Project Description

Dream Housing Finance company deals in all home loans. A customer first applies for home loan and then the company validates the customer eligibility for loan. The company wants to automate the loan eligibility process based on customer detail provided while filling an online application form. These details are gender, marital status, education, number of dependents, income, loan amount, credit history and others.

The goal of this project is to predict whether the company should approve the loan based on the applicant profile or not. This means that we have a binary classification problem. For more information see Loan Prediction presented by Analytics Vidhya.

Data

As with most Analytics Vidhya competitions, the Loan prediction data consists of a training set and a test set, both of which are .csv files:

- The training set contains data for a subset of applicants including the outcomes or "ground truth" (i.e. the "Loan_Status" response variable). We see that the training set has 614 observations (rows) and 13 variables. This set will be used to build the machine learning models.

- The test set contains data for a subset of applicants without providing the ground truth, since the project's objective is to predict these outcomes. The test set has 367 observations and 12 variables, since the "Loan_Status" variable is missing. This set will be used to see how well the developed model performs on unseen data.

A description of the variables that are encountered in the loan prediction dataset is given in Table 1.

Variable Name	Variable Description	Possible Values	Categorical/Numerical
Loan_ID	Unique Loan ID	1, 2, …, 981	Categorical
Gender	Gender of applicant	Female, Male	Categorical
Married	Marital Status	No, Yes	Categorical
Dependents	No. of dependents	0, 1, 2, 3+	Categorical
Education	Education level	Graduate, Not Graduate	Categorical
Self_Employed	Self employment status	No, Yes	Categorical
ApplicantIncome	Applicant's income	0 – 81000	Numerical
CoapplicantIncome	Co-applicant's income	0 – 41667	Numerical
LoanAmount	Loan amount in thousands	9.0 – 700.0	Numerical
Loan_Amount_Term	Term of loan in months	6.0 – 480.0	Numerical
Credit_History	Credit history meets guidelines	0, 1	Numerical
Property_Area	Area of property	Rural, Semiurban, Urban	Categorical
Loan_Status	Status of the loan	N= No, Y= Yes	Categorical

Table 1: Data description.

Descriptive Statistics and Exploratory Analysis

This section is a very important part in the analysis. By plotting exploratory graphs, we gain a better idea of the data. This section will be divided into two parts, univariate visualization and variables against response.

Before proceeding with the visualizations, we are going to calculate the number and percentage of applicants whose loan was approved from the training set.

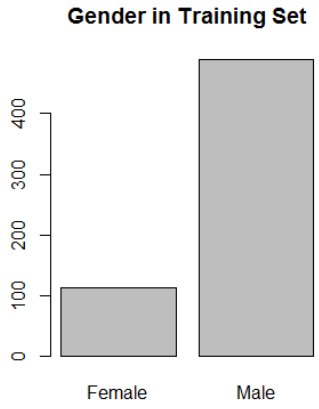
	Amount	Percentage
N	192	0.3127036
Y	422	0.6872964

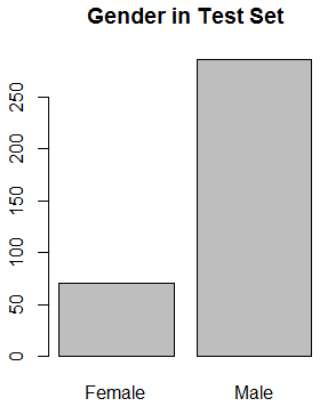
We see that only 192 applicants didn't take the loan, while 422 did, which means that the rate of approval is ~68%.

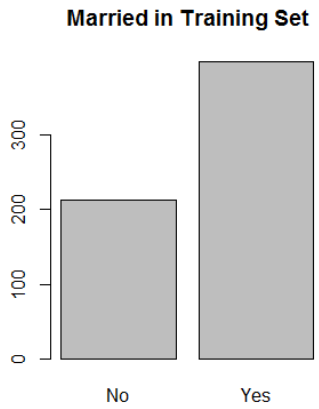
Univariate visualization

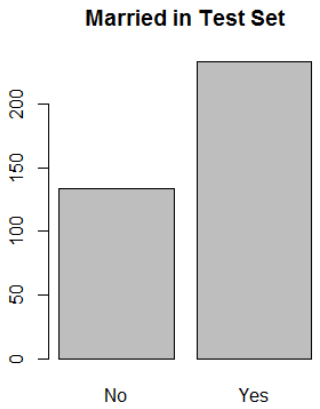
The next figures show the exploratory graphs of the data in the training and test sets. Some interesting findings from these plots are:

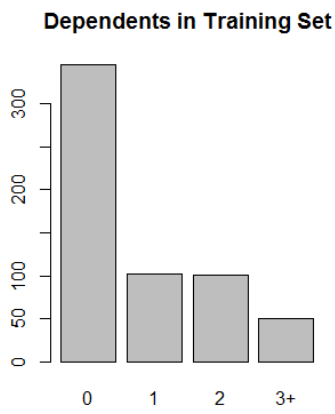
- Most applicants are males in both the training and test sets.
- Most applicants are married in both the training and test sets.
- Most applicants do not have dependents in both the training and test sets.
- Most applicants are graduates in both the training and test sets.
- The vast majority of applicants are not self employed in both the training and test sets.
- The applicant income, co-applicant income and the loan amount have many outliers and the distributions are right asymmetric.
- Most applicants have a credit history that meets guidelines in both the training and test sets.
- Property area is the only predictor variable whose distribution looks different between the training and test sets.

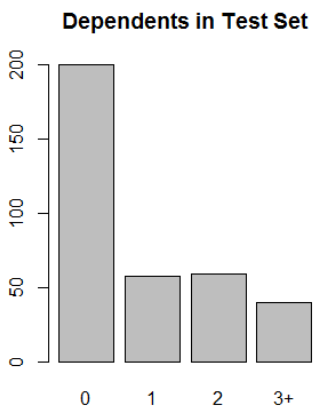

Gender in Training Set
0
100
200
300
400
Female
Male

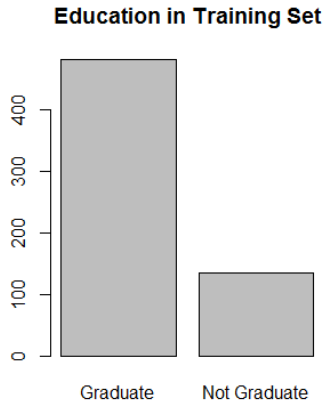

Gender in Test Set
0
50
100
150
200
250
Female
Male

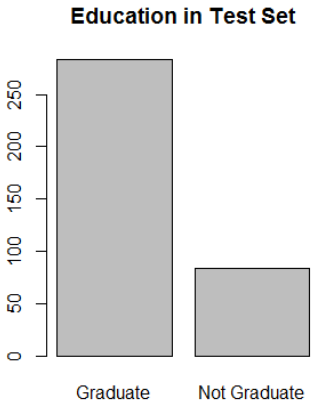

Married in Training Set
0
100
200
300
No
Yes

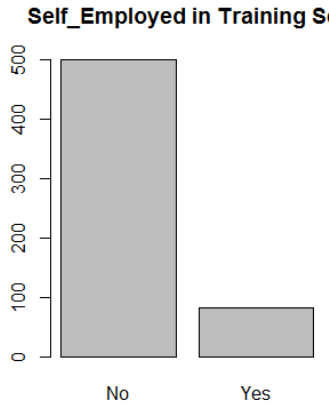

Married in Test Set
0
50
100
150
200
No
Yes

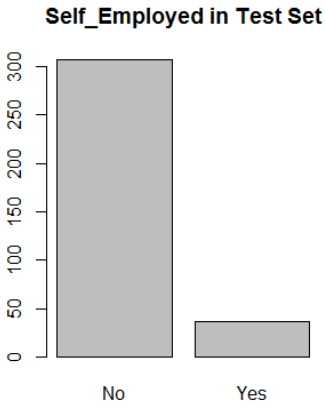

Dependents in Training Set
0
50
100
200
300
0
1
2
3+

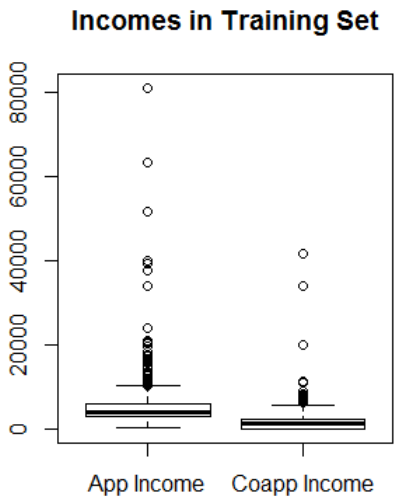

Dependents in Test Set
0
50
100
150
200
0
1
2
3+

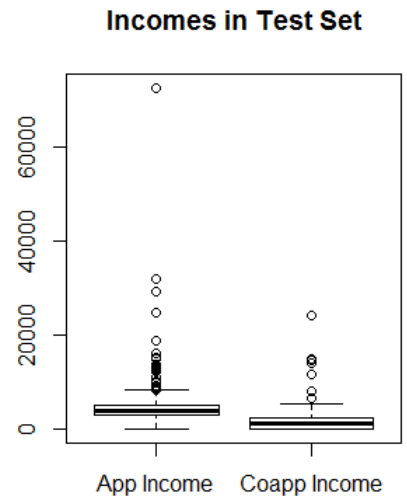

Education in Training Set
0
100
200
300
400
Graduate
Not Graduate

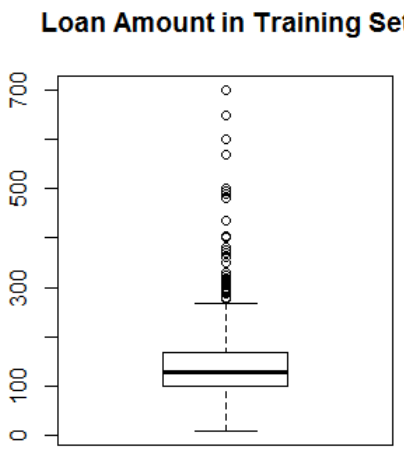

Education in Test Set
0
50
100
150
200
250
Graduate
Not Graduate

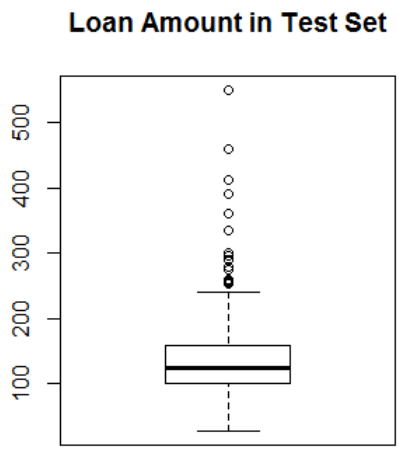

Self_Employed in Training Set
0
100
200
300
400
500
No
Yes

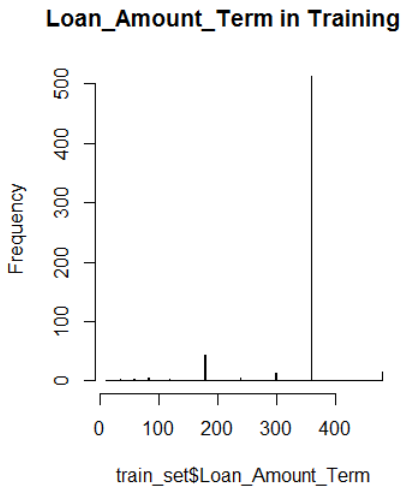

Self_Employed in Test Set
0
50
100
150
200
250
300
No
Yes

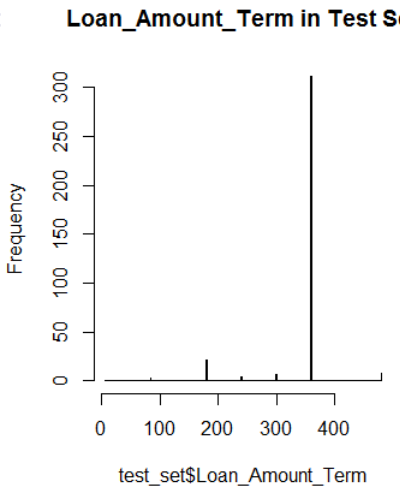

Incomes in Training Set
0
20000
40000
60000
80000
App Income
Coapp Income

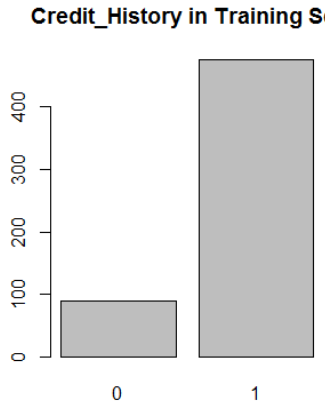

Incomes in Test Set
0
20000
40000
60000
App Income
Coapp Income

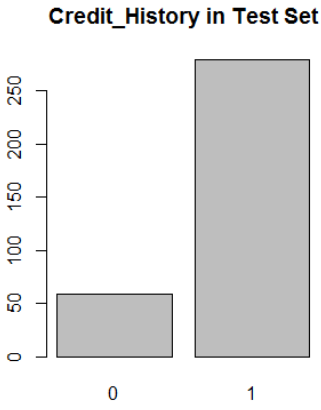

Loan Amount in Training Set
0
100
300
500
700


Loan Amount in Test Set
100
200
300
400
500


Loan_Amount_Term in Training Set
Frequency
0
100
200
300
400
500
0
100
200
300
400
train_set$Loan_Amount_Term


