# LOAN DATA SET ANALYSIS

# 1) INTRODUCTION:

Two data sets are available with first “Data for Cleaning & Modeling.csv” for model creating and second “Holdout for Testing.csv” for regression. About 400000 data points are available for creating a regression model which will predict interest on loan for data in second file.

## 1.1) OBJECTIVE:

To predict interest on loan based on data given.

## 1.2) INDEPENDENT VARIABLES:

| | |
|---|---|
| X2 | A unique id for the loan. |
| X3 | A unique id assigned for the borrower. |
| X4 | Loan amount requested |
| X5 | Loan amount funded |
| X6 | Investor-funded portion of loan |
| X7 | Number of payments (36 or 60) |
| X8 | Loan grade |
| X9 | Loan subgrade |
| X10 | Employer or job title (self-filled) |
| X11 | Number of years employed (0 to 10; 10 = 10 or more) |
| X12 | Home ownership status: RENT, OWN, MORTGAGE, OTHER. |
| X13 | Annual income of borrower |
| X14 | Income verified, not verified, or income source was verified |
| X15 | Date loan was issued |
| X16 | Reason for loan provided by borrower |
| X17 | Loan category, as provided by borrower |
| X18 | Loan title, as provided by borrower |
| X19 | First 3 numbers of zip code |
| X20 | State of borrower |
| X21 | A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested loan, divided by the borrower’s self-reported monthly income. |
| X22 | The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years |
| X23 | Date the borrower's earliest reported credit line was opened |
| X24 | Number of inquiries by creditors during the past 6 months. |
| X25 | Number of months since the borrower's last delinquency. |
| X26 | Number of months since the last public record. |
| X27 | Number of open credit lines in the borrower's credit file. |
| X28 | Number of derogatory public records |
| X29 | Total credit revolving balance |
| X30 | Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit. |
| X31 | The total number of credit lines currently in the borrower's credit file |
| X32 | The initial listing status of the loan. Possible values are – W, F |

## 1.3) DEPENDANT VARIABLES:

X1 Interest Rate on the loan.

## 1.4) APPROACHES TAKEN FOR ANALYSING DATA

For analysis of given data R-Programming is used which is a programming language used mainly for the purpose of analysis.

The following analytical approaches are taken:

- **Linear regression**
  - **Lm**
- **Regression Trees**
  - **Rpart**

## 2) EXPLORATORY DATA ANALYSIS

First process of any analysis is to understand the data. Which is done using two functions

1) str()
2) summary()

"str" gives CLASS of every variable in data-set with few examples as shown below,

```
str(dat)
'data.frame':  400000 obs. of  32 variables:
 $ X1 : chr  "11.89%" "10.71%" "16.99%" "13.11%" ...
 $ X2 : int  54734 55742 57167 57245 57416 58524 58915 59006 61390 61419 ...
 $ X3 : int  80364 114426 137225 138150 139635 149512 153417 154254 182594 182917 ...
```

"summary" function gives descriptive statistics of data as shown below

```
summary(dat)
      X1                  X2                  X3                 X4
 Length:400000      Min.   :   54734   Min.   :   70699   Length:400000
 Class :character   1st Qu.: 3151742   1st Qu.: 3727712   Class :character
 Mode  :character   Median : 8234778   Median : 9667699   Mode  :character
                    Mean   : 9984493   Mean   :11338986
                    3rd Qu.:15329598   3rd Qu.:17312192
                    Max.   :28753146   Max.   :31278050
                    NA's   :1          NA's   :1
```

## 2.1) MISSING DATA ANALYSIS:

Using following command, we find out total no. of missing values present in our training data set.

sum(is.na(dat))

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 | X15 | X16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| no. of missing values | 61010 | 1 | 1 | 1 | 1 | 1 | 1 | 61270 | 61270 | 23970 | 1 | 61361 | 61028 | 1 | 1 | 27658 |
| | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X25 | X26 | X27 | X28 | X29 | X30 | X31 | X32 |
| | 1 | 18 | 1 | 1 | 1 | 1 | 1 | 1 | 218802 | 348845 | 1 | 1 | 1 | 267 | 1 | 1 |

We can see from above table that variable "X16", "X25" and "X26" contains more than 50% of missing data

Also form str function along with our studies on the subject we remove variables which are not important for analysis, to reduce dimension and increase processing speed of our model.

