")



#confusion matrix of training set to compare prediction to actual values
confusionMatrix(data = as.factor(loan_model1_fit_train$predict_loan_paid),
                reference = as.factor(loan_model1_fit_train$loan_paid),
                mode="prec_recall")




# adding the test to analysis: fit test set
loan_model2_fit_test <-
  augment(model,
          newdata = loan_test,
          type.predict = "response") %>%
  mutate(predict_loan_paid = if_else(.fitted > 0.7, 1, 0))


#confusion matrix of test set to compare predictions to actual values
confusion_matrix_test <- confusionMatrix(data = as.factor(
  loan_model2_fit_test$predict_loan_paid),
  reference = as.factor(loan_model2_fit_test$loan_paid), mode="prec_recall")

confusion_matrix_test

#charged off class performance metrics
performance_test<-tidy(confusion_matrix_test, mode="prec_recall")
#### Save performance_test####
write.csv(performance_test, "outputs/tables/performance_class0.csv")

#accuracy metrics
accuracy_test <- as.data.frame(as.matrix(confusion_matrix_test, what="overall"))
#### Save coeff_accuracy_metrics####
write.csv(accuracy_test, "outputs/tables/accuracy_metrics.csv")

#confusion matrix
confusion_matrix<-as.table(confusion_matrix_test)
#### Save confusion_matrix####
write.csv(confusion_matrix, "outputs/tables/confusion_matrix.csv", row.names = TRUE)




#compare the test with the training sets in terms of forecasts.
#select required columns for the training and test graphs
training <- loan_model1_fit_train %>%
```

```r
  select(loan_status, .fitted) %>%
  mutate(type = "Training set")

test <- loan_model2_fit_test %>%
  select(loan_status, .fitted) %>%
  mutate(type = "Test set")

#combine training and test in one set and remove them afaterwards
both <- rbind(training, test)
rm(training, test)

#Look at the distribution of how far off the model is for
#both training and test sets
both %>%
  ggplot(aes(x = .fitted, fill = loan_status)) +
  #Specify a histogram of loan status
  geom_histogram(binwidth = 0.02, position = "dodge") +
  theme_minimal()+  #Make the theme neater
  #Define size of title, axix and legend
  theme(plot.title =element_text(size = 14),
        axis.title = element_text(size = 14),
        legend.title = element_text(size = 14),
        legend.text = element_text(size = 14),
        strip.text = element_text(size = 12))+
  #Define title, y_axis, and legend labels
  labs(title = "Figure 5: Estimated Probability of Loan being Repaid vs.
              True Status of Loan",
       x = "Estimated probability that loan is repaid",
       y = "Count",
       fill = "Loan Status") +
  #Choose color palette
  scale_fill_brewer(palette = "Set1") +
  #facet by training and test type and free/independent y_axis
  facet_wrap(type~.,
             nrow = 2,
             scales = "free_y")

#### Save the graph ####
ggsave("outputs/figures/model_distribution_training_test.png",
       width = 15, height = 10, units = "cm")



#Cross Validation
# Define train control for k fold cross validation
train_control <- trainControl(method="cv", number=10)
```

```r
# Fit logistic regression model on training set
model_cross <- train(loan_status ~
                       credit_score_category_encoded + income_bracket_encoded +
                       long_term_loan,
                     data = loan_train,
                     trControl=train_control,
                     method= "glm",
                     family="binomial")

cross_val<-model_cross$results

#cross validation accuracy metrics
cross_val_accuracy <- as.data.frame(as.matrix(cross_val, what="overall"))
#### Save coeff_accuracy_metrics####
write.csv(cross_val_accuracy, "outputs/tables/cross_val_metrics.csv")
```