Loan_Amount_Term in Test Set
Frequency
0
50
100
150
200
250
300
0
100
200
300
400
test_set$Loan_Amount_Term


Credit_History in Training Set
0
100
200
300
400
0
1


Credit_History in Test Set
0
50
100
150
200
250
0
1

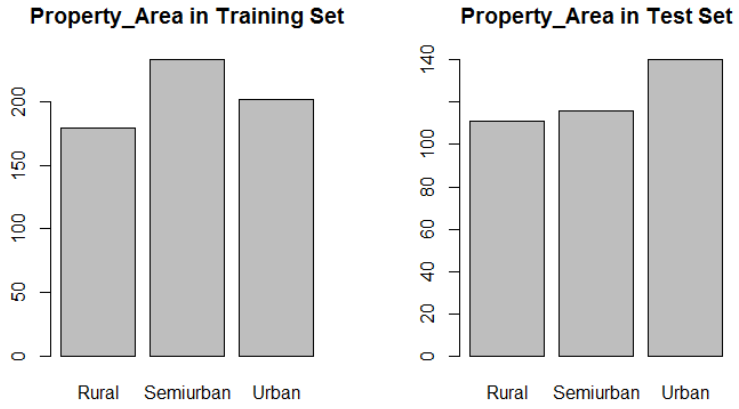
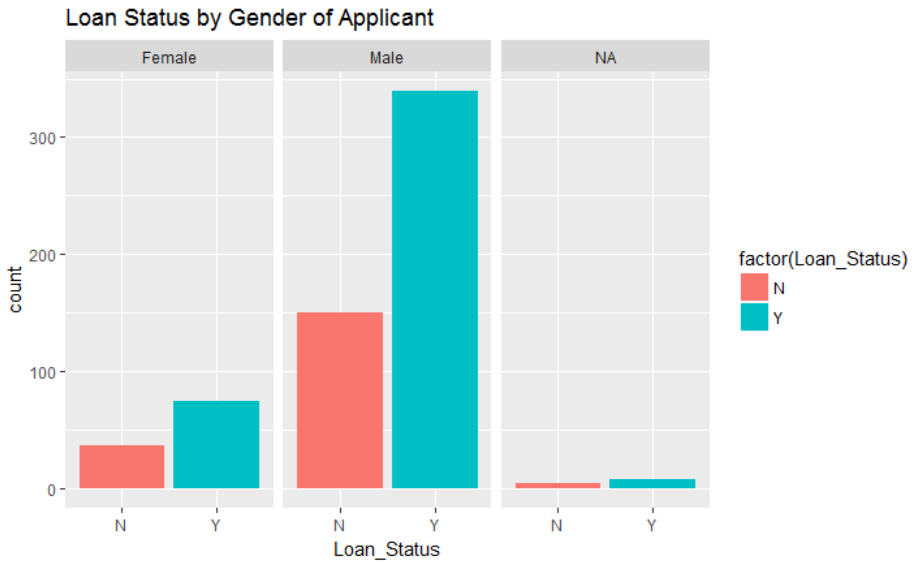


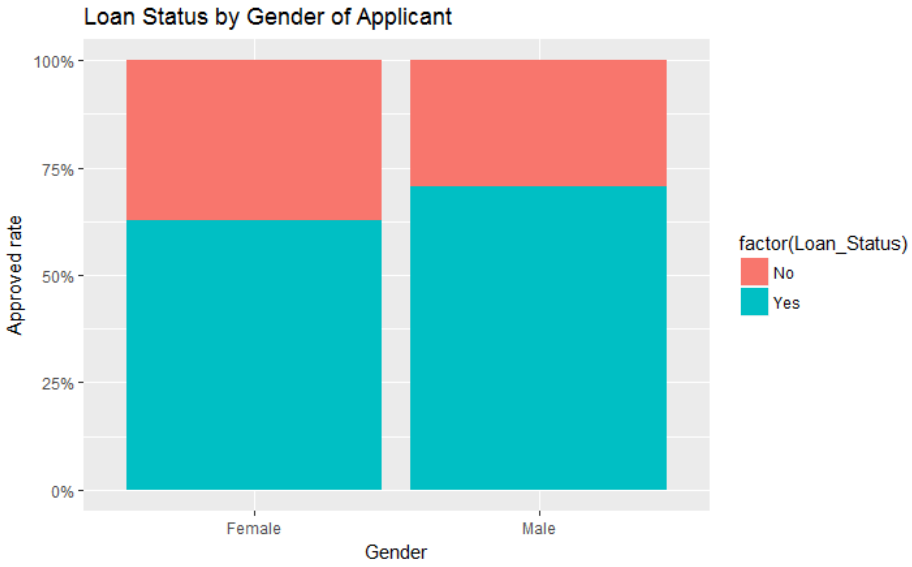
Figure 1: Exploratory graphs of the data in the training and test sets.

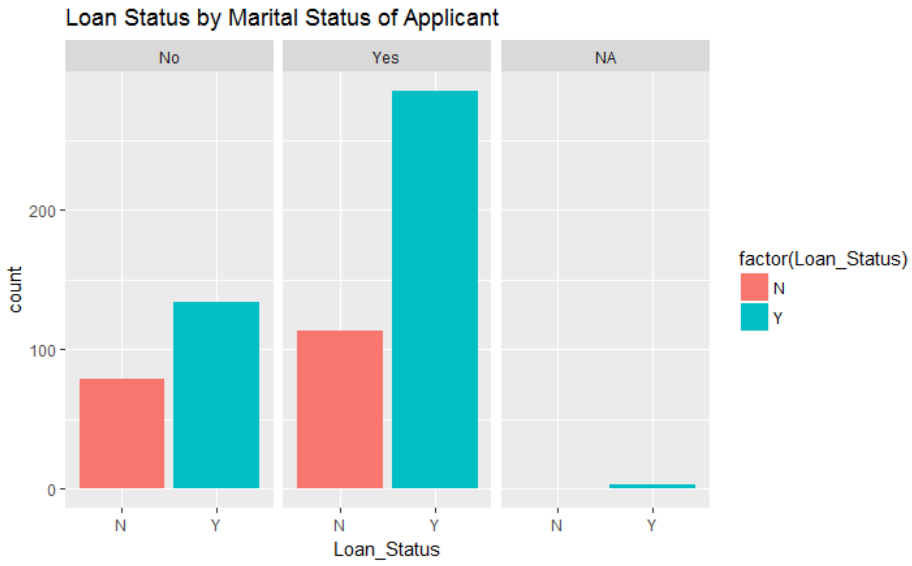
Bivariate visualization

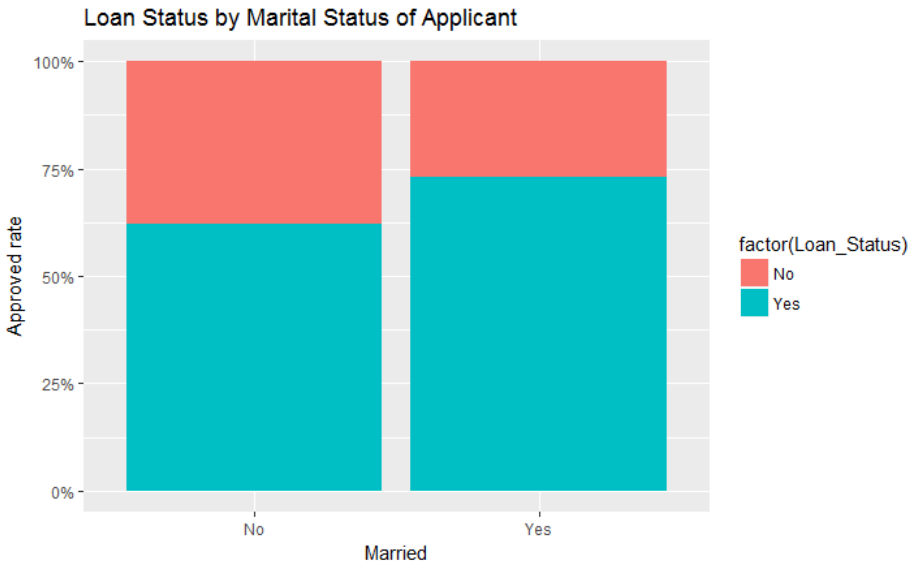
The next figures show the exploratory graphs of the response variable against each of the predictors individually in the training and test sets. Some interesting findings from these plots are:

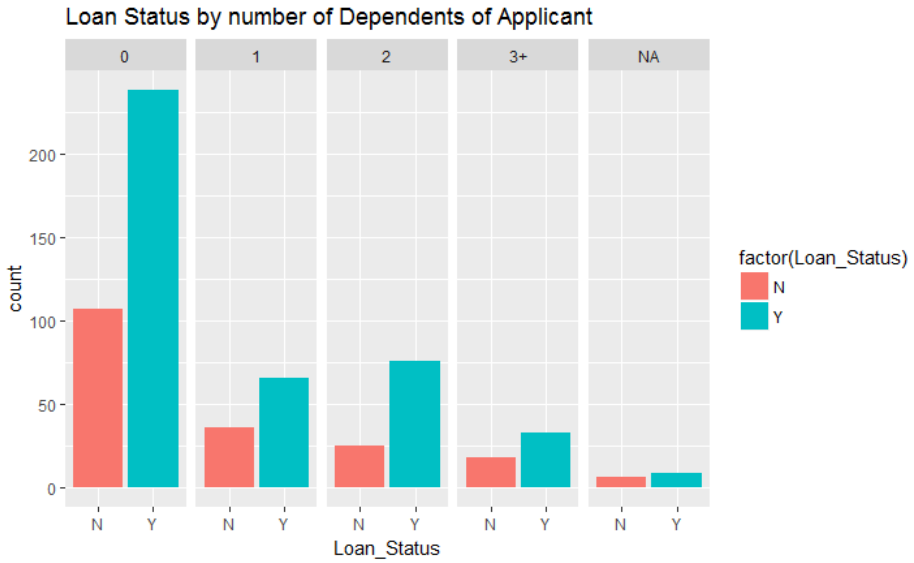
- A slightly larger proportion of female applicants are refused than male ones.
- A larger proportion of unmarried applicants are refused than married ones.
- A smaller proportion of applicants with 2 dependents are refused than applicants with other number of dependents.
- A larger proportion of non graduates are refused than graduates.
- Applicants that are self employed have a slightly worse approved rate.
- The majority of loans have a term of 360 months.
- Credit_History seems to be a very important predictor variable. The vast majority of applicants whose credit history doesn't meet guidelines are refused.
- It's easier to get a loan if the property is semi-urban and harder if it's rural.
- ApplicantIncome doesn't seem to have a significant influence on loan approval.
- CoapplicantIncome has some influence on loan approval.
- LoanAmount has some influence on loan approval.

