Loan Default Prediction by Logistic Regression

Submitted by

Souraj Mazumdar	211393
Soumya Paul	211391
Soumita Bandyopadhyay	211390
Rahul Ghosh Dastidar	211353

Under the Guidance of

Dr. Sharmishtha Mitra

Department Of Mathematics And Statistics,

IIT Kanpur

Abstract

. With the improvement of the banking sector in recent times and the increasing trend of loans, a large population asks for bank loans. But one of the major problem banking sectors face in this ever-changing economy is the increasing rate of loan defaults, and the banking authorities find it more difficult to properly assess loan requests and address the default risks of borrowers. The two most critical questions in the banking industry are (i) How risky is the borrower? and (ii) Given the borrower's risk, should we lend him/her? For the the given problem, this project proposes one of the well-known model to predict whether an customer would be given a loan by assessing certain attributes and therefore help the banking authorities by facilitating their process of selecting the right people from a given list of candidates who have applied for a loan. Here we have two datasets where we have fitted Logistic Regression Model based on train data and used that model for knowing whether a new customer is going to default his/her loan.

Acknowledgment

It is our pleasure to present a project on “Loan Default Prediction by Logistic Regression”. Every accomplishment has constant encouragement and advice from valuable and noble minds to guide us in putting our efforts in the right direction to bring out the project. We want to express our sincere gratitude to our instructor **Dr. Sharmishtha Mitra** for her constant help and support throughout the completion of the project. Without her valuable guidance and motivation, it was nearly impossible to work on this project as a team and understand the practical aspect of the course “MTH 416A: Regression Analysis”. Also we are thankful to all faculty members and seniors without whose support at various stages, this project would not have materialized. Finally my earnest thanks go to my friends who were always beside me when I needed them without any excuses.

Souraj Mazumdar

Soumya Paul

Soumita Bandyopadhyay

Rahul Ghosh Dastidar

Contents

1 Introduction

In finance, default is a failure to pull off the legal obligations (or conditions) of a loan, for instance, when a buyer fails to pay a mortgage payment, or when a company or government fails to pay a bond that has reached maturity, it's a default. A national or sovereign default is the failure or refusal of a government to reimburse its debt.

The biggest private default in history is Lehman Brothers, with over $600 billion when it filed for bankruptcy in 2008. The largest sovereign default is Greece, with $138 billion in March 2012.

In other words, default is the failure to repay a debt, including interest or principal, on a loan or security. A default can occur when a borrower is unable to pay off timely, misses payments, or avoids or stops making payments. Individuals, corporations, and even countries can default if they cannot keep up their debt obligations. Default risks are often calculated well in advance by creditors.

Defaults can have consequences, such as lowering credit scores, reducing the possibility of obtaining credit in the future, raising interest rates on existing debt, and also any new obligations.

Loan default occurs when a loanee fails to pay back a debt in keeping with the initial arrangement. Within the case of most consumer loans, this suggests that consecutive payments are missed over the course of weeks or months. Fortunately, lenders and loan servicers usually allow a grace period before penalizing the borrower after missing one payment. The amount between missing a loan payment and having the loan default is thought of as delinquency. The delinquency period gives the debtor, time to avoid default by contacting their loan servicer or making up missed payments. When a loan defaults, it's sent to a debt collection agency that contacts the borrower and receives the unpaid funds. When a borrower defaults on a loan, the repercussions may additionally include:

- Negative remarks on a borrower's credit report and lowering of their credit score, which could be a numerical value or measure of his creditworthiness
- Reduced chances of obtaining credit in the future
- Higher interest rates on existing debt also as any new debt
- Garnishment of wages and other penalties. Garnishment refers to a legal procedure that instructs a 3rd party to deduct payments directly from a borrower's wages or checking account.

The consequences of defaulting on a loan of any type are severe and may be avoided at the least cost. If one misses a payment or the loan is in delinquency for some months, the most effective thing to try to do is to contact the corporate who manages that loan. Often, loan servicers will work with debtors to make a payment plan that works for both parties. Otherwise, leaving a loan in delinquency and allowing it to default can, in the worst cases, cause a seizure of assets or wages.

In our project, we've got collected a bank data, and tried to fit a logistic model to predict whether a loan applied will get default or not, supported by some explanatory variables available in hand.

2 Objective of the Study

In this project, we wish to predict whether a customer is going to default his credit amount or not for the Test Dataset using Logistic Regression. The key steps which are involved in the process are as follows:

1. Dealing with the problem of Missing Values using suitable Data Imputation Techniques.

2. Building the model using Logistic Regression.

3. Checking the accuracy of the model.

3 Methodology

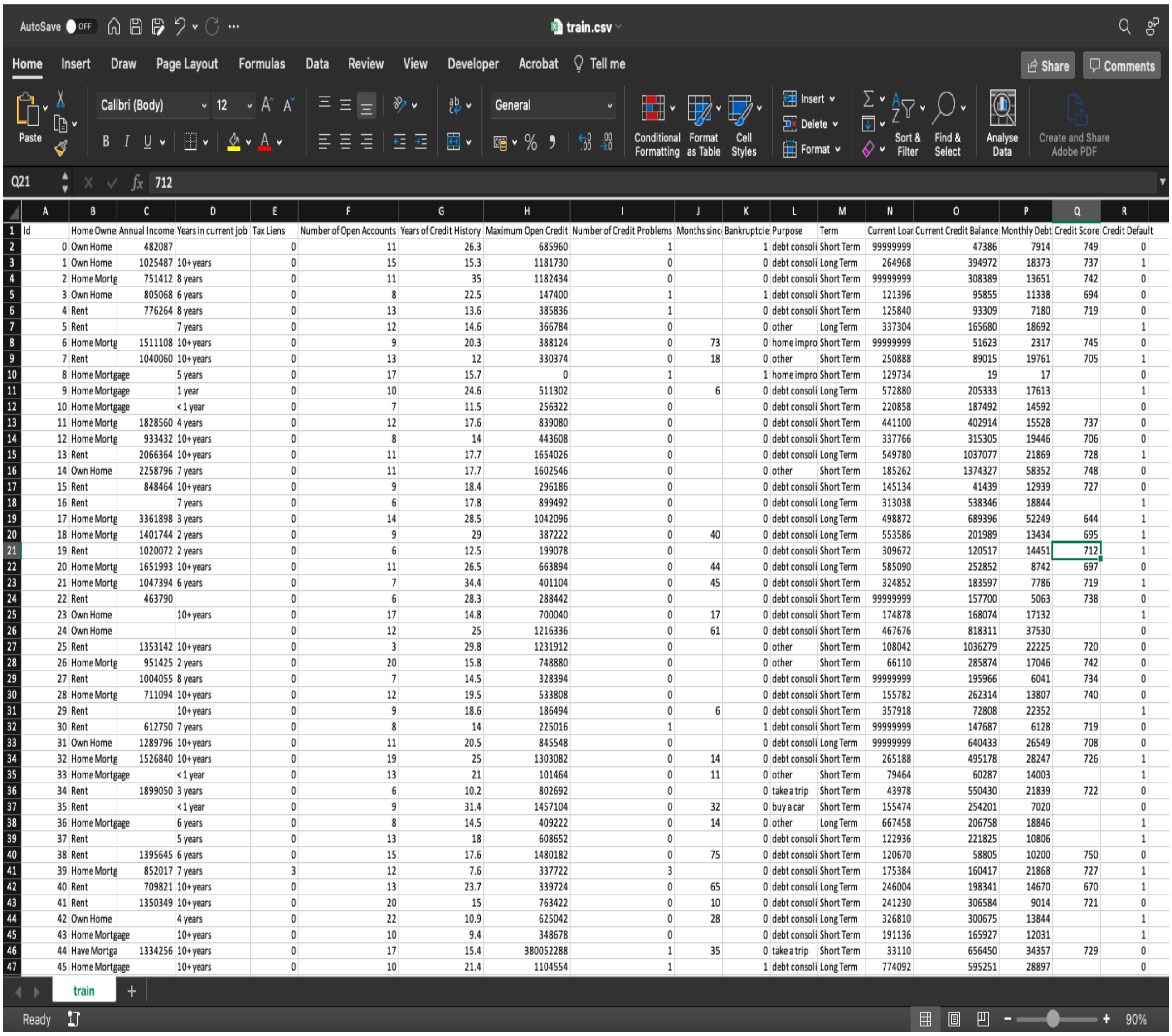
In the previous section, we formally introduced the problem statement of loan default prediction. In this section, we explain our step-by-step procedure of how we have achieved our results for predicting loan defaults. We first introduce the Loan Default dataset which we have used and explain the terminologies of the variables in the dataset like tax liens, credit score, credit default, etc. We then state our data preprocessing steps, where we state the problems present within the data like missing data, for instance, and explain how we dealt with them. Next, we introduce the logistic model we have considered and explain how we train our data using this model. Later, we analyze and evaluate the performance of the fitted model by various methods.

4 Dataset description

The Loan default dataset we have used in this study has been collected from Kaggle which is a hub of public datasets.