After this our dimension of data set reduced from "800000 X 32" to "242983 X 22"

## 2.2) FORMATING DATA

Converting data into correct format for calculations and calculating new variables for analysis is an important part of pre-processing, also known as "Feature engineering". From str function we found out that many numerical variables which in percentage, date format are read into character class. Also variables which are factors have long names. Data is processed in r with bites. And every character takes one bite to store. More characters mean more processing power and storing capacity. So converting factors into numeric variable or single word format reduces our memory usage.

"X1" & "X30" are converted into numeric factors from character variables.

"X4", "X5" & "X6" are converted into numeric variables after removing "$" & "," sign.

"X7", "X8", "X11", "X12", "X14", "X17" & "X32" are converted to factors with factor names replaced to numeric characters.

"X15" & "X23" are converted to date format with adding 01 as date.

"X34" variable is created by extracting year from "X23".

"X33" variable is created by finding a fraction of loan funded by investor.

## 2.3) REMOVING OUTLIERS

Outliers are variables which lies far away from mean. They can be calculated as,

Outlier range <- Q1 – N*(Q3– Q1) To Q3 + N * (Q3-Q1)

There are two types of outliers, minor outliers and major outliers. To calculate **minor outliers** put **N = 1.5**, to calculate **major outliers** put **N = 3**. We have removed only major outliers from data. Outliers are removed because they have very high chance of being wrong data, but not all outliers are wrong. Hence we only remove outliers which represent wrong data.

Our dimension of data reduced to "232252 X 22".

## 2.4) NORMALIZING DATA

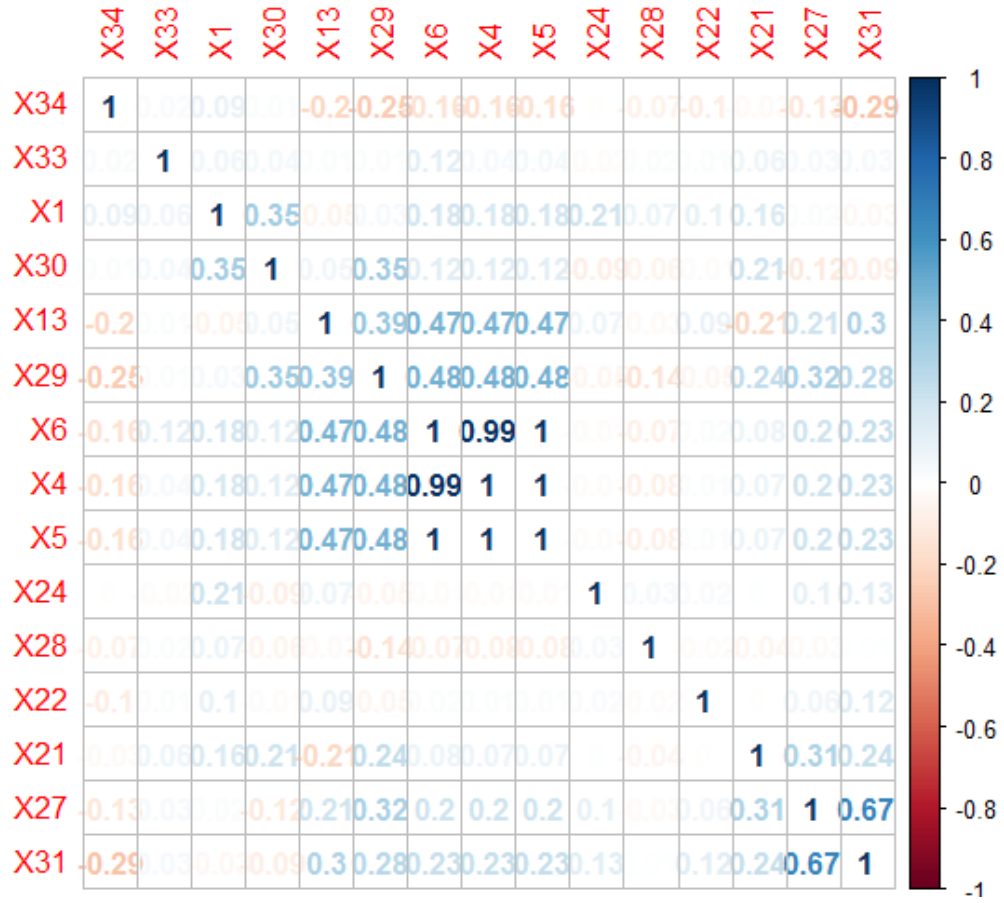
Numeric variables in our data-set have huge difference in their ranges, for eg. X4 ranges from 500 to 35000 where as X21 ranges from 0 to 39.99, because of this our model will be more bent towards X21 than X4, to remove this we normalize ranges, compresses all variable ranges to "0 to 1" by min-max method.