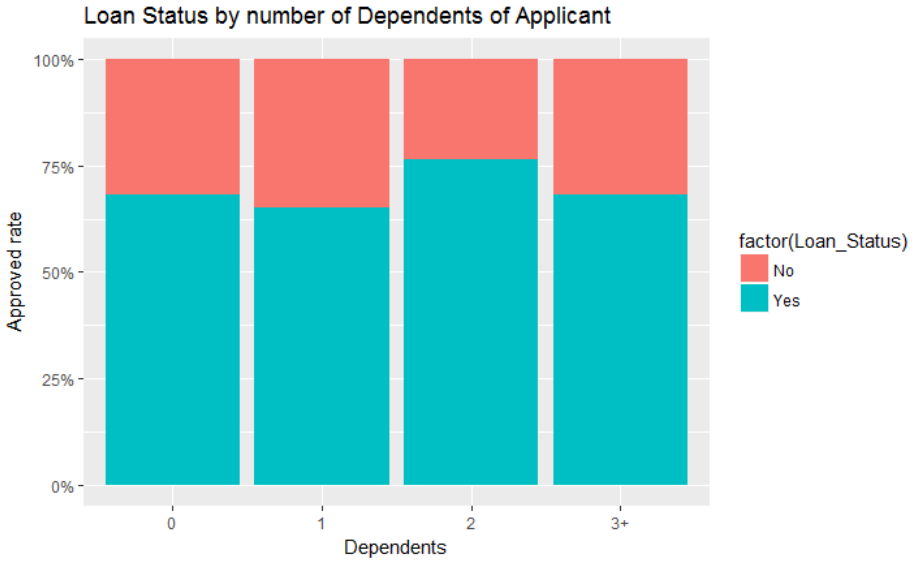

Loan Status by Gender of Applicant
Female
Male
NA
300
200
100
0
count
N
Y
Loan_Status
factor(Loan_Status)
N
Y

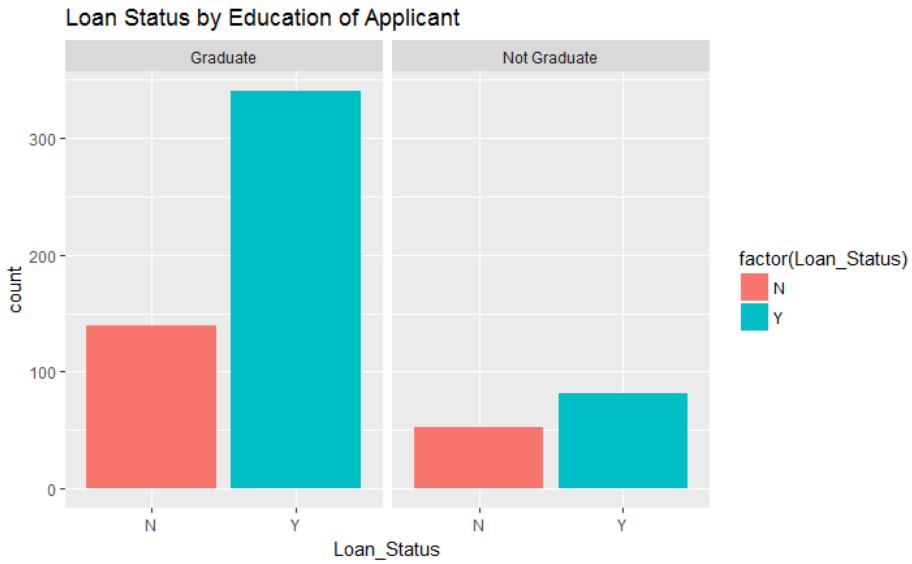

Loan Status by Gender of Applicant
100%
75%
50%
25%
0%
Approved rate
Female
Male
Gender
factor(Loan_Status)
No
Yes

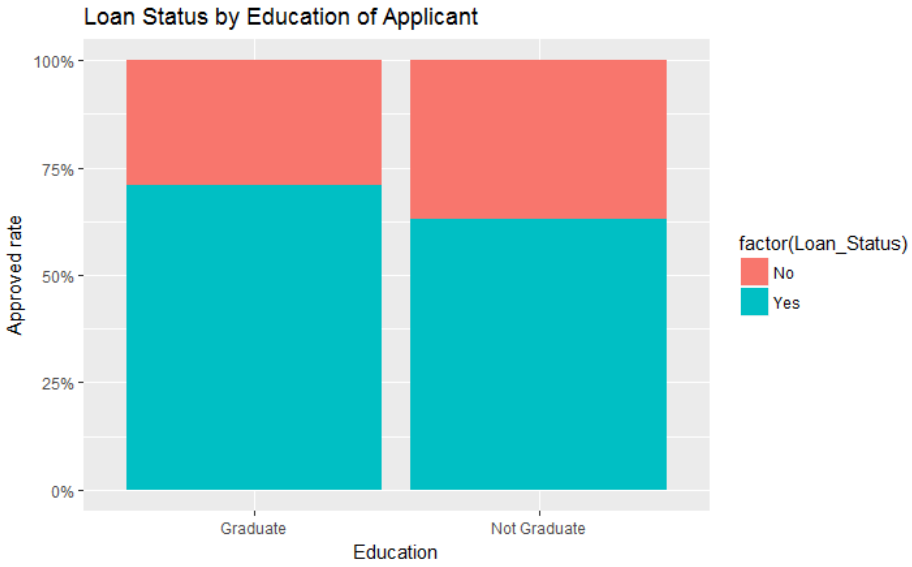

Loan Status by Marital Status of Applicant
No
Yes
NA
200
100
0
count
N
Y
Loan_Status
factor(Loan_Status)
N
Y

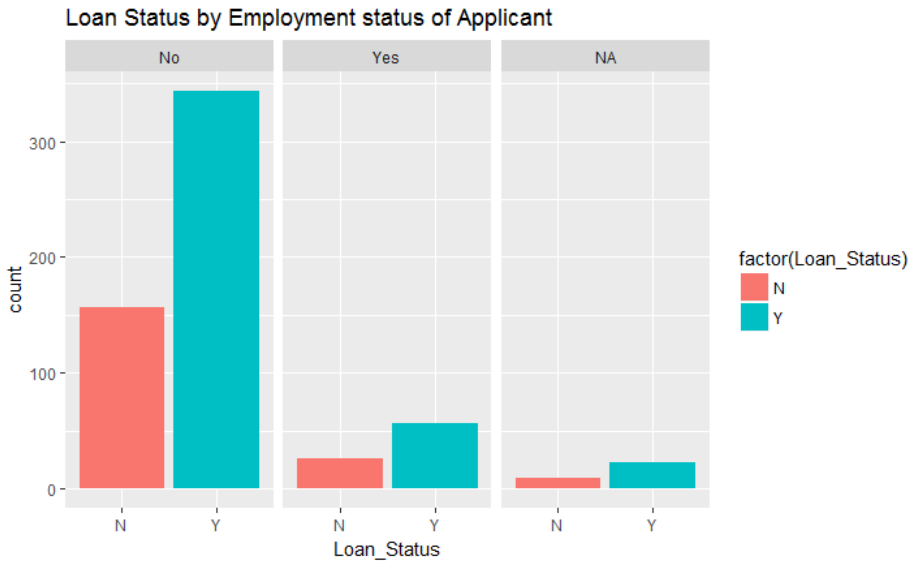

Loan Status by Marital Status of Applicant
100%
75%
50%
25%
0%
Approved rate
No
Yes
Married
factor(Loan_Status)
No
Yes

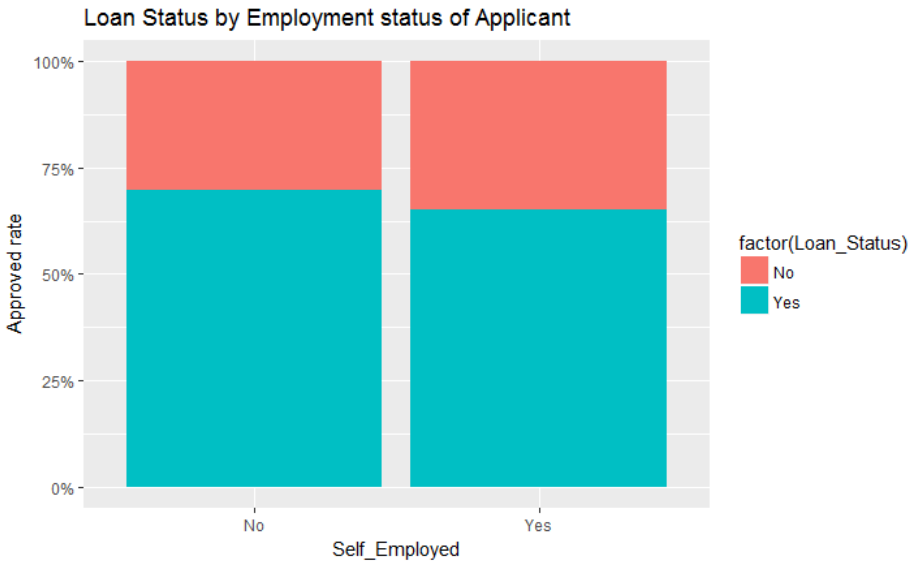

Loan Status by number of Dependents of Applicant
0
1
2
3+
NA
200
150
100
50
0
count
N
Y
Loan_Status
factor(Loan_Status)
N
Y

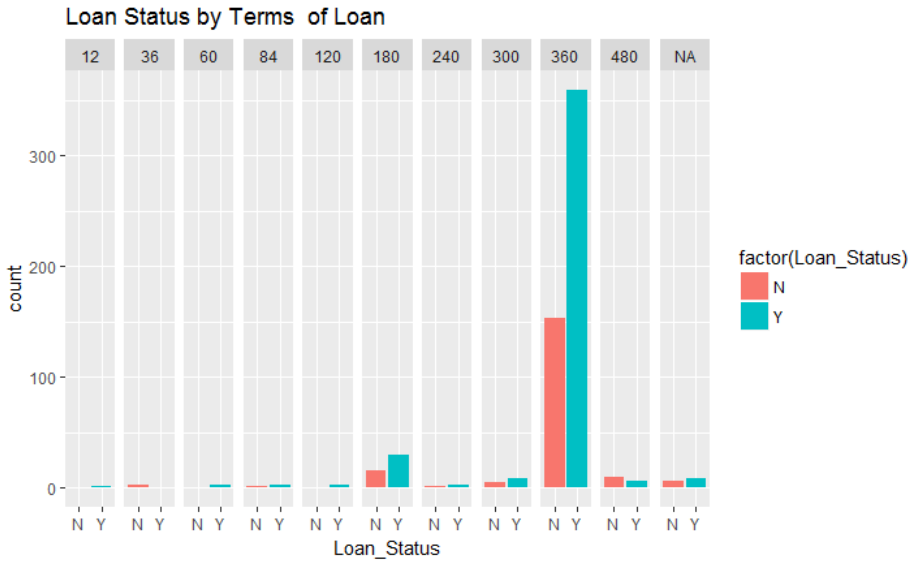

Loan Status by number of Dependents of Applicant
100%
75%
50%
25%
0%
Approved rate
0
1
2
3+
Dependents
factor(Loan_Status)
No
Yes

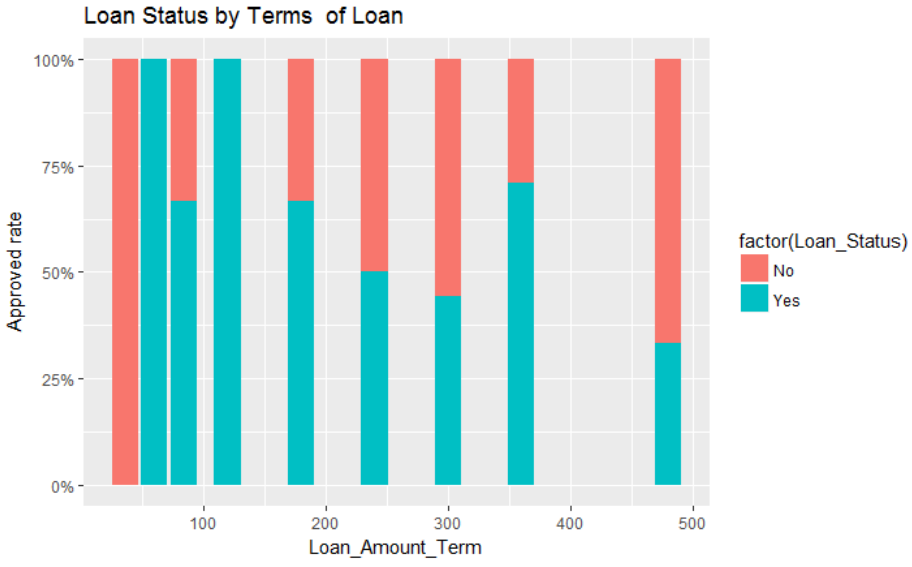

Loan Status by Education of Applicant
Graduate
Not Graduate
300
200
100
0
count
N
Y
N
Y
Loan_Status
factor(Loan_Status)
N
Y