4.1 Viewing dataset :

Our train dataset looks like :-

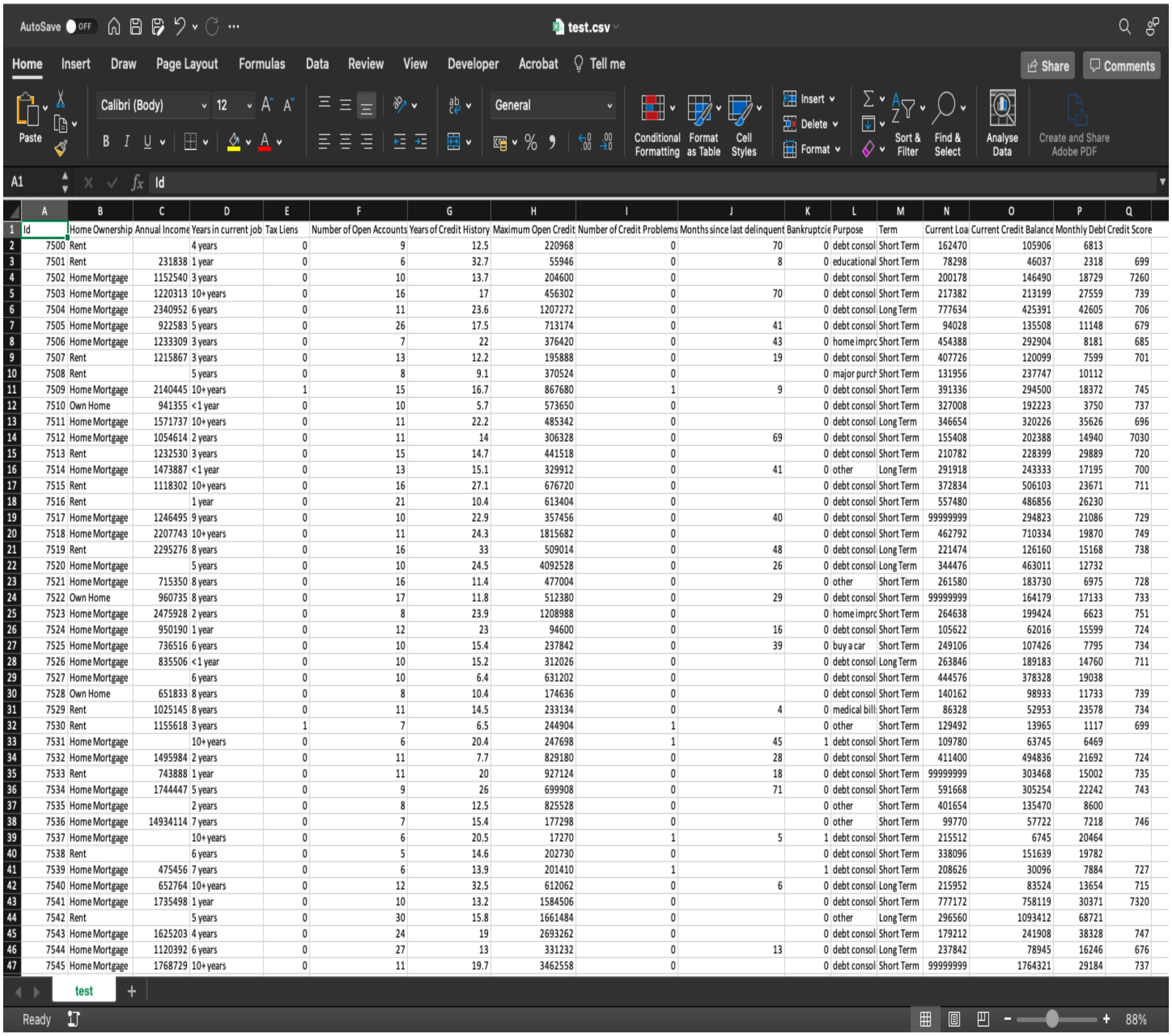


Q21 712

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Id	Home Owne	Annual Income	Years in current job	Tax Liens	Number of Open Accounts	Years of Credit History	Maximum Open Credit	Number of Credit Problems	Months sinc	Bankruptcie	Purpose	Term	Current Loar	Current Credit Balance	Monthly Debt	Credit Score	Credit Default
2	0	Own Home	482087		0	11	26.3	685960	1		1	debt consoli	Short Term	99999999	47386	7914	749	0
3	1	Own Home	1025487	10+ years	0	15	15.3	1181730	0		0	debt consoli	Long Term	264968	394972	18373	737	1
4	2	Home Mortg	751412	8 years	0	11	35	1182434	0		0	debt consoli	Short Term	99999999	308389	13651	742	0
5	3	Own Home	805068	6 years	0	8	22.5	147400	1		1	debt consoli	Short Term	121396	95855	11338	694	0
6	4	Rent	776264	8 years	0	13	13.6	385836	1		0	debt consoli	Short Term	125840	93309	7180	719	0
7	5	Rent		7 years	0	12	14.6	366784	0		0	other	Long Term	337304	165680	18692		1
8	6	Home Mortg	1511108	10+ years	0	9	20.3	388124	0	73	0	home impro	Short Term	99999999	51623	2317	745	0
9	7	Rent	1040060	10+ years	0	13	12	330374	0	18	0	other	Short Term	250888	89015	19761	705	1
10	8	Home Mortgage		5 years	0	17	15.7	0	1		1	home impro	Short Term	129734	19	17		0
11	9	Home Mortgage		1 year	0	10	24.6	511302	0	6	0	debt consoli	Long Term	572880	205333	17613		1
12	10	Home Mortgage		< 1 year	0	7	11.5	256322	0		0	debt consoli	Short Term	220858	187492	14592		0
13	11	Home Mortg	1828560	4 years	0	12	17.6	839080	0		0	debt consoli	Short Term	441100	402914	15528	737	0
14	12	Home Mortg	933432	10+ years	0	8	14	443608	0		0	debt consoli	Short Term	337766	315305	19446	706	0
15	13	Rent	2066364	10+ years	0	11	17.7	1654026	0		0	debt consoli	Long Term	549780	1037077	21869	728	1
16	14	Own Home	2258796	7 years	0	11	17.7	1602546	0		0	other	Short Term	185262	1374327	58352	748	0
17	15	Rent	848464	10+ years	0	9	18.4	296186	0		0	debt consoli	Short Term	145134	41439	12939	727	0
18	16	Rent		7 years	0	6	17.8	899492	0		0	debt consoli	Long Term	313038	538346	18844		1
19	17	Home Mortg	3361898	3 years	0	14	28.5	1042096	0		0	debt consoli	Long Term	498872	689396	52249	644	1
20	18	Home Mortg	1401744	2 years	0	9	29	387222	0	40	0	debt consoli	Long Term	553586	201989	13434	695	1
21	19	Rent	1020072	2 years	0	6	12.5	199078	0		0	debt consoli	Short Term	309672	120517	14451	712	1
22	20	Home Mortg	1651993	10+ years	0	11	26.5	663894	0	44	0	debt consoli	Long Term	585090	252852	8742	697	0
23	21	Home Mortg	1047394	6 years	0	7	34.4	401104	0	45	0	debt consoli	Short Term	324852	183597	7786	719	1
24	22	Rent	463790		0	6	28.3	288442	0		0	debt consoli	Short Term	99999999	157700	5063	738	0
25	23	Own Home		10+ years	0	17	14.8	700040	0	17	0	debt consoli	Short Term	174878	168074	17132		1
26	24	Own Home			0	12	25	1216336	0	61	0	debt consoli	Short Term	467676	818311	37530		0
27	25	Rent	1353142	10+ years	0	3	29.8	1231912	0		0	other	Short Term	108042	1036279	22225	720	0
28	26	Home Mortg	951425	2 years	0	20	15.8	748880	0		0	other	Short Term	66110	285874	17046	742	0
29	27	Rent	1004055	8 years	0	7	14.5	328394	0		0	debt consoli	Short Term	99999999	195966	6041	734	0
30	28	Home Mortg	711094	10+ years	0	12	19.5	533808	0		0	debt consoli	Short Term	155782	262314	13807	740	0
31	29	Rent		10+ years	0	9	18.6	186494	0	6	0	debt consoli	Short Term	357918	72808	22352		1
32	30	Rent	612750	7 years	0	8	14	225016	1		1	debt consoli	Short Term	99999999	147687	6128	719	0
33	31	Own Home	1289796	10+ years	0	11	20.5	845548	0		0	debt consoli	Long Term	99999999	640433	26549	708	0
34	32	Home Mortg	1526840	10+ years	0	19	25	1303082	0	14	0	debt consoli	Short Term	265188	495178	28247	726	1
35	33	Home Mortgage		< 1 year	0	13	21	101464	0	11	0	other	Short Term	79464	60287	14003		1
36	34	Rent	1899050	3 years	0	6	10.2	802692	0		0	take a trip	Short Term	43978	550430	21839	722	0
37	35	Rent		< 1 year	0	9	31.4	1457104	0	32	0	buy a car	Short Term	155474	254201	7020		0
38	36	Home Mortgage		6 years	0	8	14.5	409222	0	14	0	other	Long Term	667458	206758	18846		1
39	37	Rent		5 years	0	13	18	608652	0		0	debt consoli	Short Term	122936	221825	10806		1
40	38	Rent	1395645	6 years	0	15	17.6	1480182	0	75	0	debt consoli	Short Term	120670	58805	10200	750	0
41	39	Home Mortg	852017	7 years	3	12	7.6	337722	3		0	debt consoli	Short Term	175384	160417	21868	727	1
42	40	Rent	709821	10+ years	0	13	23.7	339724	0	65	0	debt consoli	Long Term	246004	198341	14670	670	1
43	41	Rent	1350349	10+ years	0	20	15	763422	0	10	0	debt consoli	Short Term	241230	306584	9014	721	0
44	42	Own Home		4 years	0	22	10.9	625042	0	28	0	debt consoli	Long Term	326810	300675	13844		1
45	43	Home Mortgage		10+ years	0	10	9.4	348678	0		0	debt consoli	Short Term	191136	165927	12031		1
46	44	Have Mortga	1334256	10+ years	0	17	15.4	380052288	1	35	0	take a trip	Short Term	33110	656450	34357	729	0
47	45	Home Mortgage		10+ years	0	10	21.4	1104554	1		1	debt consoli	Long Term	774092	595251	28897		0

Figure 1: Snapshot of train dataset

Our test dataset looks like :-



Id	Home Ownership	Annual Income	Years in current job	Tax Liens	Number of Open Accounts	Years of Credit History	Maximum Open Credit	Number of Credit Problems	Months since last delinquent	Bankruptcie	Purpose	Term	Current Loa	Current Credit Balance	Monthly Debt	Credit Score
7500	Rent		4 years	0	9	12.5	220968	0	70	0	debt consol	Short Term	162470	105906	6813	
7501	Rent	231838	1 year	0	6	32.7	55946	0	8	0	educational	Short Term	78298	46037	2318	699
7502	Home Mortgage	1152540	3 years	0	10	13.7	204600	0		0	debt consol	Short Term	200178	146490	18729	7260
7503	Home Mortgage	1220313	10+ years	0	16	17	456302	0	70	0	debt consol	Short Term	217382	213199	27559	739
7504	Home Mortgage	2340952	6 years	0	11	23.6	1207272	0		0	debt consol	Long Term	777634	425391	42605	706
7505	Home Mortgage	922583	5 years	0	26	17.5	713174	0	41	0	debt consol	Short Term	94028	135508	11148	679
7506	Home Mortgage	1233309	3 years	0	7	22	376420	0	43	0	home imprc	Short Term	454388	292904	8181	685
7507	Rent	1215867	3 years	0	13	12.2	195888	0	19	0	debt consol	Short Term	407726	120099	7599	701
7508	Rent		5 years	0	8	9.1	370524	0		0	major purch	Short Term	131956	237747	10112	
7509	Home Mortgage	2140445	10+ years	1	15	16.7	867680	1	9	0	debt consol	Short Term	391336	294500	18372	745
7510	Own Home	941355	< 1 year	0	10	5.7	573650	0		0	debt consol	Short Term	327008	192223	3750	737
7511	Home Mortgage	1571737	10+ years	0	11	22.2	485342	0		0	debt consol	Long Term	346654	320226	35626	696
7512	Home Mortgage	1054614	2 years	0	11	14	306328	0	69	0	debt consol	Short Term	155408	202388	14940	7030
7513	Rent	1232530	3 years	0	15	14.7	441518	0		0	debt consol	Short Term	210782	228399	29889	720
7514	Home Mortgage	1473887	< 1 year	0	13	15.1	329912	0	41	0	other	Long Term	291918	243333	17195	700
7515	Rent	1118302	10+ years	0	16	27.1	676720	0		0	debt consol	Short Term	372834	506103	23671	711
7516	Rent		1 year	0	21	10.4	613404	0		0	debt consol	Short Term	557480	486856	26230	
7517	Home Mortgage	1246495	9 years	0	10	22.9	357456	0	40	0	debt consol	Short Term	99999999	294823	21086	729
7518	Home Mortgage	2207743	10+ years	0	11	24.3	1815682	0		0	debt consol	Short Term	462792	710334	19870	749
7519	Rent	2295276	8 years	0	16	33	509014	0	48	0	debt consol	Long Term	221474	126160	15168	738
7520	Home Mortgage		5 years	0	10	24.5	4092528	0	26	0	debt consol	Long Term	344476	463011	12732	
7521	Home Mortgage	715350	8 years	0	16	11.4	477004	0		0	other	Short Term	261580	183730	6975	728
7522	Own Home	960735	8 years	0	17	11.8	512380	0	29	0	debt consol	Short Term	99999999	164179	17133	733
7523	Home Mortgage	2475928	2 years	0	8	23.9	1208988	0		0	home imprc	Short Term	264638	199424	6623	751
7524	Home Mortgage	950190	1 year	0	12	23	94600	0	16	0	debt consol	Short Term	105622	62016	15599	724
7525	Home Mortgage	736516	6 years	0	10	15.4	237842	0	39	0	buy a car	Short Term	249106	107426	7795	734
7526	Home Mortgage	835506	< 1 year	0	10	15.2	312026	0		0	debt consol	Long Term	263846	189183	14760	711
7527	Home Mortgage		6 years	0	10	6.4	631202	0		0	debt consol	Short Term	444576	378328	19038	
7528	Own Home	651833	8 years	0	8	10.4	174636	0		0	debt consol	Short Term	140162	98933	11733	739
7529	Rent	1025145	8 years	0	11	14.5	233134	0	4	0	medical bill	Short Term	86328	52953	23578	734
7530	Rent	1155618	3 years	1	7	6.5	244904	1		0	other	Short Term	129492	13965	1117	699
7531	Home Mortgage		10+ years	0	6	20.4	247698	1	45	1	debt consol	Short Term	109780	63745	6469	
7532	Home Mortgage	1495984	2 years	0	11	7.7	829180	0	28	0	debt consol	Short Term	411400	494836	21692	724
7533	Rent	743888	1 year	0	11	20	927124	0	18	0	debt consol	Short Term	99999999	303468	15002	735
7534	Home Mortgage	1744447	5 years	0	9	26	699908	0	71	0	debt consol	Short Term	591668	305254	22242	743
7535	Home Mortgage		2 years	0	8	12.5	825528	0		0	other	Short Term	401654	135470	8600	
7536	Home Mortgage	14934114	7 years	0	7	15.4	177298	0		0	other	Short Term	99770	57722	7218	746
7537	Home Mortgage		10+ years	0	6	20.5	17270	1	5	1	debt consol	Short Term	215512	6745	20464	
7538	Rent		6 years	0	5	14.6	202730	0		0	debt consol	Short Term	338096	151639	19782	
7539	Home Mortgage	475456	7 years	0	6	13.9	201410	1		1	debt consol	Short Term	208626	30096	7884	727
7540	Home Mortgage	652764	10+ years	0	12	32.5	612062	0	6	0	debt consol	Long Term	215952	83524	13654	715
7541	Home Mortgage	1735498	1 year	0	10	13.2	1584506	0		0	debt consol	Short Term	777172	758119	30371	7320
7542	Rent		5 years	0	30	15.8	1661484	0		0	other	Long Term	296560	1093412	68721	
7543	Home Mortgage	1625203	4 years	0	24	19	2693262	0		0	debt consol	Short Term	179212	241908	38328	747
7544	Home Mortgage	1120392	6 years	0	27	13	331232	0	13	0	debt consol	Long Term	237842	78945	16246	676
7545	Home Mortgage	1768729	10+ years	0	11	19.7	3462558	0		0	debt consol	Short Term	99999999	1764321	29184	737

Figure 2: Snapshot of test dataset

4.1.1 Viewing dataset in R :

```
#Setting the working directory
setwd("C:/Users/user/Dropbox")
#Fetching the train dataset
train_data=read.csv("train.csv")
#Head of the dataset showing first 6 rows of the train dataset
head(train_data)

##   Id Home.Ownership Annual.Income Years.in.current.job Tax.Liens
## 1  0       Own Home        482087                              0
## 2  1       Own Home       1025487            10+ years         0
## 3  2  Home Mortgage        751412              8 years         0
## 4  3       Own Home        805068              6 years         0
## 5  4           Rent        776264              8 years         0
## 6  5           Rent            NA              7 years         0
##   Number.of.Open.Accounts Years.of.Credit.History Maximum.Open.Credit
## 1                      11                    26.3              685960
## 2                      15                    15.3             1181730
## 3                      11                    35.0             1182434
## 4                       8                    22.5              147400
## 5                      13                    13.6              385836
## 6                      12                    14.6              366784
##   Number.of.Credit.Problems Months.since.last.delinquent Bankruptcies
## 1                         1                           NA            1
## 2                         0                           NA            0
## 3                         0                           NA            0
## 4                         1                           NA            1
## 5                         1                           NA            0
## 6                         0                           NA            0
##              Purpose       Term Current.Loan.Amount Current.Credit.Balance
## 1 debt consolidation Short Term            99999999                  47386
## 2 debt consolidation  Long Term              264968                 394972
## 3 debt consolidation Short Term            99999999                 308389
## 4 debt consolidation Short Term              121396                  95855
## 5 debt consolidation Short Term              125840                  93309
## 6              other  Long Term              337304                 165680
##   Monthly.Debt Credit.Score Credit.Default
## 1         7914          749              0
## 2        18373          737              1
## 3        13651          742              0
## 4        11338          694              0
## 5         7180          719              0
## 6        18692           NA              1
```

```
#Showing the dimension of the dataset

dim(train_data)

## [1] 7500   18
```

4.2 Interpretation of the Variables :

The variables in the dataset are described as follows:

- **Id:** It refers to the Customer Id of the customer taking the loan.
- **Home Ownership:** Home ownership refers to the information on whether the home, in which the loan applicant is currenty residing, is owned by him or it is rented or under mortgage .
- **Annual Income:**It refers to the total income of the customer during a financial year.
- **Years in current job:** It refers to the number of years the customer has been in his/her present job.
- **Tax Liens:** It refers to the no. of times a customer was penalized for failure of tax payment.
- **Number of Open Accounts:** The number of accounts (open) for a particular customer.
- **Years of Credit History:** The time span covering the issue of the first loan to the closure of the last loan of a customer.
- **Maximum Open Credit:** It is the maximum amount of credit available to the customer. The limit is revisable, and the borrower can request an increase in the maximum credit limit if the limit is not enough for their needs.
- **Number of Credit Problems:** The number of times a customer has experienced breakdown of his/her financial system caused by a sudden and severe disruption of the normal process of cash movement.
- **Months since last delinquent:** Months since last payment failure/delay.
- **Bankruptcies:** The number of times a customer has faced the issue of insolvency.
- **Purpose:** The purpose for taking the loan as stated by the customer.
- **Term:** The category of time period for which the loan has been taken - short term or long term.
- **Current Loan Amount:** The total amount of the ongoing loan the customer has taken.
- **Current Credit Balance:** The amount of credit which the customer is yet to pay back. It is basically Total Loan amount - Total amount that has been repayed.