Formula for this is:

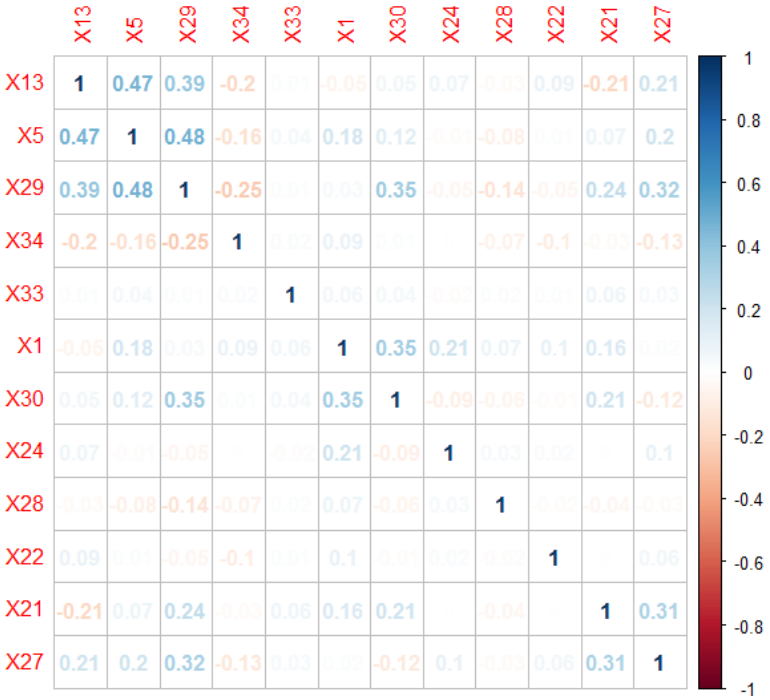
X <- (X -min(X))/(max(X)-min(X))

## 2.5) CORRELATED DATA

Data which is highly correlated provides same information to the model while inflating data accuracy. Hence it is very important to remove highly correlated data. For this we use "corrplot" library containing function called "corrplot" which plots correlation of different variables in a matrix. Its desirable to have very low correlation between two variables. Following image shows our first correlation plot.




We can see that variables “X4”,”X5”, “X6” are highly correlated with each other, also “X27” & “X31” are correlated to each other. So we remove variables “X4”, “X6” and “X31”, reducing dimension of data to “232252 X 20”. Corrplot for final variables is shown below



## 3) BUILDING DATA MODEL

First thing of building any model is to divide our data set into two parts, one for training model, on which regression model is built and second for testing purpose, to check accuracy of our model on unknown data. Our model is divided in to Train with dimensions "170000 X 20" and Test1 with dimension "62252 X 20"

## 3.1) LINEAR REGRESSION

R provides a build-in function "lm" which creates a linear regression model.

our first model gives us following result:

```
summary(model1)

Call:
lm(formula = X1 ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.72672 -0.04020  0.00232  0.04126  0.21206
```

This is first part of our model, which shows deviation of our prediction from actual value. Above result shows min. deviation of our prediction is -0.726 and max. deviation is 0.212