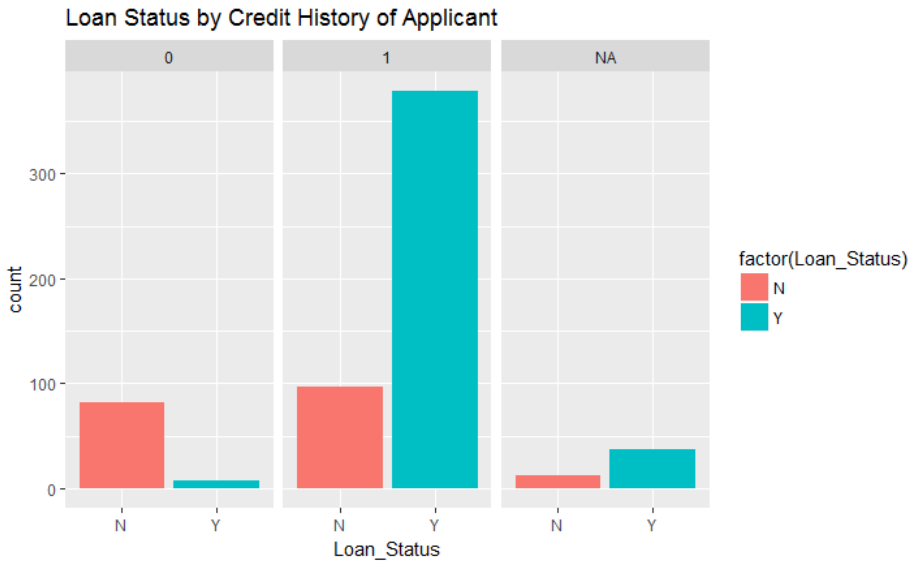

Loan Status by Education of Applicant
100%
75%
50%
25%
0%
Approved rate
Graduate
Not Graduate
Education
factor(Loan_Status)
No
Yes

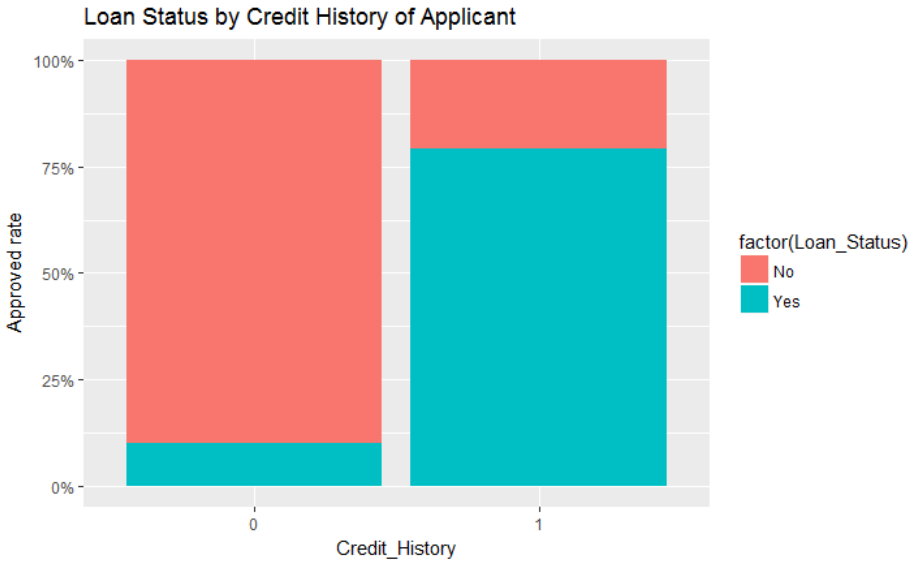

Loan Status by Employment status of Applicant
No
Yes
NA
300
200
100
0
count
N
Y
N
Y
N
Y
Loan_Status
factor(Loan_Status)
N
Y


Loan Status by Employment status of Applicant
100%
75%
50%
25%
0%
Approved rate
No
Yes
Self_Employed
factor(Loan_Status)
No
Yes

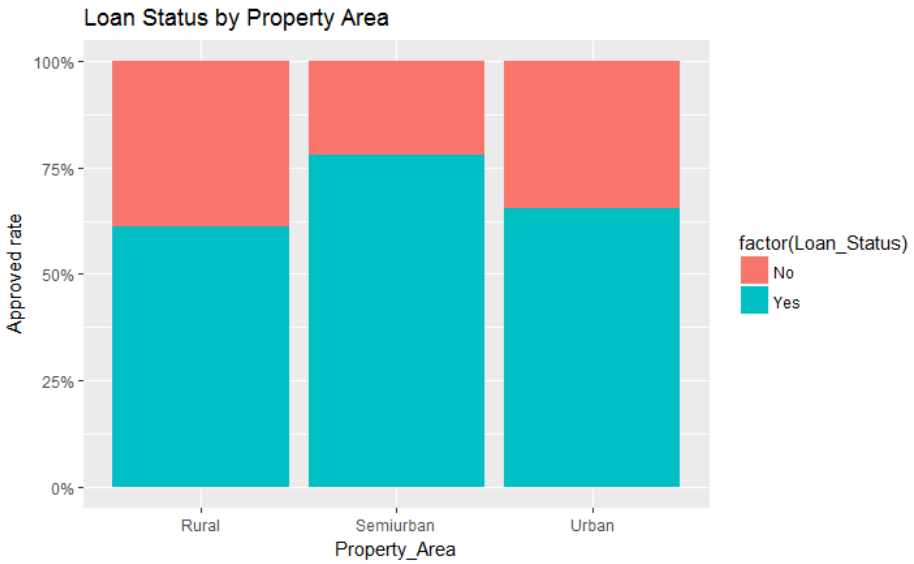

Loan Status by Terms of Loan
12
36
60
84
120
180
240
300
360
480
NA
300
200
100
0
count
N Y
Loan_Status
factor(Loan_Status)
N
Y

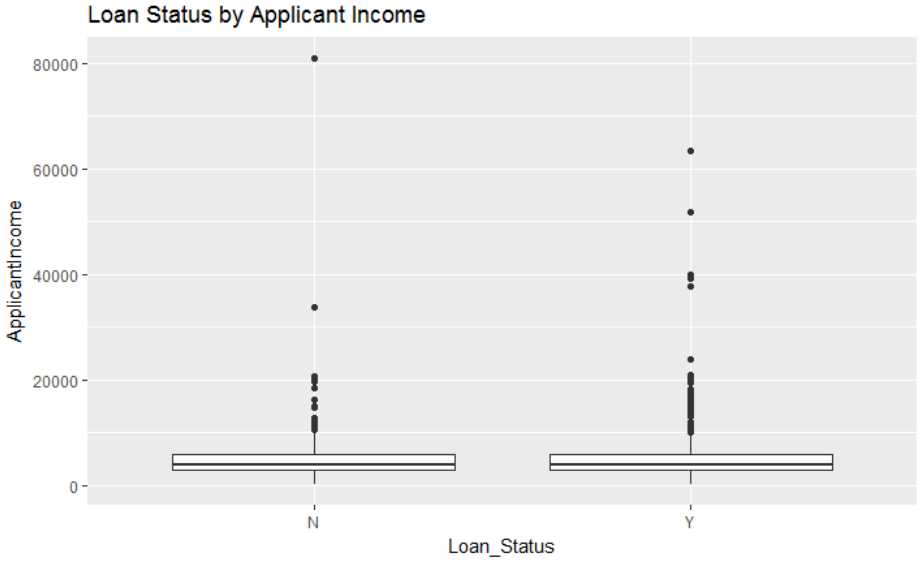

Loan Status by Terms of Loan
100%
75%
50%
25%
0%
Approved rate
100
200
300
400
500
Loan_Amount_Term
factor(Loan_Status)
No
Yes

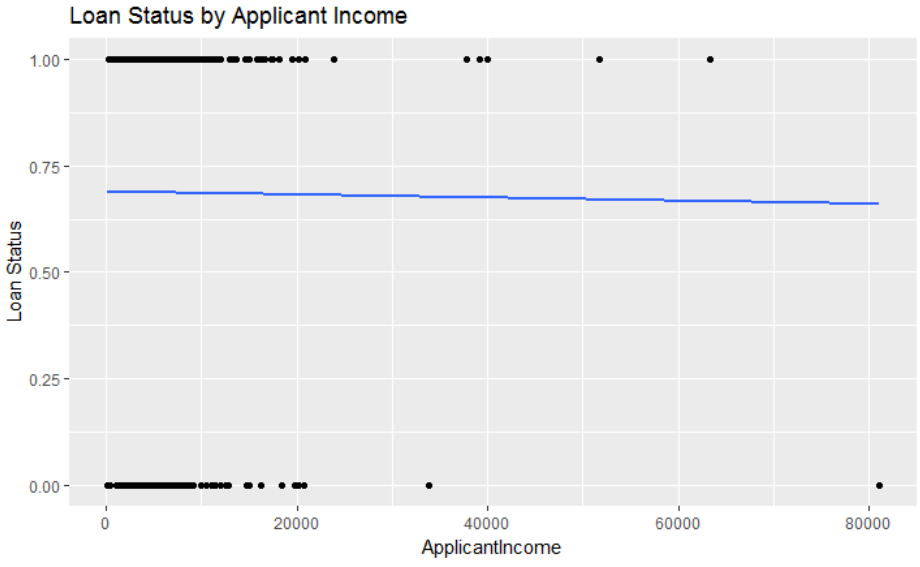

Loan Status by Credit History of Applicant
0
1
NA
300
200
100
0
count
N
Y
N
Y
N
Y
Loan_Status
factor(Loan_Status)
N
Y

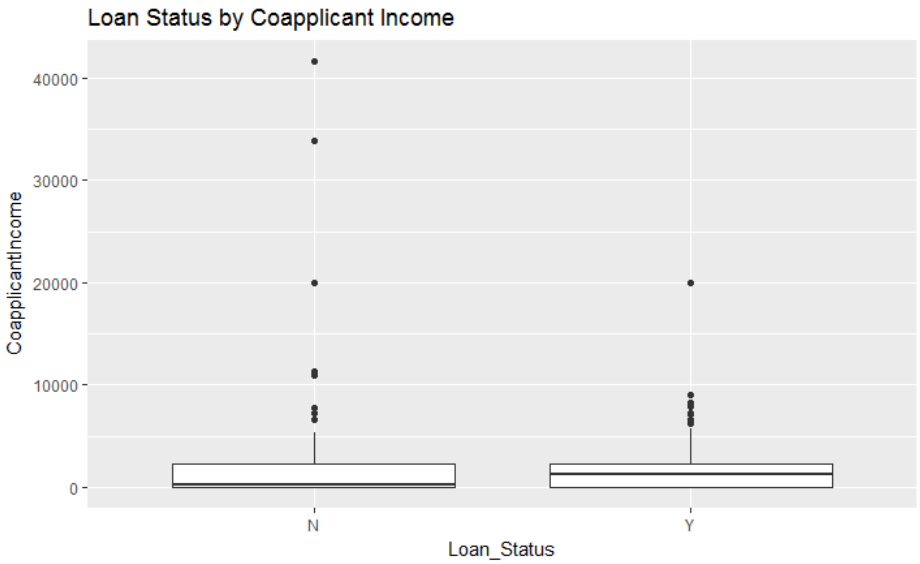

Loan Status by Credit History of Applicant
100%
75%
50%
25%
0%
Approved rate
0
1
Credit_History
factor(Loan_Status)
No
Yes