- **Monthly Debt:** It refers to the equated monthly installment (EMI), which are payments made regularly to repay an outstanding loan within a certain time frame.
- **Credit Score:** It is an indicator of a person's creditworthiness, or their ability to repay debt.
- **Credit Default:** A credit default occurs when a borrower is unable to make timely payments, misses payments, or avoids or stops making payments on interest or principal owed.

4.2.1 Exploring data structure in R :

```
#Showing the data types of the different variables
str(train_data)

## 'data.frame': 7500 obs. of  18 variables:
##  $ Id                          : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Home.Ownership              : chr  "Own Home" "Own Home" "Home Mortgage" "Own Home" ...
##  $ Annual.Income               : int  482087 1025487 751412 805068 776264 NA 1511108 1040060 NA NA ...
##  $ Years.in.current.job        : chr  "" "10+ years" "8 years" "6 years" ...
##  $ Tax.Liens                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Number.of.Open.Accounts     : int  11 15 11 8 13 12 9 13 17 10 ...
##  $ Years.of.Credit.History     : num  26.3 15.3 35 22.5 13.6 14.6 20.3 12 15.7 24.6 ...
##  $ Maximum.Open.Credit         : int  685960 1181730 1182434 147400 385836 366784 388124 330374 0 51130
##  $ Number.of.Credit.Problems   : int  1 0 0 1 1 0 0 0 1 0 ...
##  $ Months.since.last.delinquent: int  NA NA NA NA NA NA 73 18 NA 6 ...
##  $ Bankruptcies                : int  1 0 0 1 0 0 0 0 1 0 ...
##  $ Purpose                     : chr  "debt consolidation" "debt consolidation" "debt consolidation" "d
##  $ Term                        : chr  "Short Term" "Long Term" "Short Term" "Short Term" ...
##  $ Current.Loan.Amount         : int  99999999 264968 99999999 121396 125840 337304 99999999 250888 129
##  $ Current.Credit.Balance      : int  47386 394972 308389 95855 93309 165680 51623 89015 19 205333 ...
##  $ Monthly.Debt                : int  7914 18373 13651 11338 7180 18692 2317 19761 17 17613 ...
##  $ Credit.Score                : int  749 737 742 694 719 NA 745 705 NA NA ...
##  $ Credit.Default              : int  0 1 0 0 0 1 0 1 0 1 ...

#Here "int" refers  to the integer type data, "chr" refers to the
#character type data and "num" refers to the numeric type data
```

5 Dealing with missing data

Missing data needs to be treated since it triggers 3 main problems:

1. It can introduce a substantial amount of bias in the model.

2. It makes the handling and analysis of the data more difficult.

3. Efficiency can be reduced because of this as a consequence.

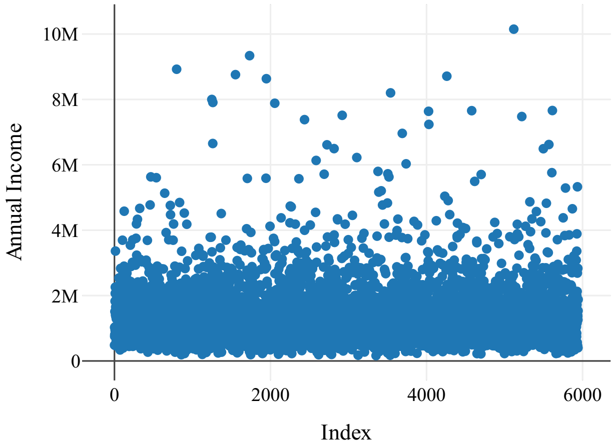
Dropping all the rows with missing values, introduces bias and affects representativeness of the results. So, we need to impute the missing data using suitable imputation techniques. The process of replacing missing data with substituted values is known as imputation and it preserves all the cases by replacing missing data with an estimated value, based on other available information.

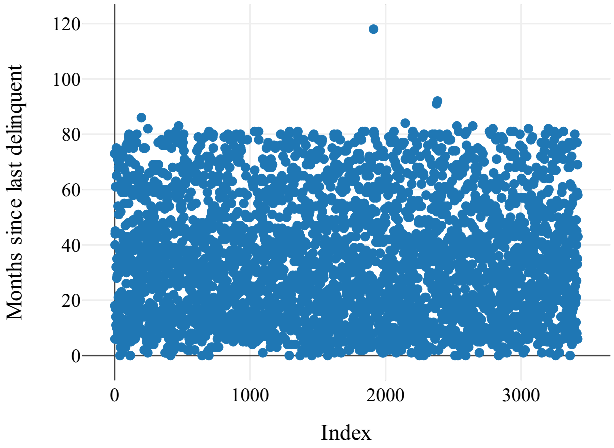
The no. of missing values of our dataset is shown in the following R output :

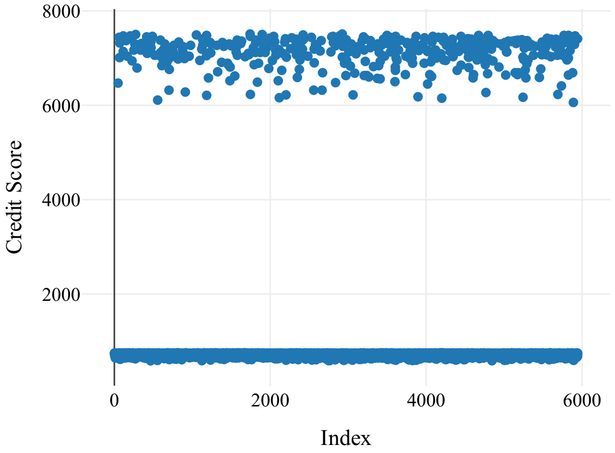
```
colSums(is.na(train_data))

##                           Id               Home.Ownership
##                            0                            0
##                Annual.Income         Years.in.current.job
##                         1557                            0
##                    Tax.Liens      Number.of.Open.Accounts
##                            0                            0
##      Years.of.Credit.History          Maximum.Open.Credit
##                            0                            0
##    Number.of.Credit.Problems Months.since.last.delinquent
##                            0                         4081
##                 Bankruptcies                      Purpose
##                           14                            0
##                         Term          Current.Loan.Amount
##                            0                            0
##       Current.Credit.Balance                 Monthly.Debt
##                            0                            0
##                 Credit.Score               Credit.Default
##                         1557                            0
```

From the above R output we can see that the no. of missing values in the variables - Annual Income, Months since last delinquent, Bankruptcies and Credit Score are 1557, 4081, 14 and 1557 respectively.

Here the no. of missing values corresponding to the variable Bankruptcies is 14, which is too small compared to the whole dataset of 7500 observations. So we will discard the corresponding rows with missing values for this variable, and plot the other 3 variables with missing values.







From the plotted diagram of the variable Credit Score, it is clear that the values of the variable are highly bipolarised; so we can't apply any kind of imputation technique for the missing values of Credit Score. So, the corresponding rows with these missing values will also be discarded.

For the other two variables, we explored 2 techniques of imputation:

1. Mean Imputation

2. Median Imputation

5.1 Mean Imputation

Mean imputation technique is the process of replacing any missing value in the data with the mean of that variable in context. In our dataset, we replaced a missing value of a variable, with the mean of the other non-missing values of that feature. Mean imputation attenuates any correlations involving the variable(s) that are imputed. This is because, in cases with imputation, there is guaranteed to be no relationship between the imputed variable and any other measured variables. Thus, mean imputation has some attractive properties for univariate analysis but becomes problematic for multivariate analysis.

From the plot of the variable Months since last delinquent, it is evident that mean imputation will be appropiate for this variable. The following R code has been used for this purpose :

```
train_data$Months.since.last.delinquent[is.na(train_data$Months.since.last.delinquent)]=mean(
+ train_data$Months.since.last.delinquent,na.rm=T)
```

5.2 Median Imputation

In Median Imputation technique, one replaces the missing values with the median of the available values of the same variable. It is used mainly when the data is skewed. Also, median imputation technique is used when there is outliers in the data.

From the plot of the Annual Income, we can see that the observations are very dense for the lower income group whereas there are few observations for the higher income group. So, here we have applied the median imputation. The following R code has been used for this purpose :

```
train_data$Annual.Income[is.na(train_data$Annual.Income)]=median(train_data$Annual.Income,na.rm=T)
```

After performing the mean and median imputation, we discard the missing values of the variables Bankruptcies and Credit Score as said earlier; the following R code has been used for this purpose :

```
train_data=na.omit(train_data)
```

Now, let us check whether we are left with any missing value in our data or not, using the following R code :

```
colSums((is.na(train_data)))

##                           Id               Home.Ownership
##                            0                            0
##                Annual.Income         Years.in.current.job
##                            0                            0
##                    Tax.Liens      Number.of.Open.Accounts
##                            0                            0
##      Years.of.Credit.History          Maximum.Open.Credit
##                            0                            0
##    Number.of.Credit.Problems Months.since.last.delinquent
##                            0                            0
##                 Bankruptcies                      Purpose
##                            0                            0
##                         Term          Current.Loan.Amount
##                            0                            0
##       Current.Credit.Balance                 Monthly.Debt
##                            0                            0
##                 Credit.Score               Credit.Default
##                            0                            0
```

So, now our data has been cleaned and we can proceed our further analysis with this dataset.

6 Exploratory Data Analysis (EDA)

An exploratory data analysis is always helpful to get an insight about the data. So here we will try to visualise different variables by various plots and diagrams and try to analyze them.

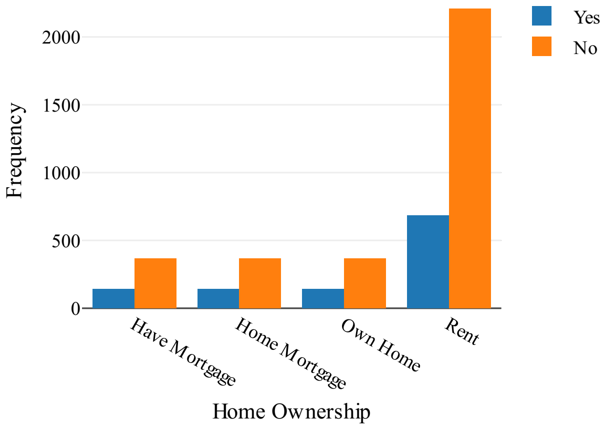
6.1 Plot of Home Ownership and Credit Default

Credit Defaults(Yes/No) for different categories of Home Ownership is shown in the following table from R output:

```
##                           train_data$Credit.Default
## train_data$Home.Ownership    0    1
##             Have Mortgage    7    1
##             Home Mortgage 2210  686
##             Own Home       368  143
##             Rent          1766  750
```

From the above table, we can see that in the category Have Mortgage, there are 7 cases of non-default and 1 case of credit default; in the category Home Mortgage, there are 2210 cases of non-default and 686 cases of credit default and so on.

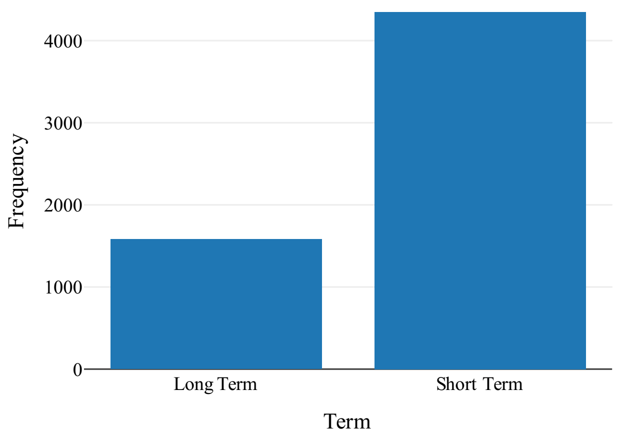
From the above table, the following plot is made :



From the above diagram we can see that the no. of default and non default cases in the categories Have Mortgage, Home Mortgage, and Own Home are very close to each other; whereas the no. of non defaults is significantly high compared to the no. of defaults for the category Rent.

6.2 Plot of Term

To see the frequency of short term and long term loans, we plot the following bar diagram:



We can see that the short term loans are more frequent than the long term loans.