Following table shows importance of each variable in creating our model. Variables with 3 stars shows most important variables. With probability of null hypothesis being zero and no star showing probability of null hypothesis being 1. Where null hypothesis being that the variable has no effect on target variable.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.509e-01  6.003e-03 -41.801  < 2e-16 ***
X5           1.516e-02  8.478e-04  17.885  < 2e-16 ***
X71          8.667e-03  4.040e-04  21.454  < 2e-16 ***
X82          1.832e-01  4.589e-04 399.119  < 2e-16 ***
X83          3.286e-01  5.101e-04 644.167  < 2e-16 ***
X84          4.650e-01  5.856e-04 794.035  < 2e-16 ***
X85          6.015e-01  7.403e-04 812.556  < 2e-16 ***
X86          7.392e-01  1.015e-03 728.047  < 2e-16 ***
X87          8.096e-01  1.827e-03 443.045  < 2e-16 ***
X111         2.392e-03  7.554e-04   3.167 0.001539 **
X112         1.945e-03  6.939e-04   2.803 0.005071 **
X113         1.656e-03  7.145e-04   2.317 0.020481 *
X114         1.286e-04  7.691e-04   0.167 0.867174
X115         4.684e-03  7.419e-04   6.313 2.75e-10 ***
X116         5.547e-03  7.721e-04   7.185 6.74e-13 ***
X117         3.647e-03  7.788e-04   4.683 2.83e-06 ***
X118         3.904e-03  8.227e-04   4.746 2.08e-06 ***
X119         4.895e-03  8.900e-04   5.500 3.80e-08 ***
X1110        4.871e-03  5.736e-04   8.493  < 2e-16 ***
X121         2.141e-02  1.238e-02   1.729 0.083779 .
X122         1.680e-03  6.348e-04   2.646 0.008150 **
X123         2.116e-03  4.357e-04   4.857 1.19e-06 ***
X124        -3.000e-03  4.232e-04  -7.090 1.35e-12 ***
X13         -2.676e-02  1.176e-03 -22.758  < 2e-16 ***
X141        -1.492e-02  3.546e-04 -42.087  < 2e-16 ***
X142        -1.486e-02  3.775e-04 -39.351  < 2e-16 ***
X15          1.280e-05  4.146e-07  30.868  < 2e-16 ***
X171         7.499e-03  1.556e-03   4.820 1.43e-06 ***
X172         1.681e-03  1.100e-03   1.528 0.126410
X173         7.247e-03  1.184e-03   6.122 9.25e-10 ***
X174         1.144e-02  1.196e-03   9.564  < 2e-16 ***
X21          3.284e-03  8.808e-04   3.729 0.000192 ***
```

```
X22          1.189e-02  9.489e-04  12.534  < 2e-16 ***
X24          2.567e-02  6.080e-04  42.223  < 2e-16 ***
X27          1.477e-02  1.108e-03  13.335  < 2e-16 ***
X28          7.656e-02  1.837e-02   4.168 3.07e-05 ***
X29         -1.743e-02  1.123e-03 -15.522  < 2e-16 ***
X30          4.848e-02  7.599e-04  63.795  < 2e-16 ***
X32w        -9.262e-03  3.277e-04 -28.261  < 2e-16 ***
X34          1.910e-02  1.462e-03  13.066  < 2e-16 ***
X33          1.263e-01  2.536e-03  49.783  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Following is third and last part of model summary which shows value of R – squared and p-value.

```
Residual standard error: 0.05805 on 169959 degrees of freedom
Multiple R-squared:  0.9248,   Adjusted R-squared:  0.9247
F-statistic: 5.223e+04 on 40 and 169959 DF,  p-value: < 2.2e-16
```

Adjusted R-squared is accuracy of our model, which is showing 92.47% and p-value is the probability of null hypothesis that variables have no effect on target variable which is 0.