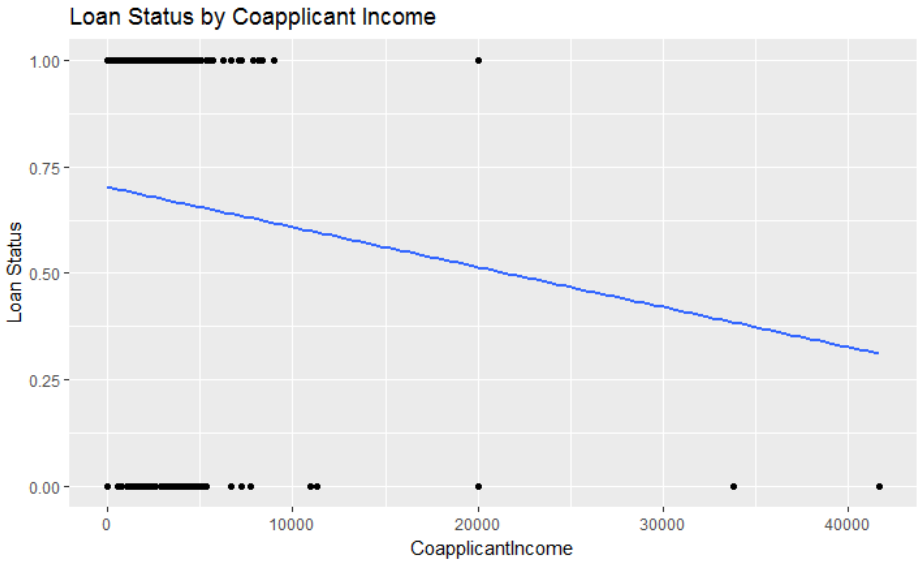

Loan Status by Property Area
Rural
Semiurban
Urban
150
100
50
0
count
N
Y
N
Y
N
Y
Loan_Status
factor(Loan_Status)
N
Y

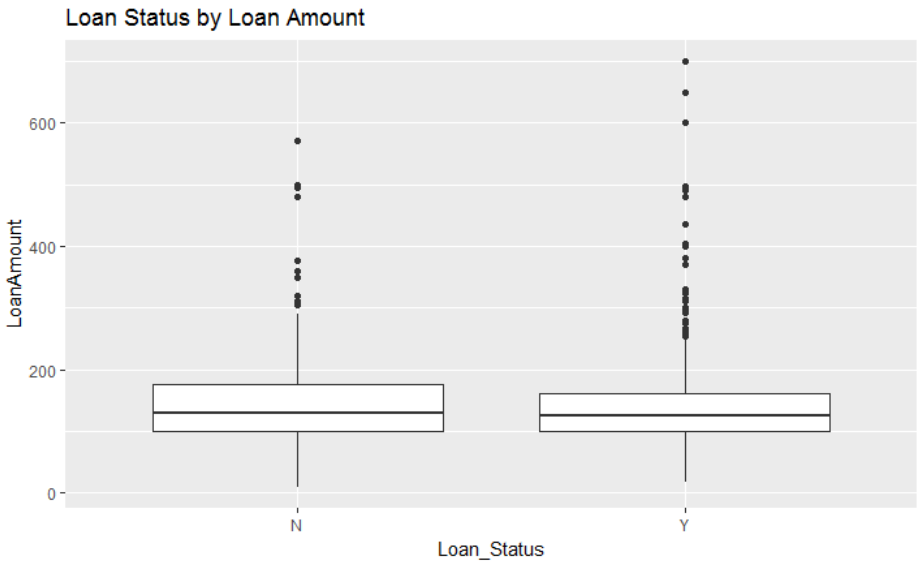

Loan Status by Property Area
100%
75%
50%
25%
0%
Approved rate
Rural
Semiurban
Urban
Property_Area
factor(Loan_Status)
No
Yes


Loan Status by Applicant Income
80000
60000
40000
20000
0
ApplicantIncome
N
Y
Loan_Status


Loan Status by Applicant Income
1.00
0.75
0.50
0.25
0.00
Loan Status
0
20000
40000
60000
80000
ApplicantIncome


Loan Status by Coapplicant Income
40000
30000
20000
10000
0
CoapplicantIncome
N
Y
Loan_Status


Loan Status by Coapplicant Income
1.00
0.75
0.50
0.25
0.00
Loan Status
0
10000
20000
30000
40000
CoapplicantIncome


Loan Status by Loan Amount
600
400
200
0
LoanAmount
N
Y
Loan_Status

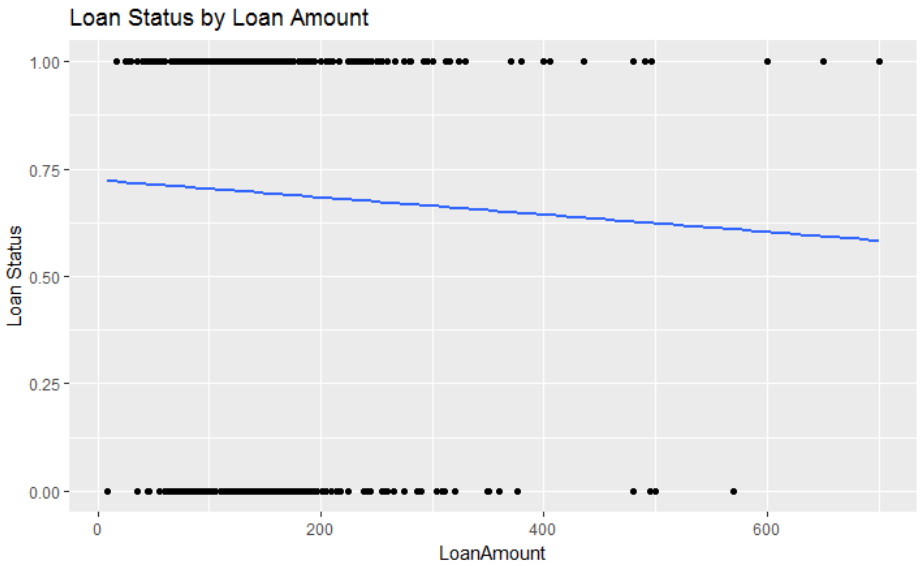


Figure 2: Exploratory graphs of the response variable against each of the predictors individually in the training and test sets.

Predictor Importance

Now, we will compute the predictor importance to evaluate the relative importance of each predictor variable. For categorical variables, the predictor importance is calculated as 1-Pval from Pearson Chi-square test and for continuous variables it is calculated as 1-Pval from ANOVA F-Test for Equity of Mean. The next table shows the predictor importance of the variables in descending (highest to lowest) order.

```
Credit_History 1
Property_Area 0.997864
Married 0.9656062
Education 0.9569004
Loan_Amount_Term 0.8781424
CoapplicantIncome 0.8570517
LoanAmount 0.6352638
Dependents 0.6321493
Gender 0.291347
ApplicantIncome 0.09271219
Self_Employed 2.553513e-15
```

From the above univariate analysis, we see that Credit_History, Property_Area, Married and Education are the most significant predictors. Surprisingly, ApplicantIncome is a very weak predictor, while CoapplicantIncome is a significant predictor.

We can also conclude that it is more suitable to not ignore the outliers from ApplicantIncome, CoapplicantIncome and LoanAmount as the most significant predictors are all categorical.

Missing values Imputation

First, we have to find how many missing values we have and in which variables and replace them with sensible values. We see that we have missing values in Gender, Married, Dependents, Self_Employed, LoanAmount, Loan_Amount_Term and Credit_History variables in the training set and Gender, Dependents, Self_Employed, LoanAmount, Loan_Amount_Term and Credit_History variables in the test set. To tackle this problem, we are going to predict the missing values with the full data set, which means that we need to combine the training and test sets together. The missing values in the full set can be seen in the next table.

```
      [,1]              [,2]
[1,] "Gender"          "24"
[2,] "Married"         "3"
[3,] "Dependents"      "25"
[4,] "Self_Employed"   "55"
[5,] "LoanAmount"      "27"
[6,] "Loan_Amount_Term" "20"
[7,] "Credit_History"  "79"
```

Variable "Married"

Only three passengers have missing Married values. We will infer these values based on CoapplicantIncome (data that seem relevant). We will consider the Married variable as "No", when the co-applicant income is zero, and "Yes" otherwise.

Variables "Gender" and "Dependents"

Replacing the missing values from Gender and Dependents with the most frequent category, might not be the best idea, since they may differ by groups of applicants. We see that there is only one applicant with both these values missing. This applicant is not married but has higher income than the co-applicant. So, the next step is to investigate which gender has higher income. Since males have higher income, we will consider the missing value as "Male".

All the other missing observations have only one of these variables missing. First, we will examine the missing values of Dependents.

Variable Dependents

We will examine the Dependents variable in relation to Married variable (data that seem relevant).

```
             0          1          2         3+
No  0.81656805 0.10650888 0.04142012 0.03550296
Yes 0.43527508 0.20064725 0.23624595 0.12783172
```

We see that the majority of unmarried people do not have dependents. However, ~ 19% of unmarried people have 1 or more dependents, so we can't impute missing dependents with zero for unmarried people.

We are going to predict missing Dependents variables using three different methods, in order to investigate which one achieves the best results.

The first method is the modal imputation. For categorical variables, an easy way to impute the values is to use modal imputation, or in other words, impute cases with the mode, or most common value. The second way is to use rpart (recursive partitioning for regression) and the third way is to perform mice (Multivariate Imputation by Chained Equations) imputation. For this imputation, since Dependents is a factor variable with more than two levels, multinomial logistic regression is used.

Once we get the predicted values, we can compare the original distribution of Dependents variables with the predicted distribution of Dependents, using modal imputation, rpart imputation and mice imputation in order to select the best prediction. The distributions can be seen in Figure 3.

We see that the best prediction is achieved using mice imputation, so we will use these values to replace the missing Dependents values.

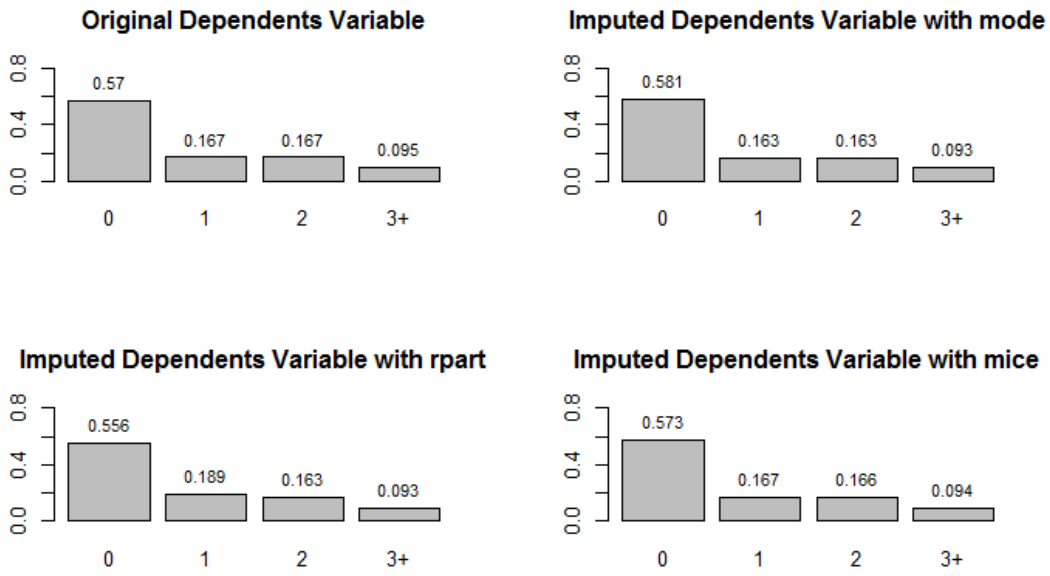


Figure 3: Comparison of the original distribution of Dependents variables with the predicted distribution of Dependents, using modal imputation, rpart imputation and mice imputation.

Variable Gender

We will use mice imputation again to predict the missing Gender values. For this prediction, since Gender is a factor variable with two levels, logistic regression is used.

The next table shows the percentages of Gender before and after the prediction. We see that the percentages of Gender before and after the prediction look very similar, which implies that our prediction was correct.

```
        Original                Predicted

   Female      Male        Female      Male
0.1899791 0.8100209     0.1926606 0.8073394
```

Variable Self_Employed

The next table shows what percentage of applicants is self employed.

```
        No       Yes
0.8714903 0.1285097
```

Since ~87% of applicants are not self employed, it would be safe to impute the missing values as "No", as there is a high probability of success. However we can investigate if there are any hidden patterns.