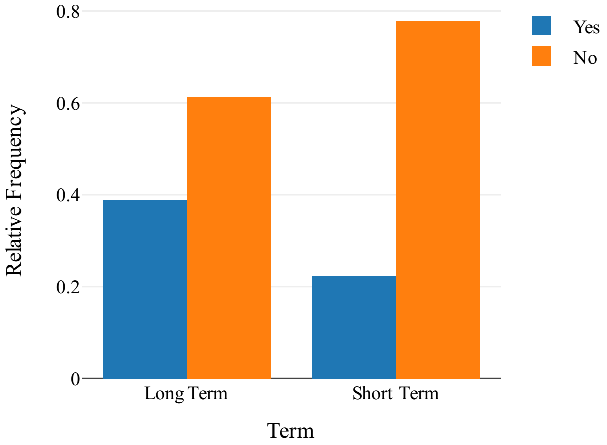
6.3 Plot of Term and Credit Default

Relative frequency of Credit Defaults(Yes/No) for Short term and Long term loans is shown in the following table from R output:

```
##             Credit.Default
## Term                 0         1
##   Long Term  0.6122707 0.3877293
##   Short Term 0.7777011 0.2222989
```

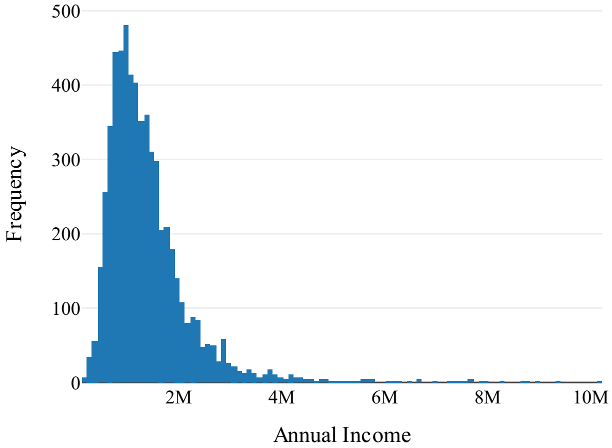
From the above table we can see that the relative frequencies of non defaults are high for both the short term and long term loans. Also, the relative frequency of credit defaults in case of short term loans is less than the relative frequency of credit defaults in case of long term loans.

The above interpretation can also be visualized through the following diagram :



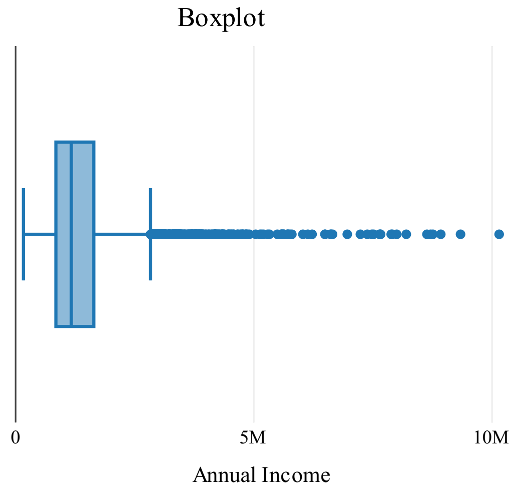
6.4 Plot of Annual Income

The frequency distribution of the variable Annual Income is shown in the following diagram :



The above plot depicts that the distribution of Annual Income is positively skewed. The frequency is highest near the Annual Income 1M.

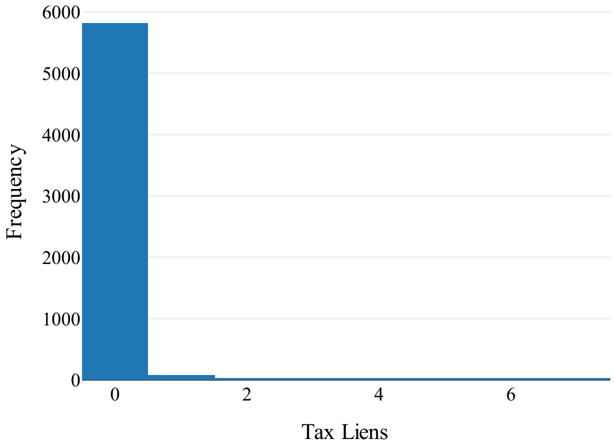
We also make a boxplot for the Annual Income variable :



From the above boxplot we can see that the distribution of Annual Income is positively skewed.

6.5 Plot of Tax Liens

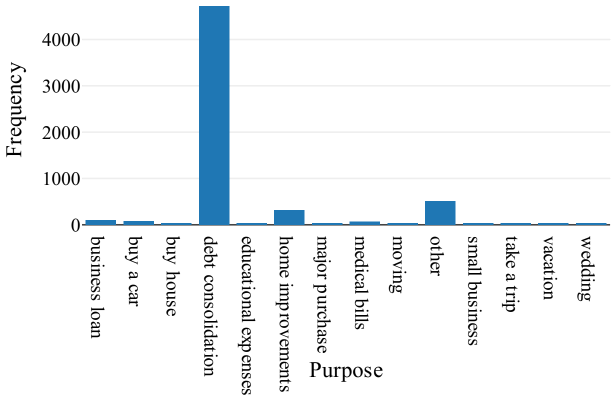
The frequency distribution of the variable Tax liens is shown in the following bar diagram :



From the above plot we can see that distribution is positively skewed and the case of '0' Tax Liens has the highest frequency.

6.6 Plot of Purpose

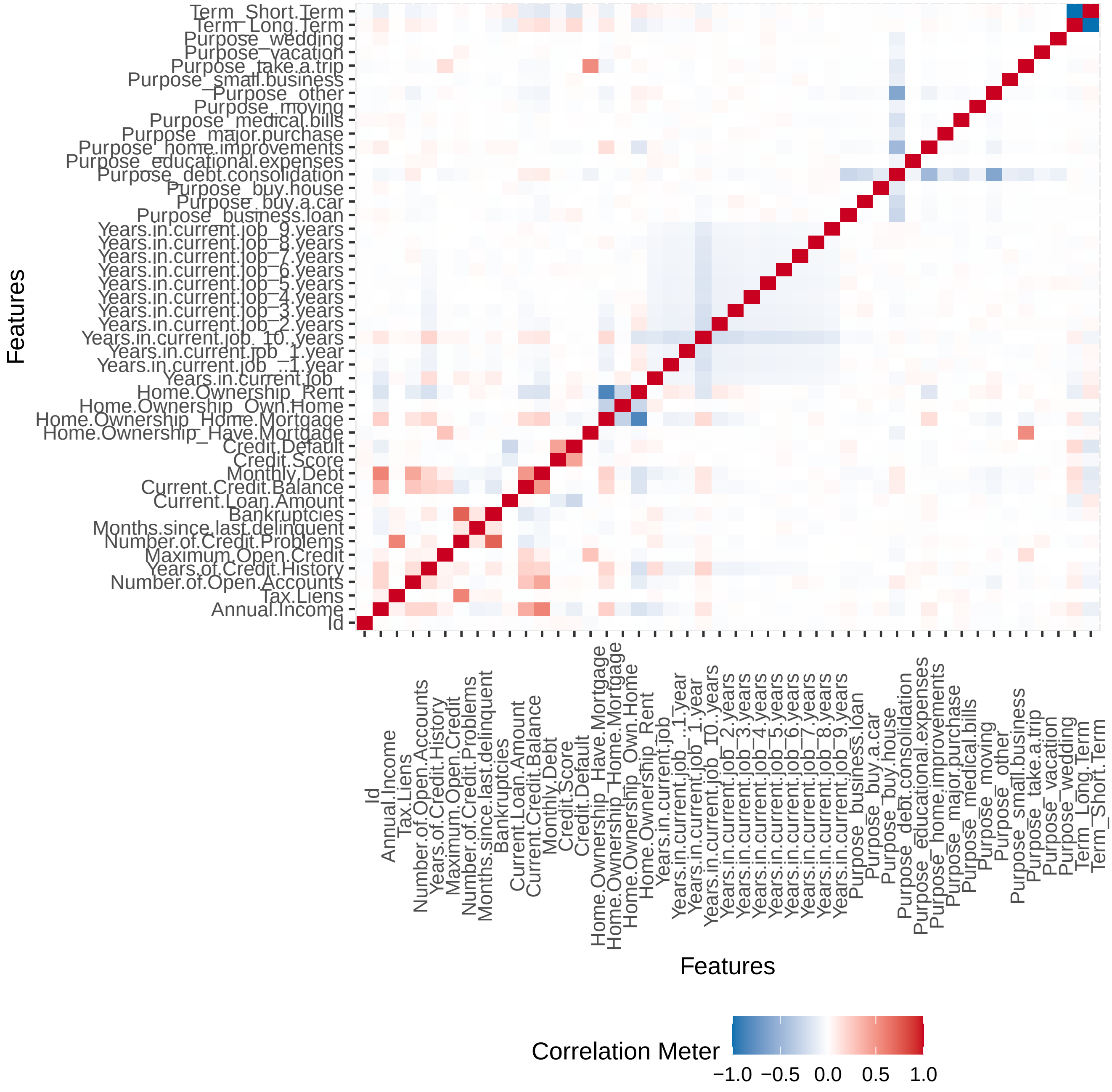
The frequencies of different purposes of taking loan is shown in the following bar diagram :



From the above plot it is clear that the most frequent purpose of taking loan is debt consolidation.

6.7 Correlation Heatmap of train data

The following correlation heatmap is showing the association among the different variables of our train dataset.



7 Multicollinearity

Now we will investigate whether the continuous regressors in our dataset are involved in multicollinearity or not.

In a regression problem with multiple regressors , multicollinearity refers to a near-linear relationship among the regressors. Multicollinearity may happen due to overspecification of model, bad data collection or sampling techniques, inclusion of too many higher order terms in a polynomial regression model etc. Multicollinearity has some serious consequences eg. exceptionally high value of parameter estimates, large variances of some parameter estimators.

Several multicollinearity diagnostic measures are available. Here we have used "Variance Inflation Factor" to detect multicollinearity among the continuous variables of our dataset. The variance inflation factor for the jth explanatory variable (when all the regressors are scaled to unit norm) is defined as:

$$VIF_j = \frac{1}{(1 - R_j^2)}$$

where R_j^2 denotes the coefficient of determination obtained when X_j is regressed on the remaining regressor variables.

In practice, usually a $VIF > 5$ indicates that the corresponding explanatory variable is involved in multicollinearity. Here we will use an iterative algorithm that drops variable with highest VIF and then checks VIF again and then drop until VIF of all variables is less than 5.

7.1 R output for Multicollinearity checking

The following R output shows VIFs of different continuous variables of our dataset.

```
library(regclass)
#Logistic regression with continuous regressors
colnames(select_if(train_data,is.numeric))

##  [1] "Id"                           "Annual.Income"
##  [3] "Tax.Liens"                    "Number.of.Open.Accounts"
##  [5] "Years.of.Credit.History"      "Maximum.Open.Credit"
##  [7] "Number.of.Credit.Problems"    "Months.since.last.delinquent"
##  [9] "Bankruptcies"                 "Current.Loan.Amount"
## [11] "Current.Credit.Balance"       "Monthly.Debt"
## [13] "Credit.Score"                 "Credit.Default"
```

```
model.cont=glm(Credit.Default~scale(Annual.Income)+scale(Years.of.Credit.History)+scale(Maximum.Open.Credi
summary(model.cont)
```

```
##
## Call:
## glm(formula = Credit.Default ~ scale(Annual.Income) + scale(Years.of.Credit.History) +
##     scale(Maximum.Open.Credit) + scale(Months.since.last.delinquent) +
##     scale(Current.Loan.Amount) + scale(Current.Credit.Balance) +
##     scale(Monthly.Debt) + scale(Credit.Score), family = binomial,
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0723  -0.7911  -0.6222   0.0644   3.3408
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -1.74619    0.27664  -6.312 2.75e-10 ***
## scale(Annual.Income)                -0.54479    0.06046  -9.011  < 2e-16 ***
## scale(Years.of.Credit.History)       0.03559    0.03549   1.003 0.315996
## scale(Maximum.Open.Credit)          -2.65017    0.68920  -3.845 0.000120 ***
## scale(Months.since.last.delinquent) -0.05013    0.03422  -1.465 0.142979
## scale(Current.Loan.Amount)          -2.43598    0.65019  -3.747 0.000179 ***
## scale(Current.Credit.Balance)        0.16980    0.07530   2.255 0.024148 *
## scale(Monthly.Debt)                  0.35530    0.05025   7.070 1.54e-12 ***
## scale(Credit.Score)                  1.72128    0.20695   8.317  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6875.7  on 5930  degrees of freedom
## Residual deviance: 5173.2  on 5922  degrees of freedom
## AIC: 5191.2
```

```
##
## Number of Fisher Scoring iterations: 10
```

```r
VIF(model.cont)
```

```
##                 scale(Annual.Income)      scale(Years.of.Credit.History)
##                             1.836466                            1.086024
##           scale(Maximum.Open.Credit) scale(Months.since.last.delinquent)
##                             2.921941                            1.007035
##           scale(Current.Loan.Amount)       scale(Current.Credit.Balance)
##                             1.002746                            3.284842
##                  scale(Monthly.Debt)                 scale(Credit.Score)
##                             2.050633                            1.012040
```

From the above R output we can observe that VIFs of all the continuous variables in our dataset are less than 5. So there is no multicollinearity issue and we can proceed with all these continuous variables.

8 Variable Selection

Variable selection is the method for selecting a subset of 'best' regressors from a pool of potential regressors.

After checking multicollinearity, we are left with all the continuous regressors of our dataset. We also have other categorical regressors in our hand. Now, to follow the principle of parsimony, i.e. include as few as regressors as possible to explain the response variability in efficient manner, we go for variable selection.

We have used Forward Selection, Backward Selection and Stepwise Selection method for the aforesaid purpose. We compare the different AIC values of the different models obtained from the different variable selection methods and continue with that set of regressors which yield the minimum AIC value.