Finally, our model is tested against test data using predict function. To get accuracy on testing data we use cor function and Mean Absolute Error which comes out to be,

```
 cor(pred1,test1$X1)
[1] 0.9608018
MAE(pred1,test1$X1)
[1] 0.04695048
```

Which means correlation of our predicted data and actual value is 96% and mean absolute error is 0.047

## 3.2) REGRESSION TREE(Rpart)

For our next model we use Rpart library. Which produces regression trees with each branch showing a rule of regression. Our model gives following output

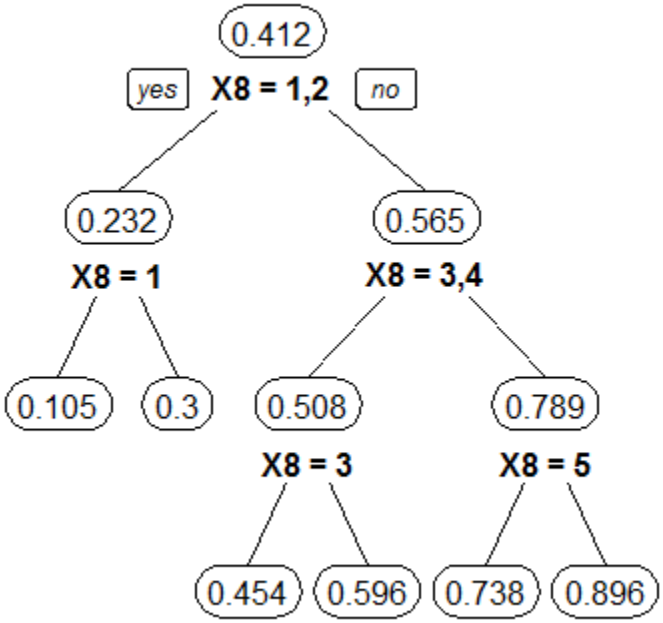
```
model2
n= 170000

node), split, n, deviance, yval
      * denotes terminal node

 1) root 170000 7612.31700 0.4117093
   2) X8=1,2 78448  935.22740 0.2324374
     4) X8=1 27089   62.73443 0.1045812 *
     5) X8=2 51359  196.09570 0.2998744 *
   3) X8=3,4,5,6,7 91552 1995.56400 0.5653217
     6) X8=3,4 72874  607.57690 0.5079244
      12) X8=3 45211  144.03360 0.4541887 *
      13) X8=4 27663  119.63470 0.5957473 *
     7) X8=5,6,7 18678  211.21500 0.7892626
      14) X8=5 12652   72.64392 0.7384302 *
      15) X8=6,7 6026   37.24012 0.8959887 *
```

For each node in the tree, the number of examples reaching the decision point is listed. For instance, all 170000 examples begin at the root node, of which 78448 have X8 = 1 .85 and 51359 have X8 = 2. Because X8 was used first in the tree, it is the single most important predictor of interest.

Nodes indicated by

* are terminal or leaf nodes, which means that they result in a prediction (listed here as yval). For example, node 5 has a yval of 0.29987.

We can also visualise the rules using rpart.plot library which gives us following result



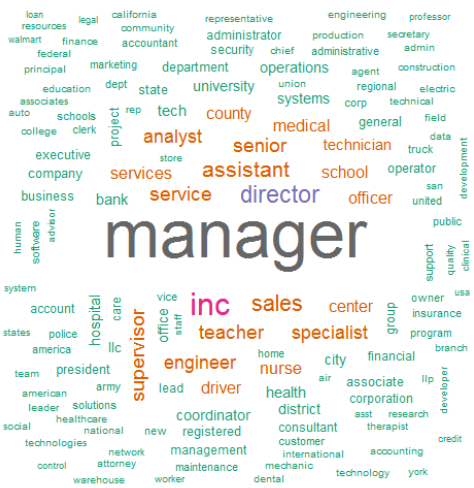

To get accuracy of our model again we use cor and MAE functions

```
> cor(pred2,test1$X1)
[1] 0.9568445
> MAE(pred2,test1$X1)
[1] 0.04916234
```

We are getting 95.7% accuracy.

## 4) ANALYSIS OF X10 AND X16

Finally, we analysis peoples who taking more loans and reason behind most loans. After doing text analysis on variables X10/ Employer job title and X16/Reason for loan we get following output.




from this wordcloud we can see that most people who applied for loan are managers, directors, teachers etc. it is beneficial to know people from which field and posts are getting loans so that we can focus on them and try and get focus on people from other categories as well.

this image shows reasons behind taking loan. We can see that most of people who took loan did it to return borrowed money or pay their debt and loan and bills.