For this purpose, we will examine the relationship between Self_Employed variable and Gender and Education variables (data that seem relevant).

```
  Self_Employed Gender    Education   x
1            No Female     Graduate 127
2           Yes Female     Graduate  17
3            No   Male     Graduate 499
4           Yes   Male     Graduate  77
5            No Female Not Graduate  30
6           Yes Female Not Graduate   4
7            No   Male Not Graduate 151
8           Yes   Male Not Graduate  21
```

Figure 4 illustrates the relationship among Self_Employed, Gender and Education variables. We see that the vast majority of applicants are not self employed regardless of their gender and education. Therefore, we can impute the missing Self_Employed values using the mode="No".

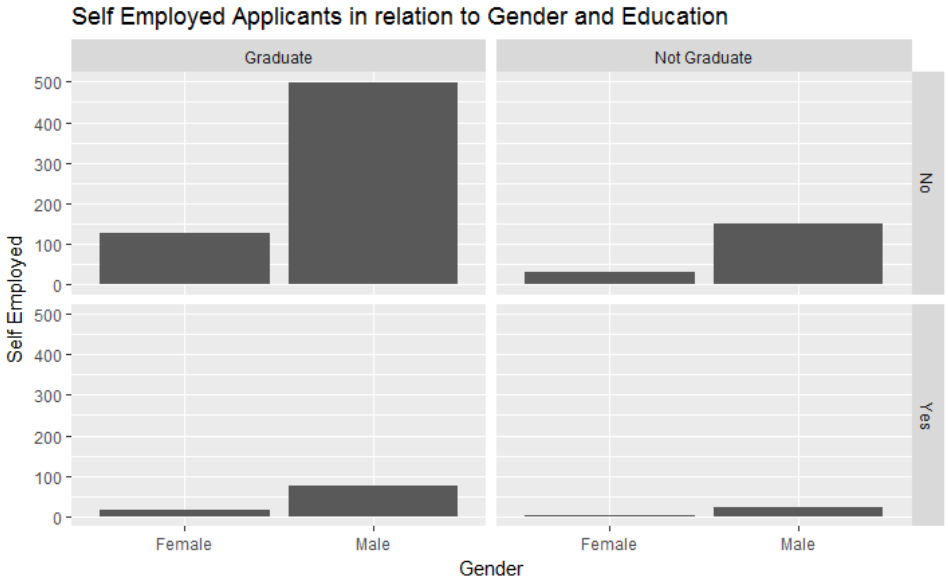


Figure 4: Self employed applicants in relation to gender and education.

The next table shows what percentage of applicants is self employed after the prediction.

```
        No       Yes
0.8786952 0.1213048
```

We see that the percentages of self employed applicants before and after the prediction look very similar, which implies that our prediction was correct.

Variable Credit_History

From the bivariate analysis we conducted before, we noticed that Credit_History is a high impact variable. If credit history is not available, it possibly means that the applicant has not had many credit activities in the past. The safest approach is to treat this variable as a separate category. Therefore, we will replace the missing values with “Not Available”.

Variable LoanAmount

The next table shows a summary of the LoanAmount variable, which includes the mean, the median, the min and max values, the 1st and 3rd quartiles and the number of missing values.

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 9.0   100.0   126.0   142.5   162.0   700.0      27
```

We will examine the relationship between the LoanAmount variable and Education and Self_Employed variables (data that seem relevant). We will replace the missing LoanAmount values with the median values of the LoanAmount corresponding to Self_Employed and Education variables. The next table provides the median LoanAmount values for all the groups of unique values of Self_Employed and Education variables.

```
     Education Self_Employed LoanAmount
1     Graduate            No        130
2 Not Graduate            No        117
3     Graduate           Yes        150
4 Not Graduate           Yes        130
```

After substitution is done, the summary of LoanAmount variable becomes:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
9.0   101.0   128.0   142.1   160.0   700.0
```

We see that the distribution of LoanAmount variable before and after the prediction looks very similar, which implies that our prediction was correct.

Variable Loan_Amount_Term

The next table shows a summary of the Loan_Amount_Term variable, which includes the mean, the median, the min and max values, the 1st and 3rd quartiles and the number of missing values.

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 6.0   360.0   360.0   342.2   360.0   480.0      20
```

We see that the 1st quartile, median and 3rd quartile values are 360, which means that the majority of Loan_Amount_Term values are 360. The next table shows how many unique observations belong to each loan amount term.

```
6  12  36  60  84 120 180 240 300 350 360 480
1   2   3   3   7   4  66   8  20   1 823  23
```

We will examine the relationship between the Loan_Amount_Term variable and the LoanAmount variable (data that seem relevant). The next table provides the median LoanAmount values for all the unique values of Loan_Amount_Term.

```
   Loan_Amount_Term LoanAmount
1                 6       95.0
2                12      185.5
3                36      118.0
4                60      139.0
5                84      108.0
6               120       25.0
7               180      117.0
8               240      100.0
9               300      135.5
10              350      133.0
11              360      130.0
12              480      113.0
```

There seems to be no linear relationship between LoanAmount and Loan_Amount_Term. The next table provides the median LoanAmount values and the number of observations for all unique values of Loan_Amount_Term.