8.1 R output for Forward Selection Method

```r
#Logistic Regression
train_data[sapply(train_data, is.character)] <- lapply(train_data[sapply(train_data, is.character)],as.fac
log.reg<-glm(Credit.Default~.,data=train_data,family = binomial)
step.model_for<- step(log.reg,direction = "forward")
```

```
## Start:  AIC=5046.2
## Credit.Default ~ Id + Home.Ownership + Annual.Income + Years.in.current.job +
##     Tax.Liens + Number.of.Open.Accounts + Years.of.Credit.History +
##     Maximum.Open.Credit + Number.of.Credit.Problems + Months.since.last.delinquent +
##     Bankruptcies + Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
```

```
summary(step.model_for)
```

```
##
## Call:
## glm(formula = Credit.Default ~ Id + Home.Ownership + Annual.Income +
##     Years.in.current.job + Tax.Liens + Number.of.Open.Accounts +
##     Years.of.Credit.History + Maximum.Open.Credit + Number.of.Credit.Problems +
##     Months.since.last.delinquent + Bankruptcies + Purpose + Term +
##     Current.Loan.Amount + Current.Credit.Balance + Monthly.Debt +
##     Credit.Score, family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6159  -0.7451  -0.5290   0.0359   3.5011
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -6.122e-01  1.279e+00  -0.479 0.632192
## Id                             9.370e-06  1.620e-05   0.578 0.563096
## Home.OwnershipHome Mortgage    1.063e+00  1.228e+00   0.866 0.386563
## Home.OwnershipOwn Home         1.113e+00  1.232e+00   0.903 0.366427
## Home.OwnershipRent             1.415e+00  1.226e+00   1.154 0.248473
## Annual.Income                 -6.618e-07  7.676e-08  -8.623  < 2e-16 ***
## Years.in.current.job< 1 year  -6.102e-01  1.971e-01  -3.095 0.001966 **
## Years.in.current.job1 year    -5.805e-01  2.035e-01  -2.853 0.004335 **
## Years.in.current.job10+ years -5.085e-01  1.595e-01  -3.188 0.001431 **
## Years.in.current.job2 years   -4.822e-01  1.866e-01  -2.584 0.009763 **
## Years.in.current.job3 years   -5.859e-01  1.933e-01  -3.030 0.002442 **
## Years.in.current.job4 years   -5.811e-01  2.059e-01  -2.822 0.004768 **
## Years.in.current.job5 years   -5.199e-01  1.981e-01  -2.625 0.008675 **
## Years.in.current.job6 years   -3.011e-01  2.065e-01  -1.458 0.144799
## Years.in.current.job7 years   -8.662e-01  2.230e-01  -3.885 0.000102 ***
## Years.in.current.job8 years   -3.725e-01  2.160e-01  -1.725 0.084567 .
## Years.in.current.job9 years   -4.662e-01  2.468e-01  -1.889 0.058844 .
## Tax.Liens                     -7.985e-02  2.111e-01  -0.378 0.705165
## Number.of.Open.Accounts        2.254e-02  8.386e-03   2.687 0.007200 **
## Years.of.Credit.History        3.675e-03  5.498e-03   0.668 0.503841
## Maximum.Open.Credit           -5.568e-07  1.288e-07  -4.321 1.55e-05 ***
## Number.of.Credit.Problems      1.786e-01  1.657e-01   1.078 0.280977
## Months.since.last.delinquent  -3.939e-03  2.418e-03  -1.629 0.103283
## Bankruptcies                  -1.926e-01  1.927e-01  -0.999 0.317554
## Purposebuy a car              -9.928e-01  3.928e-01  -2.527 0.011491 *
## Purposebuy house              -8.615e-01  6.076e-01  -1.418 0.156222
## Purposedebt consolidation     -1.162e+00  2.442e-01  -4.759 1.94e-06 ***
## Purposeeducational expenses   -1.680e+00  1.205e+00  -1.395 0.163128
## Purposehome improvements      -1.013e+00  2.892e-01  -3.502 0.000462 ***
## Purposemajor purchase         -1.087e+00  5.401e-01  -2.012 0.044177 *
## Purposemedical bills          -7.125e-01  4.090e-01  -1.742 0.081538 .
## Purposemoving                 -4.676e+00  3.381e+00  -1.383 0.166736
## Purposeother                  -9.548e-01  2.662e-01  -3.587 0.000335 ***
## Purposesmall business          7.694e-02  5.955e-01   0.129 0.897201
## Purposetake a trip            -6.179e-01  6.156e-01  -1.004 0.315486
## Purposevacation               -1.296e+00  1.259e+00  -1.029 0.303357
## Purposewedding                -8.778e-01  8.670e-01  -1.012 0.311339
## TermShort Term                -9.185e-01  7.661e-02 -11.989  < 2e-16 ***
## Current.Loan.Amount           -7.960e-08  3.204e-08  -2.484 0.012983 *
## Current.Credit.Balance         6.732e-07  2.552e-07   2.638 0.008338 **
## Monthly.Debt                   2.535e-05  4.667e-06   5.432 5.56e-08 ***
## Credit.Score                   1.233e-03  1.980e-04   6.224 4.84e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 6875.7  on 5930  degrees of freedom
## Residual deviance: 4962.2  on 5889  degrees of freedom
## AIC: 5046.2
##
## Number of Fisher Scoring iterations: 11
```

8.2 R output for Backward Selection Method

```
step.model_back<- step(log.reg,direction = "backward")
```

```
## Start:  AIC=5046.2
## Credit.Default ~ Id + Home.Ownership + Annual.Income + Years.in.current.job +
##     Tax.Liens + Number.of.Open.Accounts + Years.of.Credit.History +
##     Maximum.Open.Credit + Number.of.Credit.Problems + Months.since.last.delinquent +
##     Bankruptcies + Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                                Df Deviance    AIC
## - Tax.Liens                     1   4962.3 5044.3
## - Id                            1   4962.5 5044.5
## - Years.of.Credit.History       1   4962.6 5044.6
## - Years.in.current.job         11   4982.7 5044.7
## - Bankruptcies                  1   4963.2 5045.2
## - Number.of.Credit.Problems     1   4963.3 5045.3
## <none>                              4962.2 5046.2
## - Months.since.last.delinquent  1   4964.9 5046.9
## - Current.Credit.Balance        1   4969.2 5051.2
## - Number.of.Open.Accounts       1   4969.4 5051.4
## - Purpose                      13   4995.4 5053.4
## - Home.Ownership                3   4984.5 5062.5
## - Maximum.Open.Credit           1   4988.8 5070.8
## - Monthly.Debt                  1   4991.7 5073.7
## - Annual.Income                 1   5052.2 5134.2
## - Term                          1   5105.1 5187.1
## - Current.Loan.Amount           1   5370.5 5452.5
## - Credit.Score                  1   5920.4 6002.4
##
## Step:  AIC=5044.34
## Credit.Default ~ Id + Home.Ownership + Annual.Income + Years.in.current.job +
##     Number.of.Open.Accounts + Years.of.Credit.History + Maximum.Open.Credit +
##     Number.of.Credit.Problems + Months.since.last.delinquent +
##     Bankruptcies + Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                                Df Deviance    AIC
## - Id                            1   4962.7 5042.7
## - Years.in.current.job         11   4982.8 5042.8
## - Years.of.Credit.History       1   4962.8 5042.8
## - Bankruptcies                  1   4963.3 5043.3
## - Number.of.Credit.Problems     1   4964.0 5044.0
## <none>                              4962.3 5044.3
## - Months.since.last.delinquent  1   4965.0 5045.0
## - Current.Credit.Balance        1   4969.3 5049.3
## - Number.of.Open.Accounts       1   4969.5 5049.5
## - Purpose                      13   4995.5 5051.5
## - Home.Ownership                3   4984.5 5060.5
## - Maximum.Open.Credit           1   4989.0 5069.0
## - Monthly.Debt                  1   4991.9 5071.9
## - Annual.Income                 1   5052.6 5132.6
## - Term                          1   5105.3 5185.3
## - Current.Loan.Amount           1   5370.5 5450.5
## - Credit.Score                  1   5920.5 6000.5
##
## Step:  AIC=5042.69
## Credit.Default ~ Home.Ownership + Annual.Income + Years.in.current.job +
##     Number.of.Open.Accounts + Years.of.Credit.History + Maximum.Open.Credit +
##     Number.of.Credit.Problems + Months.since.last.delinquent +
##     Bankruptcies + Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                               Df Deviance    AIC
## - Years.in.current.job        11   4983.1 5041.1
## - Years.of.Credit.History       1   4963.1 5041.1
## - Bankruptcies                  1   4963.7 5041.7
## - Number.of.Credit.Problems     1   4964.3 5042.3
## <none>                              4962.7 5042.7
## - Months.since.last.delinquent  1   4965.3 5043.3
## - Current.Credit.Balance        1   4969.6 5047.6
## - Number.of.Open.Accounts       1   4969.9 5047.9
## - Purpose                      13   4996.0 5050.0
## - Home.Ownership                3   4984.9 5058.9
## - Maximum.Open.Credit           1   4989.3 5067.3
## - Monthly.Debt                  1   4992.4 5070.4
## - Annual.Income                 1   5052.9 5130.9
## - Term                          1   5105.7 5183.7
## - Current.Loan.Amount           1   5371.0 5449.0
## - Credit.Score                  1   5923.0 6001.0
##
## Step:  AIC=5041.07
## Credit.Default ~ Home.Ownership + Annual.Income + Number.of.Open.Accounts +
##     Years.of.Credit.History + Maximum.Open.Credit + Number.of.Credit.Problems +
##     Months.since.last.delinquent + Bankruptcies + Purpose + Term +
##     Current.Loan.Amount + Current.Credit.Balance + Monthly.Debt +
##     Credit.Score
##
##                               Df Deviance    AIC
## - Bankruptcies                  1   4983.9 5039.9
## - Number.of.Credit.Problems     1   4985.0 5041.0
## <none>                              4983.1 5041.1
## - Years.of.Credit.History       1   4985.3 5041.3
## - Months.since.last.delinquent  1   4985.6 5041.6
## - Number.of.Open.Accounts       1   4989.0 5045.0
## - Current.Credit.Balance        1   4989.8 5045.8
## - Purpose                      13   5014.7 5046.7
## - Home.Ownership                3   5003.8 5055.8
## - Maximum.Open.Credit           1   5008.6 5064.6
## - Monthly.Debt                  1   5013.3 5069.3
## - Annual.Income                 1   5083.2 5139.2
## - Term                          1   5121.3 5177.3
## - Current.Loan.Amount           1   5389.6 5445.6
## - Credit.Score                  1   5941.8 5997.8
##
## Step:  AIC=5039.87
## Credit.Default ~ Home.Ownership + Annual.Income + Number.of.Open.Accounts +
##     Years.of.Credit.History + Maximum.Open.Credit + Number.of.Credit.Problems +
##     Months.since.last.delinquent + Purpose + Term + Current.Loan.Amount +
##     Current.Credit.Balance + Monthly.Debt + Credit.Score
##
##                                 Df Deviance    AIC
## - Number.of.Credit.Problems     1   4985.0 5039.0
## - Years.of.Credit.History       1   4985.9 5039.9
## <none>                              4983.9 5039.9
## - Months.since.last.delinquent  1   4986.6 5040.6
## - Number.of.Open.Accounts       1   4989.7 5043.7
## - Current.Credit.Balance        1   4990.7 5044.7
## - Purpose                      13   5015.9 5045.9
## - Home.Ownership                3   5004.7 5054.7
## - Maximum.Open.Credit           1   5009.2 5063.2
## - Monthly.Debt                  1   5014.3 5068.3
## - Annual.Income                 1   5083.4 5137.4
## - Term                          1   5122.7 5176.7
## - Current.Loan.Amount           1   5390.3 5444.3
## - Credit.Score                  1   5944.0 5998.0
##
## Step:  AIC=5039.02
## Credit.Default ~ Home.Ownership + Annual.Income + Number.of.Open.Accounts +
##     Years.of.Credit.History + Maximum.Open.Credit + Months.since.last.delinquent +
##     Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                                Df Deviance    AIC
## <none>                              4985.0 5039.0
## - Months.since.last.delinquent  1   4987.4 5039.4
## - Years.of.Credit.History       1   4987.4 5039.4
## - Number.of.Open.Accounts       1   4991.0 5043.0
## - Current.Credit.Balance        1   4991.4 5043.4
## - Purpose                      13   5017.0 5045.0
## - Home.Ownership                3   5005.8 5053.8
## - Maximum.Open.Credit           1   5010.5 5062.5
## - Monthly.Debt                  1   5015.6 5067.6
## - Annual.Income                 1   5083.9 5135.9
## - Term                          1   5123.4 5175.4
## - Current.Loan.Amount           1   5392.0 5444.0
## - Credit.Score                  1   5944.1 5996.1
```

```
summary(step.model_back)
```

```
##
## Call:
## glm(formula = Credit.Default ~ Home.Ownership + Annual.Income +
##     Number.of.Open.Accounts + Years.of.Credit.History + Maximum.Open.Credit +
##     Months.since.last.delinquent + Purpose + Term + Current.Loan.Amount +
##     Current.Credit.Balance + Monthly.Debt + Credit.Score, family = binomial,
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5506  -0.7471  -0.5433   0.0363   3.6133
##
## Coefficients:
```

```
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.122e+00  1.264e+00  -0.888 0.374485
## Home.OwnershipHome Mortgage   1.021e+00  1.223e+00   0.835 0.403842
## Home.OwnershipOwn Home        1.085e+00  1.227e+00   0.885 0.376376
## Home.OwnershipRent            1.358e+00  1.221e+00   1.112 0.266160
## Annual.Income                -6.852e-07  7.589e-08  -9.030  < 2e-16 ***
## Number.of.Open.Accounts       2.039e-02  8.320e-03   2.451 0.014250 *
## Years.of.Credit.History       8.076e-03  5.229e-03   1.544 0.122487
## Maximum.Open.Credit          -5.389e-07  1.274e-07  -4.231 2.33e-05 ***
## Months.since.last.delinquent -3.631e-03  2.382e-03  -1.524 0.127448
## Purposebuy a car             -9.620e-01  3.913e-01  -2.459 0.013943 *
## Purposebuy house             -8.288e-01  6.050e-01  -1.370 0.170754
## Purposedebt consolidation    -1.119e+00  2.424e-01  -4.617 3.90e-06 ***
## Purposeeducational expenses  -1.414e+00  1.178e+00  -1.200 0.230291
## Purposehome improvements     -9.503e-01  2.872e-01  -3.309 0.000936 ***
## Purposemajor purchase        -1.040e+00  5.347e-01  -1.945 0.051799 .
## Purposemedical bills         -6.770e-01  4.076e-01  -1.661 0.096692 .
## Purposemoving                -4.662e+00  3.427e+00  -1.360 0.173779
## Purposeother                 -8.940e-01  2.644e-01  -3.381 0.000722 ***
## Purposesmall business         3.140e-02  5.950e-01   0.053 0.957919
## Purposetake a trip           -5.380e-01  6.107e-01  -0.881 0.378312
## Purposevacation              -1.008e+00  1.187e+00  -0.849 0.395842
## Purposewedding               -7.924e-01  8.624e-01  -0.919 0.358204
## TermShort Term               -8.970e-01  7.597e-02 -11.808  < 2e-16 ***
## Current.Loan.Amount          -7.975e-08  3.246e-08  -2.457 0.014002 *
## Current.Credit.Balance        6.363e-07  2.525e-07   2.520 0.011745 *
## Monthly.Debt                  2.572e-05  4.657e-06   5.524 3.32e-08 ***
## Credit.Score                  1.228e-03  1.936e-04   6.345 2.22e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6875.7  on 5930  degrees of freedom
## Residual deviance: 4985.0  on 5904  degrees of freedom
## AIC: 5039
##
## Number of Fisher Scoring iterations: 11
```

8.3 R output for Stepwise Selection Method

```
step.model_both<- step(log.reg,direction = "both")

## Start:  AIC=5046.2
## Credit.Default ~ Id + Home.Ownership + Annual.Income + Years.in.current.job +
##     Tax.Liens + Number.of.Open.Accounts + Years.of.Credit.History +
##     Maximum.Open.Credit + Number.of.Credit.Problems + Months.since.last.delinquent +
##     Bankruptcies + Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                                 Df Deviance    AIC
## - Tax.Liens                      1   4962.3 5044.3
## - Id                             1   4962.5 5044.5
## - Years.of.Credit.History        1   4962.6 5044.6
## - Years.in.current.job          11   4982.7 5044.7
## - Bankruptcies                   1   4963.2 5045.2
## - Number.of.Credit.Problems      1   4963.3 5045.3
## <none>                              4962.2 5046.2
## - Months.since.last.delinquent  1   4964.9 5046.9
## - Current.Credit.Balance         1   4969.2 5051.2
## - Number.of.Open.Accounts        1   4969.4 5051.4
## - Purpose                       13   4995.4 5053.4
## - Home.Ownership                 3   4984.5 5062.5
## - Maximum.Open.Credit            1   4988.8 5070.8
## - Monthly.Debt                   1   4991.7 5073.7
## - Annual.Income                  1   5052.2 5134.2
## - Term                           1   5105.1 5187.1
## - Current.Loan.Amount            1   5370.5 5452.5
## - Credit.Score                  1   5920.4 6002.4
##
## Step:  AIC=5044.34
## Credit.Default ~ Id + Home.Ownership + Annual.Income + Years.in.current.job +
##     Number.of.Open.Accounts + Years.of.Credit.History + Maximum.Open.Credit +
##     Number.of.Credit.Problems + Months.since.last.delinquent +
##     Bankruptcies + Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                                 Df Deviance    AIC
## - Id                            1   4962.7 5042.7
## - Years.in.current.job         11   4982.8 5042.8
## - Years.of.Credit.History       1   4962.8 5042.8
## - Bankruptcies                  1   4963.3 5043.3
## - Number.of.Credit.Problems     1   4964.0 5044.0
## <none>                              4962.3 5044.3
## - Months.since.last.delinquent  1   4965.0 5045.0
## + Tax.Liens                     1   4962.2 5046.2
## - Current.Credit.Balance        1   4969.3 5049.3
## - Number.of.Open.Accounts       1   4969.5 5049.5
## - Purpose                      13   4995.5 5051.5
## - Home.Ownership                3   4984.5 5060.5
## - Maximum.Open.Credit           1   4989.0 5069.0
## - Monthly.Debt                  1   4991.9 5071.9
## - Annual.Income                 1   5052.6 5132.6
## - Term                          1   5105.3 5185.3
## - Current.Loan.Amount           1   5370.5 5450.5
## - Credit.Score                  1   5920.5 6000.5
##
## Step:  AIC=5042.69
## Credit.Default ~ Home.Ownership + Annual.Income + Years.in.current.job +
##     Number.of.Open.Accounts + Years.of.Credit.History + Maximum.Open.Credit +
##     Number.of.Credit.Problems + Months.since.last.delinquent +
##     Bankruptcies + Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                                Df Deviance    AIC
## - Years.in.current.job         11   4983.1 5041.1
## - Years.of.Credit.History       1   4963.1 5041.1
## - Bankruptcies                  1   4963.7 5041.7
## - Number.of.Credit.Problems     1   4964.3 5042.3
## <none>                              4962.7 5042.7
## - Months.since.last.delinquent  1   4965.3 5043.3
## + Id                            1   4962.3 5044.3
## + Tax.Liens                     1   4962.5 5044.5
## - Current.Credit.Balance        1   4969.6 5047.6
## - Number.of.Open.Accounts       1   4969.9 5047.9
## - Purpose                      13   4996.0 5050.0
## - Home.Ownership                3   4984.9 5058.9
## - Maximum.Open.Credit           1   4989.3 5067.3
## - Monthly.Debt                  1   4992.4 5070.4
## - Annual.Income                 1   5052.9 5130.9
## - Term                          1   5105.7 5183.7
## - Current.Loan.Amount           1   5371.0 5449.0
## - Credit.Score                  1   5923.0 6001.0
##
## Step:  AIC=5041.07
## Credit.Default ~ Home.Ownership + Annual.Income + Number.of.Open.Accounts +
##     Years.of.Credit.History + Maximum.Open.Credit + Number.of.Credit.Problems +
##     Months.since.last.delinquent + Bankruptcies + Purpose + Term +
##     Current.Loan.Amount + Current.Credit.Balance + Monthly.Debt +
##     Credit.Score
##
##                                Df Deviance    AIC
## - Bankruptcies                  1   4983.9 5039.9
## - Number.of.Credit.Problems     1   4985.0 5041.0
## <none>                              4983.1 5041.1
## - Years.of.Credit.History       1   4985.3 5041.3
## - Months.since.last.delinquent  1   4985.6 5041.6
## + Years.in.current.job         11   4962.7 5042.7
## + Id                            1   4982.8 5042.8
## + Tax.Liens                     1   4982.9 5042.9
## - Number.of.Open.Accounts       1   4989.0 5045.0
## - Current.Credit.Balance        1   4989.8 5045.8
## - Purpose                      13   5014.7 5046.7
## - Home.Ownership                3   5003.8 5055.8
## - Maximum.Open.Credit           1   5008.6 5064.6
## - Monthly.Debt                  1   5013.3 5069.3
## - Annual.Income                 1   5083.2 5139.2
## - Term                          1   5121.3 5177.3
## - Current.Loan.Amount           1   5389.6 5445.6
## - Credit.Score                  1   5941.8 5997.8
##
## Step:  AIC=5039.87
## Credit.Default ~ Home.Ownership + Annual.Income + Number.of.Open.Accounts +
##     Years.of.Credit.History + Maximum.Open.Credit + Number.of.Credit.Problems +
##     Months.since.last.delinquent + Purpose + Term + Current.Loan.Amount +
##     Current.Credit.Balance + Monthly.Debt + Credit.Score
##
##                                 Df Deviance    AIC
## - Number.of.Credit.Problems     1   4985.0 5039.0
## - Years.of.Credit.History       1   4985.9 5039.9
## <none>                              4983.9 5039.9
## - Months.since.last.delinquent  1   4986.6 5040.6
## + Bankruptcies                  1   4983.1 5041.1
## + Id                            1   4983.6 5041.6
## + Years.in.current.job         11   4963.7 5041.7
## + Tax.Liens                     1   4983.7 5041.7
## - Number.of.Open.Accounts       1   4989.7 5043.7
## - Current.Credit.Balance        1   4990.7 5044.7
## - Purpose                      13   5015.9 5045.9
## - Home.Ownership                3   5004.7 5054.7
## - Maximum.Open.Credit           1   5009.2 5063.2
## - Monthly.Debt                  1   5014.3 5068.3
## - Annual.Income                 1   5083.4 5137.4
## - Term                          1   5122.7 5176.7
## - Current.Loan.Amount           1   5390.3 5444.3
## - Credit.Score                  1   5944.0 5998.0
##
## Step:  AIC=5039.02
## Credit.Default ~ Home.Ownership + Annual.Income + Number.of.Open.Accounts +
##     Years.of.Credit.History + Maximum.Open.Credit + Months.since.last.delinquent +
##     Purpose + Term + Current.Loan.Amount + Current.Credit.Balance +
##     Monthly.Debt + Credit.Score
##
##                                 Df Deviance    AIC
## <none>                              4985.0 5039.0
## - Months.since.last.delinquent  1   4987.4 5039.4
## - Years.of.Credit.History       1   4987.4 5039.4
## + Number.of.Credit.Problems     1   4983.9 5039.9
## + Tax.Liens                     1   4984.2 5040.2
## + Years.in.current.job         11   4964.3 5040.3
## + Id                            1   4984.8 5040.8
## + Bankruptcies                  1   4985.0 5041.0
## - Number.of.Open.Accounts       1   4991.0 5043.0
## - Current.Credit.Balance        1   4991.4 5043.4
## - Purpose                      13   5017.0 5045.0
## - Home.Ownership                3   5005.8 5053.8
## - Maximum.Open.Credit           1   5010.5 5062.5
## - Monthly.Debt                  1   5015.6 5067.6
## - Annual.Income                 1   5083.9 5135.9
## - Term                          1   5123.4 5175.4
## - Current.Loan.Amount           1   5392.0 5444.0
## - Credit.Score                  1   5944.1 5996.1
```

```
summary(step.model_both)
```

```
##
## Call:
## glm(formula = Credit.Default ~ Home.Ownership + Annual.Income +
##     Number.of.Open.Accounts + Years.of.Credit.History + Maximum.Open.Credit +
##     Months.since.last.delinquent + Purpose + Term + Current.Loan.Amount +
##     Current.Credit.Balance + Monthly.Debt + Credit.Score, family = binomial,
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5506  -0.7471  -0.5433   0.0363   3.6133
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.122e+00  1.264e+00  -0.888 0.374485
## Home.OwnershipHome Mortgage   1.021e+00  1.223e+00   0.835 0.403842
## Home.OwnershipOwn Home        1.085e+00  1.227e+00   0.885 0.376376
## Home.OwnershipRent            1.358e+00  1.221e+00   1.112 0.266160
## Annual.Income               -6.852e-07  7.589e-08  -9.030  < 2e-16 ***
## Number.of.Open.Accounts       2.039e-02  8.320e-03   2.451 0.014250 *
## Years.of.Credit.History       8.076e-03  5.229e-03   1.544 0.122487
## Maximum.Open.Credit         -5.389e-07  1.274e-07  -4.231 2.33e-05 ***
## Months.since.last.delinquent -3.631e-03  2.382e-03  -1.524 0.127448
## Purposebuy a car            -9.620e-01  3.913e-01  -2.459 0.013943 *
## Purposebuy house            -8.288e-01  6.050e-01  -1.370 0.170754
## Purposedebt consolidation   -1.119e+00  2.424e-01  -4.617 3.90e-06 ***
## Purposeeducational expenses  -1.414e+00  1.178e+00  -1.200 0.230291
## Purposehome improvements    -9.503e-01  2.872e-01  -3.309 0.000936 ***
## Purposemajor purchase       -1.040e+00  5.347e-01  -1.945 0.051799 .
## Purposemedical bills        -6.770e-01  4.076e-01  -1.661 0.096692 .
## Purposemoving               -4.662e+00  3.427e+00  -1.360 0.173779
## Purposeother                -8.940e-01  2.644e-01  -3.381 0.000722 ***
## Purposesmall business         3.140e-02  5.950e-01   0.053 0.957919
## Purposetake a trip          -5.380e-01  6.107e-01  -0.881 0.378312
## Purposevacation              -1.008e+00  1.187e+00  -0.849 0.395842
## Purposewedding               -7.924e-01  8.624e-01  -0.919 0.358204
## TermShort Term               -8.970e-01  7.597e-02 -11.808  < 2e-16 ***
## Current.Loan.Amount          -7.975e-08  3.246e-08  -2.457 0.014002 *
## Current.Credit.Balance        6.363e-07  2.525e-07   2.520 0.011745 *
## Monthly.Debt                  2.572e-05  4.657e-06   5.524 3.32e-08 ***
## Credit.Score                  1.228e-03  1.936e-04   6.345 2.22e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6875.7  on 5930  degrees of freedom
## Residual deviance: 4985.0  on 5904  degrees of freedom
## AIC: 5039
##
## Number of Fisher Scoring iterations: 11
```

The AIC values corressponding to the models obtained from forward, backward and stepwise selection method are 5046.2,5039 and 5039 respectively. As the AIC value corresponding to backward and stepwise selection methods are minimum we can work with any one of the models. We will work with the regressors obtained by stepwise selection as it is a combination of both forward and backward selection method.

9 Model : Logistic Regression

Now we are about to fit our Logistic Regression model. Our response variable Y is a categorical varable with only 2 categories, i.e. 0 (which indicates that the loan will not default) and 1(which indicates that the loan will default). Logistic regression makes use of the link function $ln\frac{\pi}{1-\pi}$.