```
   Loan_Amount_Term LoanAmount Freq
1                 6       95.0    1
2                12      185.5    2
3                36      118.0    3
4                60      139.0    3
5                84      108.0    7
6               120       25.0    4
7               180      117.0   66
8               240      100.0    8
9               300      135.5   20
10              350      133.0    1
11              360      130.0  823
12              480      113.0   23
```

We see that LoanAmount does not have a big influence on Loan_Amount_Term as it could be expected.

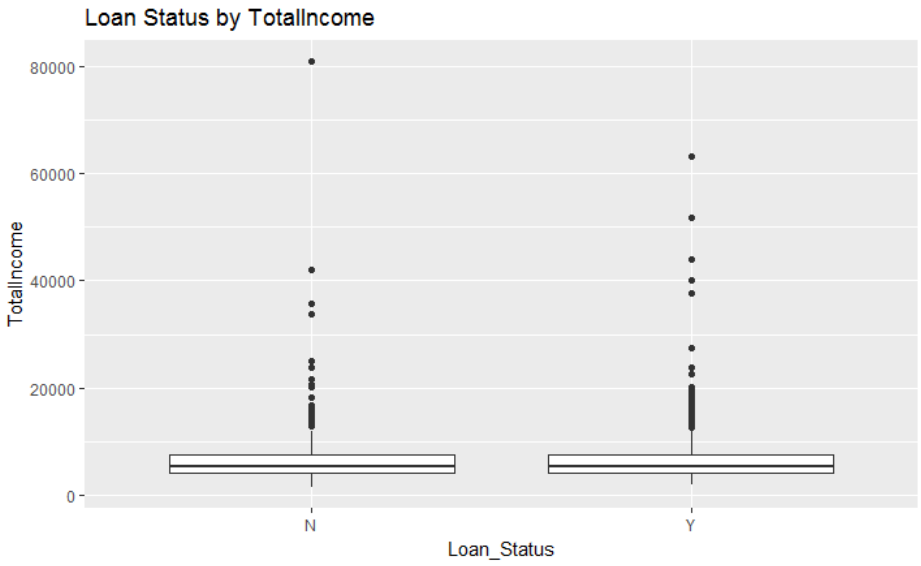
Since the vast majority of the loans had a term of 360 months, we will replace the missing values with the mode, or most common value, which is 360. In addition, since there are only a few unique Loan_Amount_Term values, we will convert this variable from integer to factor.

Feature Engineering

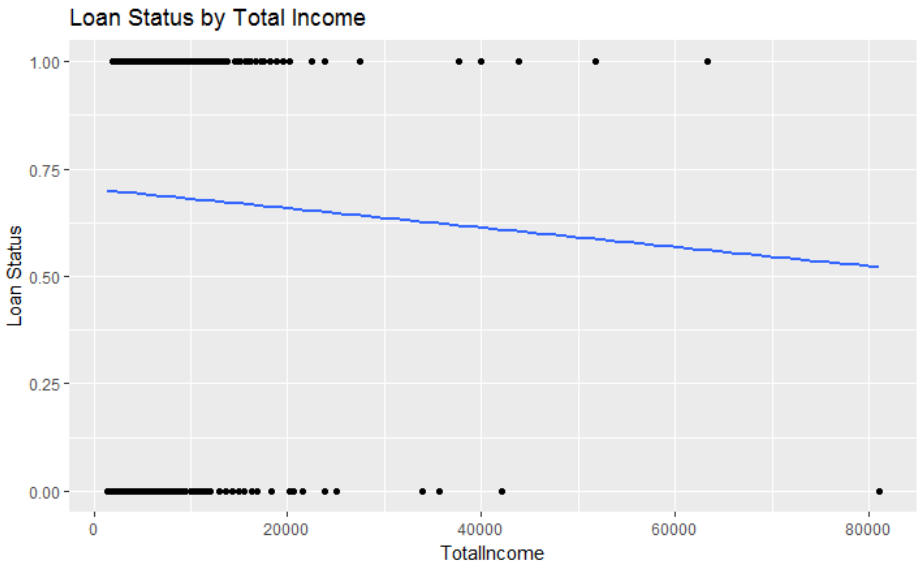
Some variables should be combined or transformed someway in order to build a better model. We will make several graphs and computations to determine which transformation of variables can be added as new features.

Variable TotalIncome

It is possible that some applicants have lower income but strong support co-applicants. So it might be a good idea to combine both incomes as total income. Figure 5 shows the Loan_Status by TotalIncome using a box plot and a dot plot with regression line. We see that TotalIncome has some influence on loan approval, but it is not very significant.



(a)

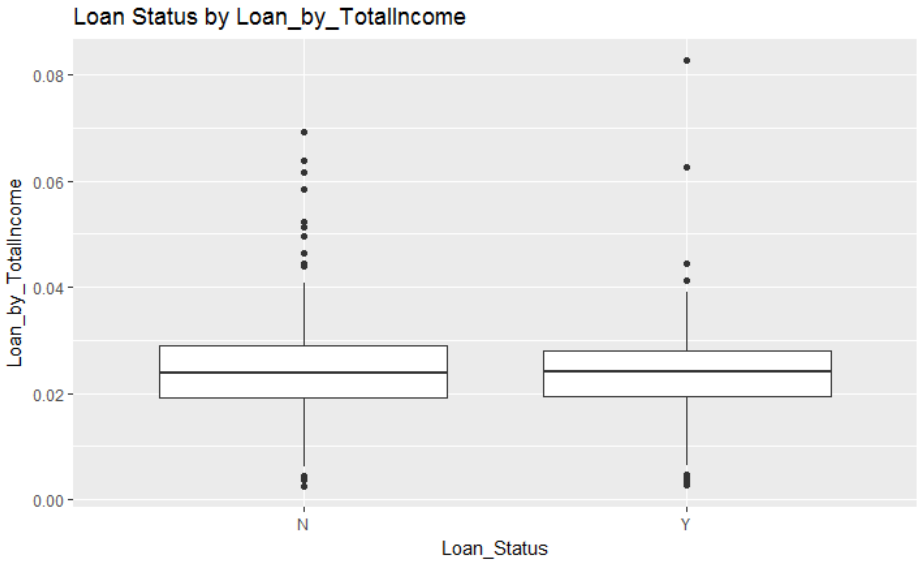


(b)

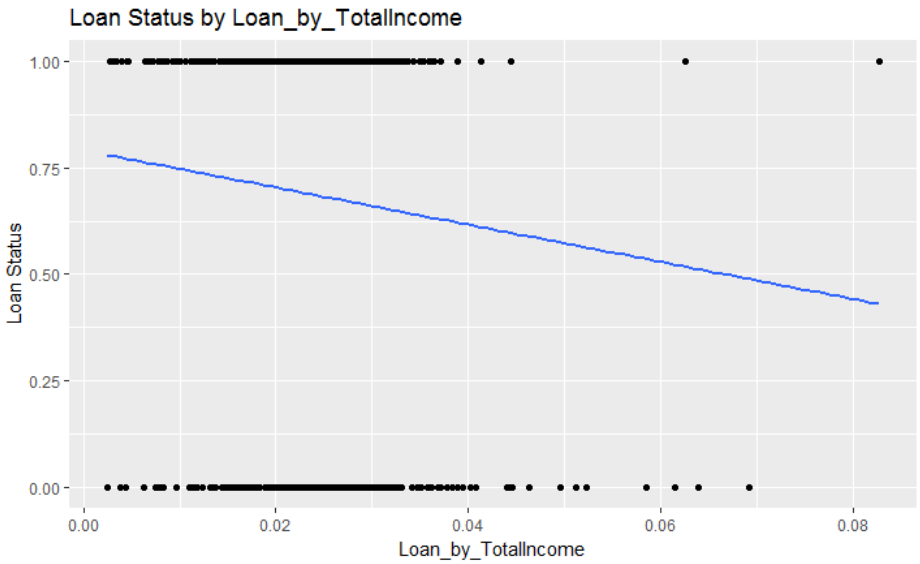
Figure 5: Loan_Status by TotalIncome using (a) a box plot and (b) a dot plot with regression line.

Variable Loan_by_TotalIncome

We will also create a variable as the loan amount divided by the sum of applicant and co-applicant income (TotalIncome). This variable gives an idea of how well the applicant is suited to pay back his loan. Figure 6 shows the Loan_Status by Loan_by_TotalIncome using a box plot and a dot plot with regression line. We observe that there seems to be a strong negative relationship between loan amount/total income and approved rate.



(a)

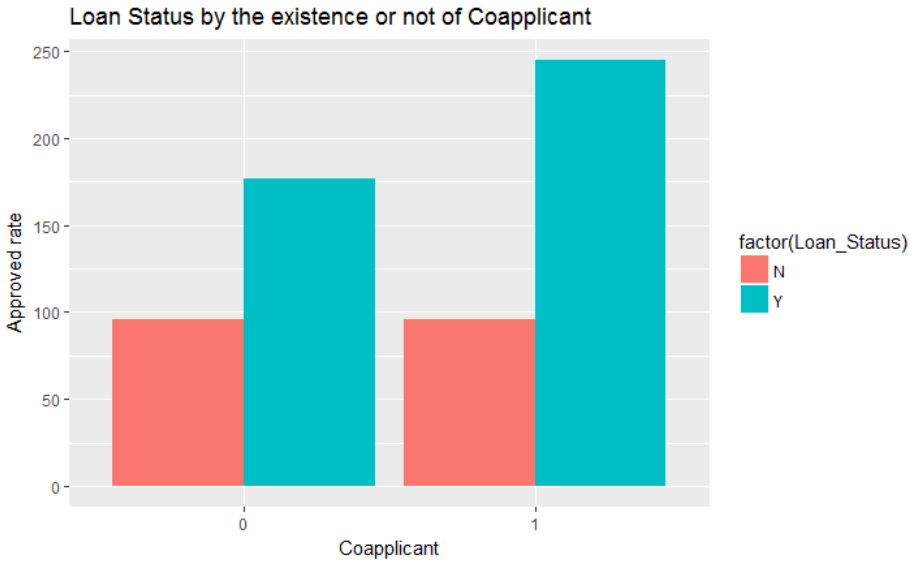


(b)

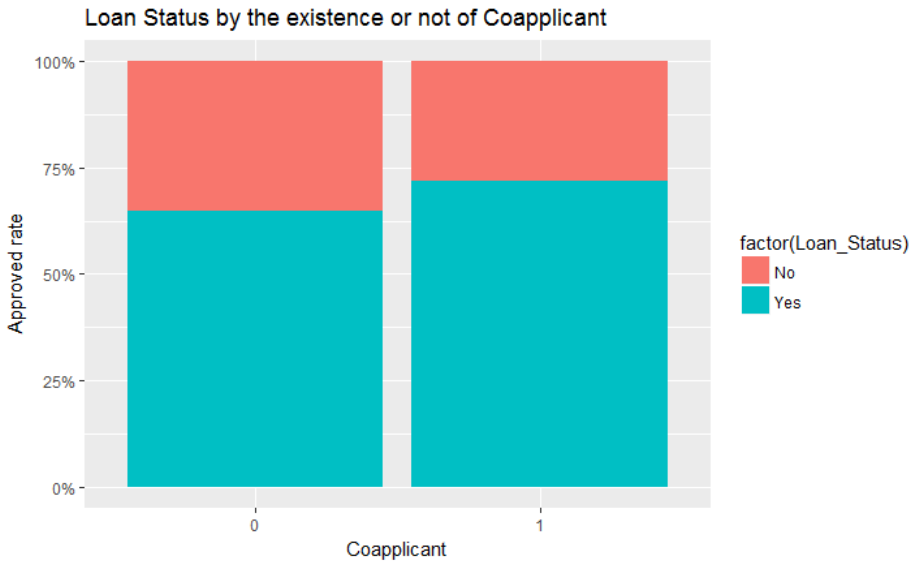
Figure 6: Loan_Status by Loan_by_TotalIncome using (a) a box plot and (b) a dot plot with regression line.

Variable Coapplicant

It may be advantageous to create a variable that indicates whether there is co-applicant or not. We will consider that a co-applicant exists if the CoapplicantIncome is larger than zero, or the applicant is married. Figure 7 shows the number and percentage of approved loans in relation to the existence or not of a co-applicant. We notice that most applicants have a co-applicant and that a loan is slightly more likely to be approved if there is a co-applicant.



(a)

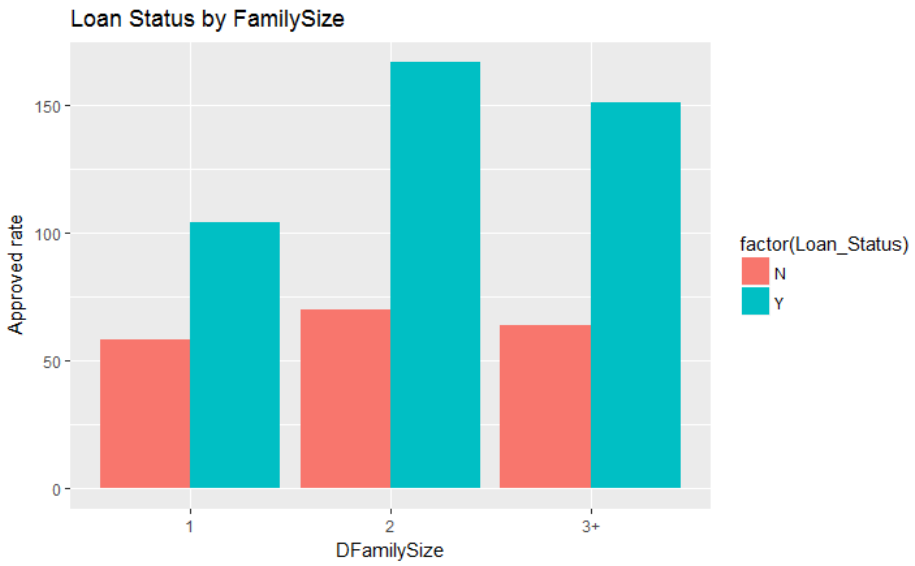


(b)

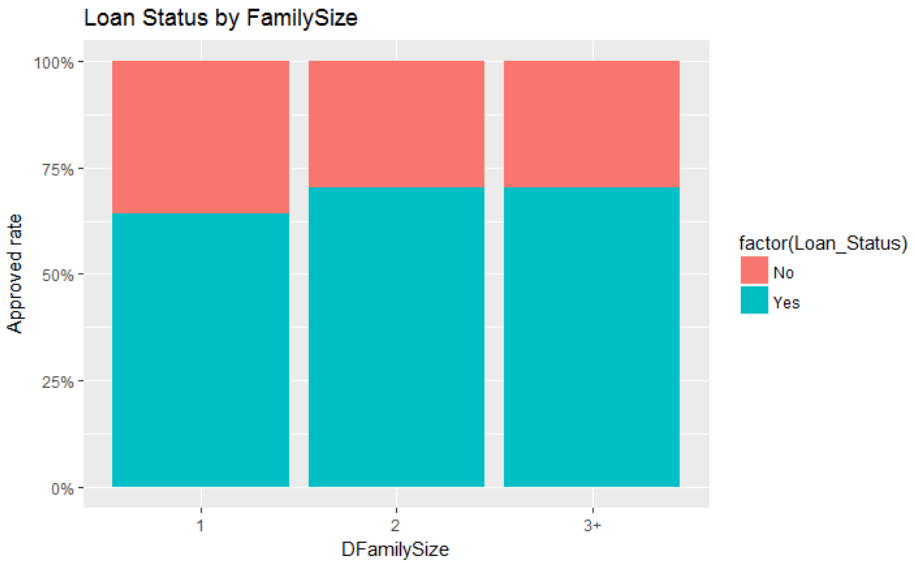
Figure 7: Figure (a) shows the number and Figure (b) the percentage of approved loans in relation to the existence or not of a co-applicant.

Variable FamilySize

We will create a FamilySize variable, which contains the applicant itself, the co-applicant (if exists) and the number of dependents. Figure 8 shows the number and percentage of approved loans in relation to the family size. We see that if an applicant has a co-applicant or one dependent (DFamilySize=2), or has a family of size >=3, it is slightly more likely to take the loan, but if the family size is 1, i.e., if there is only the applicant, it doesn't help to take the loan.



(a)

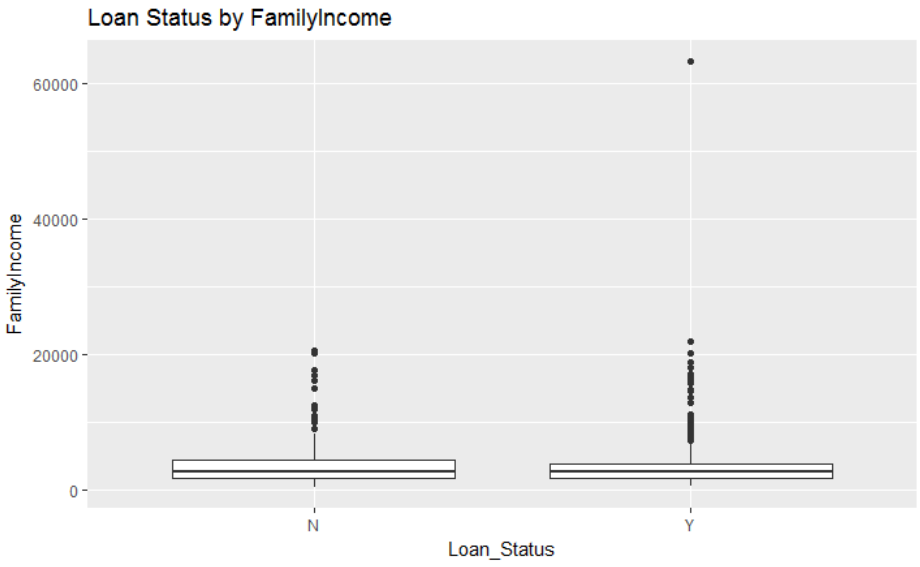


(b)

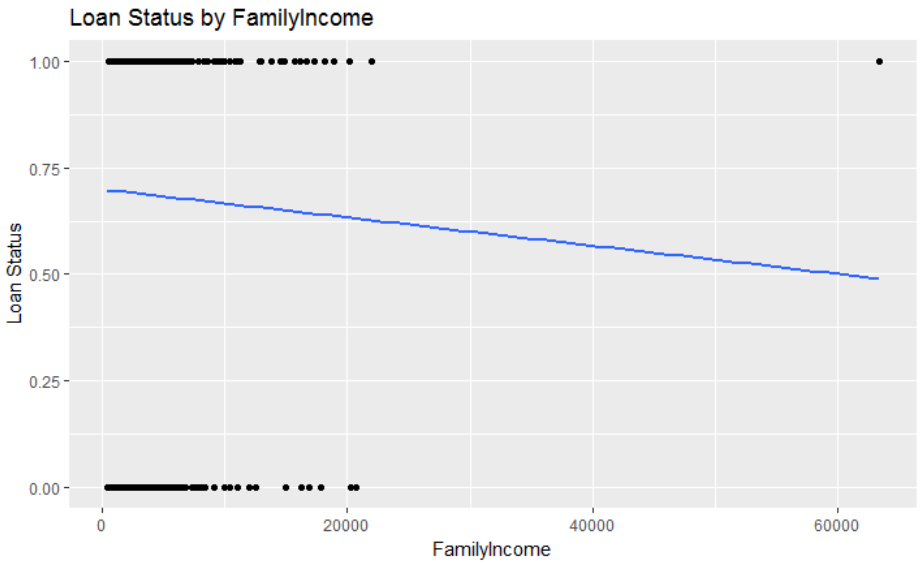
Figure 8: Figure (a) shows the number and Figure (b) the percentage of approved loans in relation to the family size.

Variable FamilyIncome

We can also create a variable as the sum of applicant and co-applicant income (TotalIncome) divided by the family size. This variable gives an idea of the actual income of a family. Figure 9 shows the Loan_Status by FamilyIncome using a box plot and a dot plot with regression line. As we can see, there seems to be a strong negative relationship between family income and approved rate.



(a)

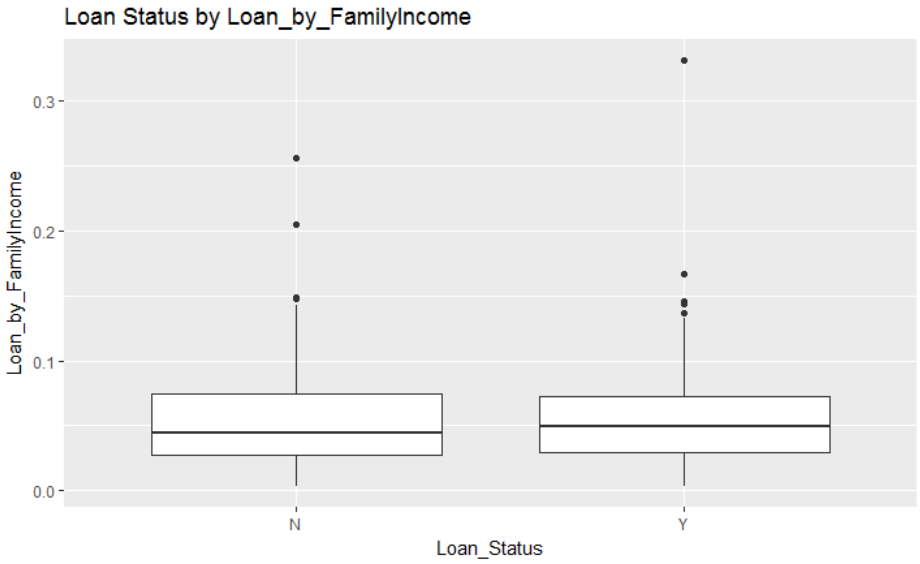


(b)

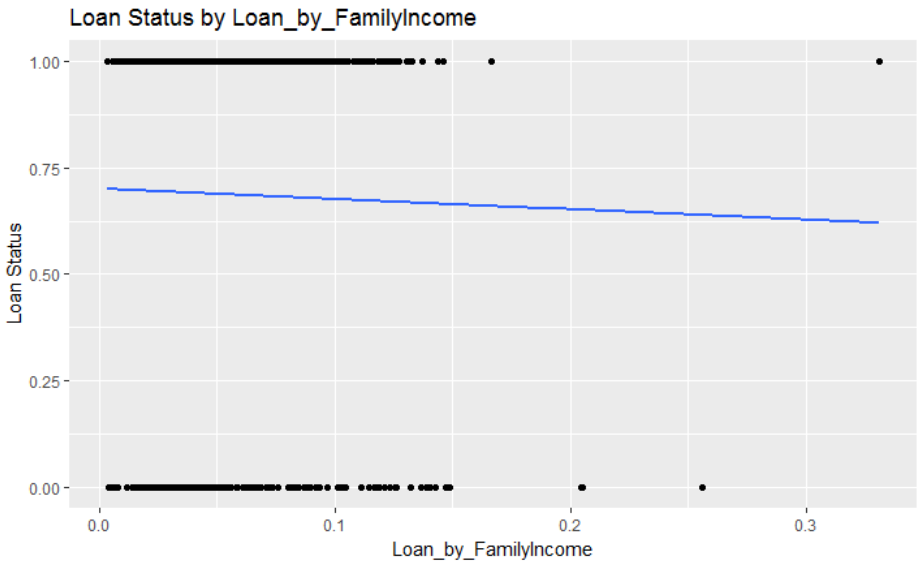
Figure 9: Loan_Status by FamilyIncome using (a) a box plot and (b) a dot plot with regression line.

Variable Loan_by_FamilyIncome

We can also create a variable as the loan amount divided by family income. This variable gives an idea of how well a family is suited to pay back its loan. Figure 10 shows the Loan_Status by Loan_by_FamilyIncome using a box plot and a dot plot with regression line. As we can see, there seems to be a weak negative relationship between loan amount/family income and approved rate.



(a)

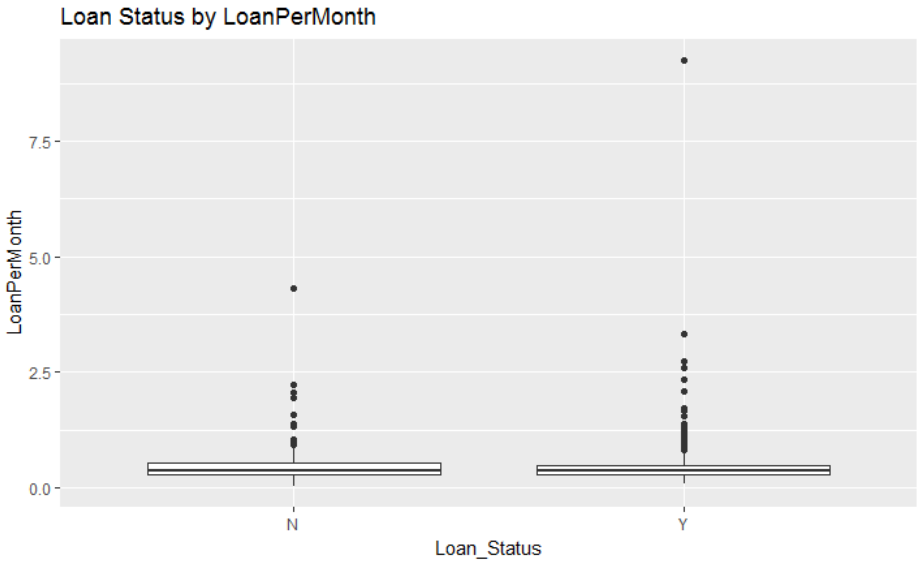


(b)

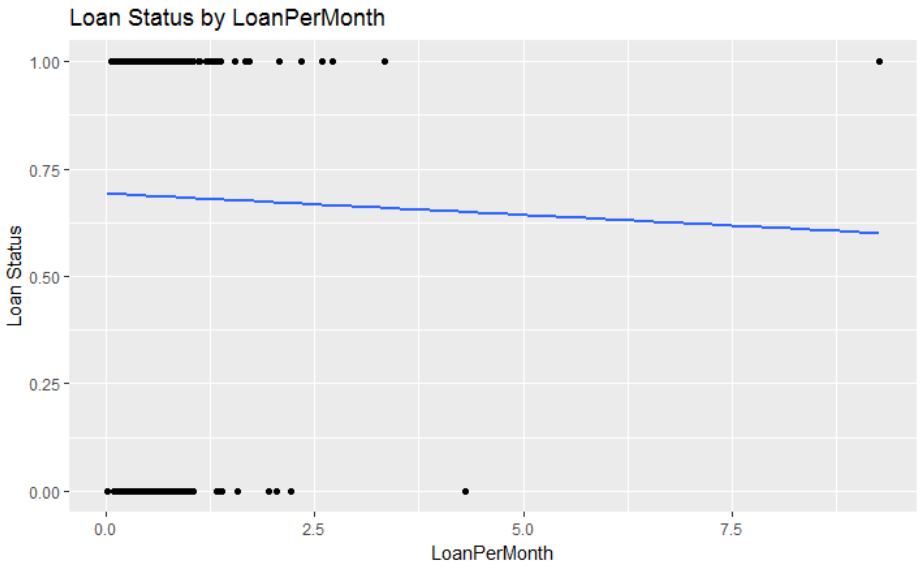
Figure 10: Loan_Status by Loan_by_FamilyIncome using (a) a box plot and (b) a dot plot with regression line.

Variable LoanPerMonth

Another useful variable would be the loan amount divided by loan term. This variable is actually the monthly installment made by a borrower to a lender. We will ignore the interest rate per month here. Figure 11 shows the Loan_Status by LoanPerMonth using a box plot and a dot plot with regression line. As we can see, there seems to be a weak negative relationship between loan per month and approved rate.



(a)



(b)

Figure 11: Loan_Status by LoanPerMonth using (a) a box plot and (b) a dot plot with regression line.

Extreme values treatment

In this problem, the extreme values are practically possible, i.e. some people might apply for high value loans due to specific needs. So instead of treating them as outliers, it's probably a good idea to take a log transformation of the monetary variables to nullify their effect.

Variable LoanAmount

The next table shows a basic summary LoanAmount.

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
9.0   101.0   128.0   142.1   160.0   700.0
```