The logistic regression model is given by -

$$Y_i \sim Binomial(n_i, \pi_i)$$

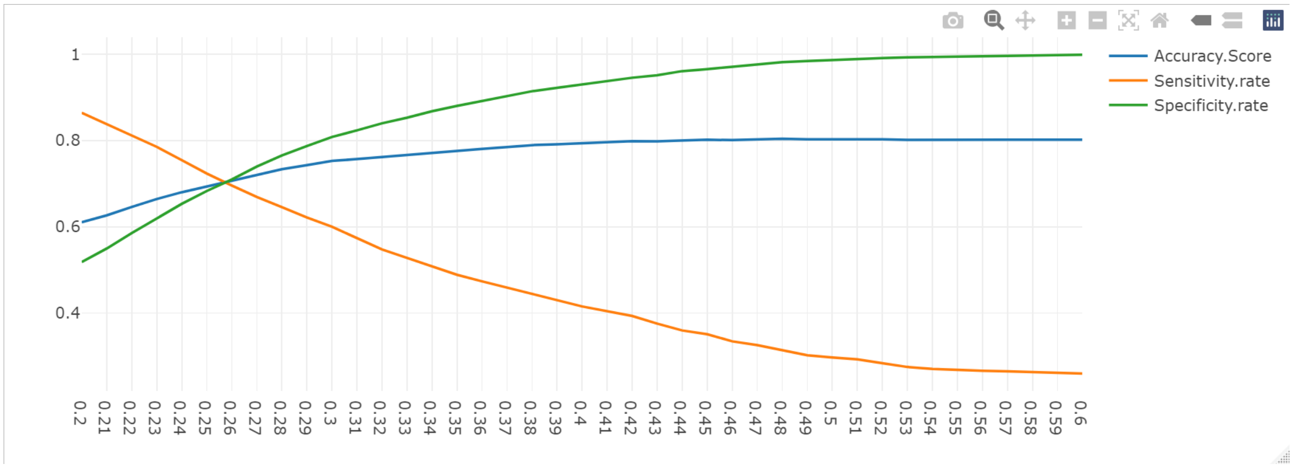
$$ln\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} = \boldsymbol{X}\underset{\sim}{\boldsymbol{\beta}} \quad for \quad i = 1, 2, ..., k$$

where $x_{ij}$is the element in the i^{th}row and $(j+1)^{th}$column of the design matrix X. Here the regressors are qualitative and/or qualitative.

Here the unknown probabilities $\pi_i' s$ are estimated using Maximum Likelihood Method. Then we classify all estimated probabilities as $\widehat{Y}_i = 1, \quad if \quad \widehat{\pi}_i > c \quad and \quad 0 \quad if \quad \widehat{\pi}_i \leq c$ where c is a constant. In our model we will choose the value of c which will yield moderately high values of each of the measures - accuracy score, sensitivity rate and specificity rate (the rates are defined later) .

9.1 Finding the Threshold Value c

The following graph shows the values of accuracy rate, specificity rate and sensitivity rate for different values of c.



From the above plot we choose the value of c to be 0.3, which gives moderately high values of each of the measures - accuracy score, sensitivity rate and specificity rate.

9.2 Model fitting using R

From the variable selection method, we have seen the Stepwise selection method produces the least AIC value. Now we fit the logistic model with that set of regressors and the summary of the model is shown below :

```
summary(step.model_both)

##
## Call:
## glm(formula = Credit.Default ~ Home.Ownership + Annual.Income +
##     Number.of.Open.Accounts + Years.of.Credit.History + Maximum.Open.Credit +
##     Months.since.last.delinquent + Purpose + Term + Current.Loan.Amount +
##     Current.Credit.Balance + Monthly.Debt + Credit.Score, family = binomial,
##     data = train_data)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5506  -0.7471  -0.5433   0.0363   3.6133
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.122e+00  1.264e+00  -0.888 0.374485
## Home.OwnershipHome Mortgage   1.021e+00  1.223e+00   0.835 0.403842
## Home.OwnershipOwn Home        1.085e+00  1.227e+00   0.885 0.376376
## Home.OwnershipRent            1.358e+00  1.221e+00   1.112 0.266160
## Annual.Income                -6.852e-07  7.589e-08  -9.030  < 2e-16 ***
## Number.of.Open.Accounts       2.039e-02  8.320e-03   2.451 0.014250 *
## Years.of.Credit.History       8.076e-03  5.229e-03   1.544 0.122487
## Maximum.Open.Credit          -5.389e-07  1.274e-07  -4.231 2.33e-05 ***
## Months.since.last.delinquent -3.631e-03  2.382e-03  -1.524 0.127448
## Purposebuy a car             -9.620e-01  3.913e-01  -2.459 0.013943 *
## Purposebuy house             -8.288e-01  6.050e-01  -1.370 0.170754
## Purposedebt consolidation    -1.119e+00  2.424e-01  -4.617 3.90e-06 ***
## Purposeeducational expenses  -1.414e+00  1.178e+00  -1.200 0.230291
## Purposehome improvements     -9.503e-01  2.872e-01  -3.309 0.000936 ***
## Purposemajor purchase        -1.040e+00  5.347e-01  -1.945 0.051799 .
## Purposemedical bills         -6.770e-01  4.076e-01  -1.661 0.096692 .
## Purposemoving                -4.662e+00  3.427e+00  -1.360 0.173779
## Purposeother                 -8.940e-01  2.644e-01  -3.381 0.000722 ***
## Purposesmall business         3.140e-02  5.950e-01   0.053 0.957919
## Purposetake a trip           -5.380e-01  6.107e-01  -0.881 0.378312
## Purposevacation              -1.008e+00  1.187e+00  -0.849 0.395842
## Purposewedding               -7.924e-01  8.624e-01  -0.919 0.358204
## TermShort Term               -8.970e-01  7.597e-02 -11.808  < 2e-16 ***
## Current.Loan.Amount          -7.975e-08  3.246e-08  -2.457 0.014002 *
## Current.Credit.Balance        6.363e-07  2.525e-07   2.520 0.011745 *
## Monthly.Debt                  2.572e-05  4.657e-06   5.524 3.32e-08 ***
## Credit.Score                  1.228e-03  1.936e-04   6.345 2.22e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6875.7  on 5930  degrees of freedom
## Residual deviance: 4985.0  on 5904  degrees of freedom
## AIC: 5039
##
## Number of Fisher Scoring iterations: 11
```

From the above table, we have obtained the estimates and std. errors of the coefficients of the regressors. In general, $\widehat{\beta}_j$(estimated coefficient of the jth regressor) is the change in log-odds of $Y = 1$ for a unit change in x_j. In other words $e^{\widehat{\beta}_j}$ is the change in odds ratio of $Y = 1$ for a unit change in x_j. Also the estimates which have less p-value, are more significant, and the estimates having larger p-value are less significant. As for example -

- 1 unit change in **Annual Income** will increase the odds of loan default by $\exp(-6.852\times10^{-7}) = 0.99$, keeping the other regressors fixed, and its p-value indicates that it is significant in determining the loan default.
- A **Short Term** loan is $\exp(-8.970 \times 10^{-01}) = 0.4077912$ times more likely to get default than a Long Term loan, keeping the other regressors fixed, and its p-value also indicates that it is significant in determining the loan default.

Now we fit the train data using the above model :

```
#Predicted_values
step_model_probs<-predict(step.model_both, data=train_data, type = "response")
step_model_predict<-rep(0,nrow(train_data))
step_model_predict[step_model_probs>0.3]=1
step_model_predict

##    [1] 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0
##   [38] 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 1
##   [75] 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0
##  [112] 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1
##  [149] 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0
##  [186] 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
##   [223] 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0
##   [260] 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
##   [297] 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1
##   [334] 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1
##   [371] 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1
##   [408] 0 0 0 0 0 1 0 1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 1
##   [445] 1 0 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 0
##   [482] 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0
##   [519] 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0
##   [556] 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 1
##   [593] 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 1 0
##   [630] 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 1 0 0 1 0
##   [667] 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 0
##   [704] 0 0 1 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0
##   [741] 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0
##   [778] 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 0
##   [815] 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0
##   [852] 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 0 0 0 1 1 0 0 0 0 0 1 0 1 1 1 1 0 1 1 1 0 0
##   [889] 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0
##   [926] 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 1 0 0 0 0 1
##   [963] 0 1 0 0 0 1 0 0 0 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0
##  [1000] 1 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
##  [1037] 0 1 0 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0
##  [1074] 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [1111] 0 1 0 1 1 0 1 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0
##  [1148] 0 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1 0 1 0 0
##  [1185] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0
##  [1222] 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [1259] 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 1 0 0 0
##  [1296] 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0
##  [1333] 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
##  [1370] 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0 1 0
##  [1407] 0 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 0
##  [1444] 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 1 0 0
## [1481] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 0
## [1518] 1 1 1 1 1 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0
## [1555] 1 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0
## [1592] 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 1
## [1629] 0 0 1 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0
## [1666] 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
## [1703] 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0
## [1740] 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1
## [1777] 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 1 0 0 1
## [1814] 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 0 0
## [1851] 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1
## [1888] 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1
## [1925] 0 1 1 0 1 1 0 0 1 1 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1
## [1962] 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0
## [1999] 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
## [2036] 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
## [2073] 0 0 1 0 0 1 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0
## [2110] 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1
## [2147] 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1
## [2184] 1 0 0 0 0 0 1 1 1 1 0 0 0 1 0 1 0 0 1 1 1 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 1
## [2221] 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1
## [2258] 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
## [2295] 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 1
## [2332] 1 1 0 1 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0
## [2369] 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 0 1
## [2406] 1 0 1 1 0 0 0 1 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
## [2443] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 0 1 1 0 0 1 0 1 1 0 0
## [2480] 1 1 0 0 1 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2517] 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0
## [2554] 0 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 1 1 1 0
## [2591] 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0
## [2628] 0 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 1 0 1
## [2665] 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0
## [2702] 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0
## [2739] 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0
## [2776] 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0
## [2813] 0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0
## [2850] 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 0 1 0 1 0 1
## [2887] 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0
## [2924] 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 1 1 0 1 1 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0
## [2961] 1 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 1 0
## [2998] 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0
## [3035] 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0
## [3072] 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
## [3109] 0 0 1 1 0 1 0 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0
## [3146] 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0
## [3183] 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
## [3220] 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 1 0 1 1
## [3257] 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0
## [3294] 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0
## [3331] 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1
## [3368] 1 1 0 1 0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 1
## [3405] 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0
## [3442] 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
## [3479] 1 0 0 0 1 0 1 1 1 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
## [3516] 1 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0
## [3553] 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1
## [3590] 0 0 0 0 1 1 0 1 0 1 1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 0 0
## [3627] 0 1 0 0 1 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3664] 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 0 0 0
## [3701] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0
## [3738] 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0
## [3775] 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 1 0
## [3812] 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0
## [3849] 0 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 1
## [3886] 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0
## [3923] 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 0 1
## [3960] 0 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 0 1
## [3997] 0 1 1 0 1 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 0 1 0
## [4034] 0 1 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 0
## [4071] 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0
## [4108] 0 0 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
## [4145] 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0
## [4182] 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 0 1 0 1 1 1 0 1 1
## [4219] 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0
## [4256] 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0
## [4293] 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0 1 0 1 1 1 0 0 0 1 1 1 0 1 0 0 1 0 0 0
## [4330] 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
## [4367] 1 0 0 1 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0
## [4404] 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 0
## [4441] 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0
## [4478] 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 1
## [4515] 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0
## [4552] 0 0 0 0 1 0 0 1 0 1 0 0 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0
## [4589] 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 0 1 0 0 1 0 0 0 0
## [4626] 0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
## [4663] 1 1 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4700] 0 0 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
## [4737] 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 1
## [4774] 1 0 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0
## [4811] 0 1 0 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 1
## [4848] 0 0 1 1 0 0 1 0 1 0 1 0 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1 0 1 0 0 0
## [4885] 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1
## [4922] 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0
## [4959] 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 0 0 1
## [4996] 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 1 0 0 0 1
## [5033] 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0
## [5070] 1 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0
## [5107] 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 1 0
## [5144] 1 0 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 0 0 1 1 1 0 0 1 0 0 1
## [5181] 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1
## [5218] 1 1 1 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1
## [5255] 1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 0 0
## [5292] 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0
## [5329] 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 0
## [5366] 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 0 1 0 1 0 0 0 0 1 1 0 0 0 0 1
## [5403] 0 1 1 0 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [5440] 0 0 1 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0
## [5477] 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 1
## [5514] 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0
## [5551] 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
## [5588] 0 1 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0
## [5625] 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 1 0 1 1 1
## [5662] 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 1 1 1
## [5699] 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1
## [5736] 0 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1
## [5773] 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0
## [5810] 1 0 0 1 1 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1
## [5847] 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0
## [5884] 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 1 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0
## [5921] 0 0 0 1 0 0 0 1 0 1 0
```

Now, to evaluate the accuracy of the fitted model, we use a confusion matrix.

9.3 Confusion Matrix

Confusion matrix, as the name suggests, describes the performance of the model by showing the ways in which the model is confused when it makes predictions. In a classification problem where the response has two categories, the confusion matrix is of order 2×2 , showing the no. of following four cases :

- **True Positives :** The cases in which we predicted YES and the actual output was also YES.
- **True Negatives :** The cases in which we predicted NO and the actual output was NO.
- **False Positives :** The cases in which we predicted YES and the actual output was also NO.
- **False Negatives :** The cases in which we predicted NO and the actual output was also YES.

The confusion matrix for our fitted model is shown below by the following R output:

```
#Confusion matrix
conf_mat<-table(step_model_predict , train_data$Credit.Default)
conf_mat

##
## step_model_predict    0    1
##                  0 3551  670
##                  1  800  910
```

Confusion Matrix :

	Observed Values	
Predicted Values	0	1
0	$3551(f_{00})$	$670(f_{01})$
1	$800(f_{10})$	$910(f_{11})$

9.4 Some Simple Diagnostics from the Confusion Matrix

9.4.1 Accuracy Score

$$\text{Accuracy Score} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}}$$

The higher the accuracy rate, the better is the model. For our model, the accuracy rate is calculated below :

```
#Accuracy score

acc_score<-sum(diag(conf_mat))/sum(conf_mat)

acc_score

## [1] 0.7521497
```

Hence the accuracy rate for our model is 75.21%.

9.4.2 Sensitivity Rate/True Positive Rate

$$\text{Sensitivity Rate} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

The higher the accuracy rate, the better is the model. For our model, the Sensitivity rate is calculated below :

```
#Sensitivity rate

sen_rate<-conf_mat[2,2]/sum(conf_mat[1,2]+conf_mat[2,2])

sen_rate

## [1] 0.5759494
```

Hence the Sensitivity rate for our model is 57.59 %.

9.4.3 Specificity Rate/True Negative Rate

$$\text{Specificity Rate} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

The higher the Specificity rate, the better is the model. For our model, the Specificity rate is calculated below :

```
#Specificity rate

spec_rate<-conf_mat[1,1]/sum(conf_mat[2,1]+conf_mat[1,1])

spec_rate

## [1] 0.8161342
```

Hence the Specificity rate for our model is 81.61 %.

- With respect to the Accuracy Score and Specificity Rate, the performance of the model is good enough; whereas the Sensitivity Rate is moderate.

9.5 Some More Diagnostics

9.5.1 Deviance Test

Deviance statistic is given by-

$$D = -2ln\frac{L(\beta, \underset{\sim}{x}|y_1, y_2, \ldots\ldots, y_n, fitted\ model)}{L(\beta, \underset{\sim}{x}|y_1, y_2, \ldots\ldots, y_n, saturated\ model)} \ \sim \ \chi^2_{n-p-1}$$

where n = total no. of observations and p = total no. of regressors.

Small value of observed D implies fitted model is close to the saturated model and large value of observed D implies fitted model is further away from the saturated model.

If $D_{obs} < \ \chi^2_{1-\alpha,\ n-p-1}$ (α being the level of significance), we conclude the fitted model explains the data as efficiently as the saturated model; otherwise we conclude that the fitted model is not close to the saturated model.

Here $n = 5931, p = 26$.

Now we perform the Deviance test using R :

```
#Deviance test
step.model_both$deviance<qchisq(0.95,nrow(train_data)-26-1)

## [1] TRUE
```

From the above R output, we conclude that the fitted model is close to the saturated model at 5% level of significance.

9.5.2 Pearson's χ^2 Test

Now we are going to test the following hypothesis : $H_0 : Y\ and\ \widehat{Y}\ are\ independent$ against $H_1 :\ Y\ and\ \widehat{Y}\ are\ not\ independent$

Here the test statistic is given by -

$$\chi^2_p = \frac{n(f_{11}f_{00} - f_{10}f_{01})^2}{f_{1.}f_{0.}f_{.0}f_{.1}} \ \sim \ \chi^2_1$$

where $f_{i.} = f_{i0} + f_{i1}; i = 0, 1$; $f_{.j} = f_{0j} + f_{1j}; j = 0, 1$.

If observed $\chi^2_p > \chi^2_{1-\alpha,1}$ (α being the level of significance), we reject H_0 otherwise we accept H_0.

Now we perform the test using R :

```
#Goodness of fit
chisq_test<-chisq.test(train_data$Credit.Default,step_model_predict)
chisq_Val<-chisq_test$statistic
chisq_Val>qchisq(0.95,1)
```

```
## X-squared
##      TRUE
```

From the above R output, we conclude that H_0 is rejected i.e. Y *and* $\widehat{Y}$ *are not independent*.

As H_1 is accepted, we will calculate the following measures of association between Y *and* $\widehat{Y}$.

9.5.3 Phi Coefficient

The Phi coefficient is given by -

$$\phi = \frac{f_{11}f_{00} - f_{10}f_{01}}{\sqrt{f_{1.}f_{0.}f_{.0}f_{.1}}}$$

We know that $\phi \in [-1, 1]$ and the higher value of ϕ indicates stronger association between Y *and* $\widehat{Y}$.

Now we calculate the Phi Coefficient using R :

```
#Phi_coefficient
library(psych)
phi(conf_mat, digits = 3)

## [1] 0.383
```

From the observed value of ϕ, we conclude that there is significant association between Y *and* $\widehat{Y}$.

9.5.4 Contingency Coefficient

The Contingency coefficient is given by -

$$P = \sqrt{\frac{\chi_p^2}{\chi_p^2 + n}}$$

We know that $P \in (0, 1)$ and the higher value of P indicates stronger association between Y *and* $\widehat{Y}$.

Now we calculate the Contingency Coefficient using R :

```
#Contingency coefficient
con_coeff<-sqrt(chisq_Val/(nrow(train_data)+chisq_Val))
con_coeff

## X-squared
## 0.3570324
```

The observed value of P indicates that the association between Y *and* $\widehat{Y}$ is moderate.

10 Prediction for Test Data

Now we will move for predicting the test data using the fitted logistic model. At first we will fetch the data and check whether there are any missing value. After cleaning tha data, we will predict whether a loan will default or not, for each of the observations in test data.

So at first we show a **summary of the data** using R :

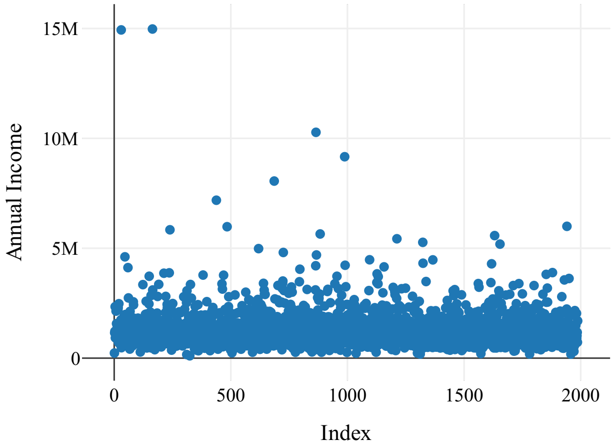
```
test_data=read.csv("test.csv")
View(test_data)
test_data[sapply(test_data, is.character)] <- lapply(test_data[sapply(test_data, is.character)],as.factor)
str(test_data)

## 'data.frame': 2500 obs. of  17 variables:
##  $ Id                          : int  7500 7501 7502 7503 7504 7505 7506 7507 7508 7509 ...
##  $ Home.Ownership              : Factor w/ 4 levels "Have Mortgage",..: 4 4 2 2 2 2 2 4 4 2 ...
##  $ Annual.Income               : num  NA 231838 1152540 1220313 2340952 ...
##  $ Years.in.current.job        : Factor w/ 12 levels "","< 1 year",..: 7 3 6 4 9 8 6 6 8 4 ...
##  $ Tax.Liens                   : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Number.of.Open.Accounts     : num  9 6 10 16 11 26 7 13 8 15 ...
##  $ Years.of.Credit.History     : num  12.5 32.7 13.7 17 23.6 17.5 22 12.2 9.1 16.7 ...
##  $ Maximum.Open.Credit         : num  220968 55946 204600 456302 1207272 ...
##  $ Number.of.Credit.Problems   : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Months.since.last.delinquent: num  70 8 NA 70 NA 41 43 19 NA 9 ...
##  $ Bankruptcies                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Purpose                     : Factor w/ 14 levels "business loan",..: 4 5 4 4 4 4 6 4 7 4 ...
##  $ Term                        : Factor w/ 2 levels "Long Term","Short Term": 2 2 2 2 1 2 2 2 2 2 ...
##  $ Current.Loan.Amount         : num  162470 78298 200178 217382 777634 ...
##  $ Current.Credit.Balance      : num  105906 46037 146490 213199 425391 ...
##  $ Monthly.Debt                : num  6813 2318 18729 27559 42605 ...
##  $ Credit.Score                : num  NA 699 7260 739 706 679 685 701 NA 745 ...

dim(test_data)

## [1] 2500   17
```

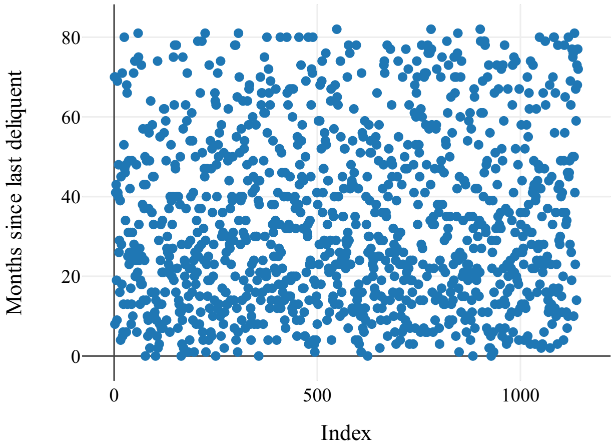
Now, the no. of missing values in different variables are shown in the following R output :

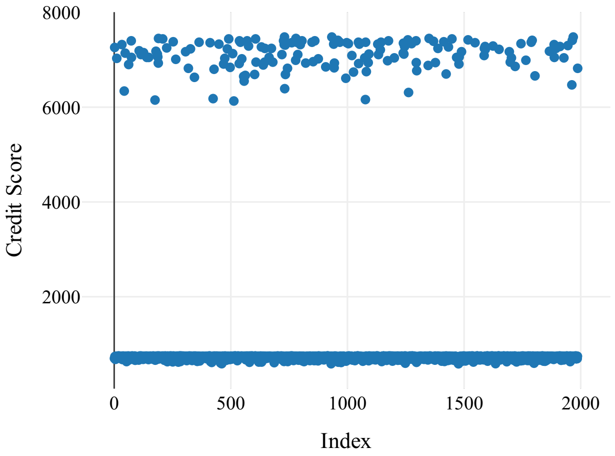
```
colSums(is.na(test_data))

##                           Id               Home.Ownership
##                            0                            0
##                Annual.Income         Years.in.current.job
##                          513                            0
##                    Tax.Liens      Number.of.Open.Accounts
##                            0                            0
##      Years.of.Credit.History          Maximum.Open.Credit
##                            0                            0
##    Number.of.Credit.Problems Months.since.last.delinquent
##                            0                         1358
##                 Bankruptcies                      Purpose
##                            3                            0
##                         Term          Current.Loan.Amount
##                            0                            0
##       Current.Credit.Balance                 Monthly.Debt
##                            0                            0
##                 Credit.Score
##                          513
```

From the above table, we can see that the no. of missing values in Annual Income, Months since last delinquent, Bankruptcies and Credit Score are 513, 1358, 3 and 513 respectively.

Here the no. of missing values corresponding to the variable Bankruptcies is 3, which is too small compared to the whole dataset of 2500 observations. So we will discard the corresponding rows with missing values for this variable, and plot the other 3 variables with missing values. -


15M
10M
5M
0
Annual Income
0
500
1000
1500
2000
Index


80
60
40
20
0
Months since last deliquent
0
500
1000
Index



From the plot of the variable Months since last delinquent, it is evident that mean imputation will be appropiate for this variable.

From the plot of the Annual Income, we can see that the observations are very dense for the lower income group whereas there are few observations for the higher income group. So, here we have to apply the median imputation.

From the plotted diagram of the variable Credit Score, it is clear that the values of the variable are highly bipolarised; so we can't apply any kind of imputation technique for the missing values of Credit Score. So, the corresponding rows with these missing values will also be discarded.

The above three missing value operations are done by the following R codes :

```
test_data$Months.since.last.delinquent[is.na(test_data$Months.since.last.delinquent)]=
+ median(test_data$Months.since.last.delinquent,na.rm=T)
test_data$Annual.Income[is.na(test_data$Annual.Income)]=median(test_data$Annual.Income,na.rm=T)
test_data=na.omit(test_data)
```

Now, let us check whether we are left with any missing value in our data or not, using the following R code :

```
colSums((is.na(test_data)))

##                            Id              Home.Ownership
##                             0                           0
##                 Annual.Income          Years.in.current.job
##                            0                            0
##                    Tax.Liens      Number.of.Open.Accounts
##                            0                            0
##      Years.of.Credit.History          Maximum.Open.Credit
##                            0                            0
##    Number.of.Credit.Problems Months.since.last.delinquent
##                            0                            0
##                 Bankruptcies                      Purpose
##                            0                            0
##                         Term          Current.Loan.Amount
##                            0                            0
##       Current.Credit.Balance                 Monthly.Debt
##                            0                            0
##                 Credit.Score
##                            0
```

So, now our data has been cleaned and we can proceed our further prediction with this dataset.

Now, we will predict whether the loan will default for the observations in the test data.

```
#Predicted values for test_data
credit_default_probs<-step.model_both %>% predict(newdata=test_data, type = "response")
test_data$Credit.Default<-rep(0,nrow(test_data))
test_data$Credit.Default[credit_default_probs>0.3]=1
test_data$Credit.Default
```

```
##    [1] 0 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0
##   [38] 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 0 1 1
##   [75] 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 0
##  [112] 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1
##  [149] 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 0 1
##  [186] 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0
##  [223] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0
##  [260] 1 0 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0
##  [297] 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0
##  [334] 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0
##  [371] 0 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 1 0
##  [408] 0 0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0
##  [445] 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0
##  [482] 0 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0 0
##  [519] 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0
##  [556] 1 0 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 0 0 1 0 1
##  [593] 0 0 0 0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [630] 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
##  [667] 0 0 1 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0
##  [704] 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 1 1 1 0 0 0 1 0 0 0
##  [741] 1 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 1 1
##  [778] 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 1 1 0
##  [815] 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 0 0 0 1
##  [852] 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0
##  [889] 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
##  [926] 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 1 1 0 0 0 1 0 0 1 0 0
##  [963] 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 0 1
## [1000] 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0
## [1037] 0 1 1 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
## [1074] 0 1 1 1 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0
## [1111] 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0
## [1148] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0
## [1185] 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 1
## [1222] 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0
## [1259] 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1
## [1296] 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 0 1 0 1
## [1333] 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0 1 0 1
## [1370] 1 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0
## [1407] 0 0 0 0 1 0 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 0 0
## [1444] 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 1 0 0
## [1481] 0 0 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1
## [1518] 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0
## [1555] 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 1 0 0
## [1592] 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0
## [1629] 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1
## [1666] 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 1 1 0 1 0 0 1 0
## [1703] 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0
## [1740] 0 1 1 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0
## [1777] 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 1
## [1814] 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0
## [1851] 0 0 0 1 0 0 1 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0
## [1888] 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0
## [1925] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0
## [1962] 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1
```

10.1 Writing our output to a new file

Let us save our predicted values with Customer Id's to a new csv file Sample.csv

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[2] D.W. Hosmer, S. Lemeshow, and R.X. Sturdivant. *Applied Logistic Regression*. Wiley Series in Probability and Statistics. Wiley, 2013.

[3] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

[4] Douglas C. Montgomery, Elizabeth A. Peck, and Geoffrey G. Vining. *Introduction to Linear Regression Analysis (4th ed.)*. Wiley & Sons, 2006.