Figure 12 shows the original distribution of LoanAmount with its log version. It is obvious that after the log transformation, the distribution looks much closer to normal and the effect of extreme values has been significantly subsided.

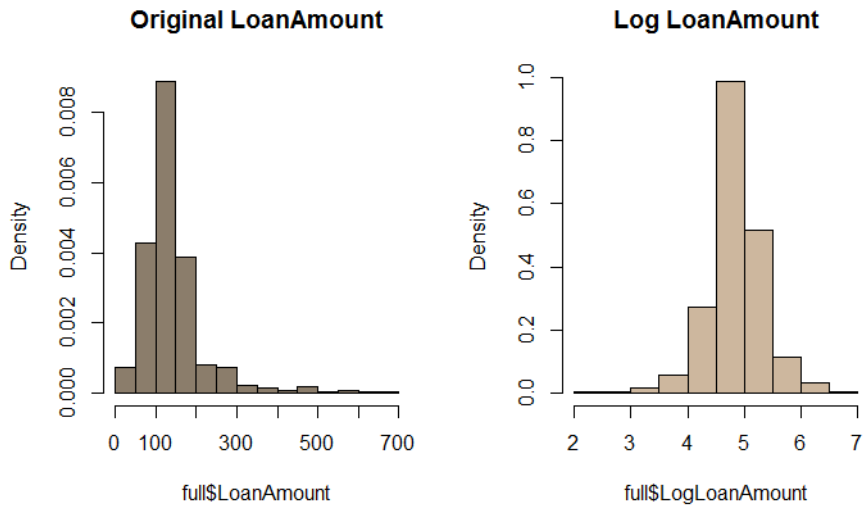


Figure 12: Comparison of the original distribution of LoanAmount with its log version.

Variable TotalIncome

Since ApplicantIncome and CoapplicantIncome are better predictors when they are combined in one variable, we will take the log transformation of TotalIncome. The next table shows a basic summary of TotalIncome.

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
1442    4166    5314    6782    7308   81000
```

Figure 13 shows the original distribution of TotalIncome with its log version. As we can see, after log transformation the distribution looks much closer to normal and the effect of extreme values has been significantly subsided.

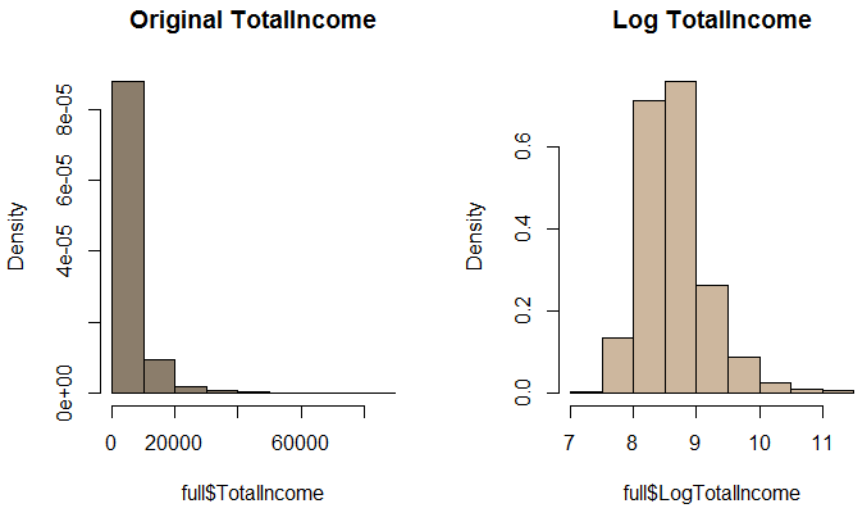


Figure 13: Comparison of the original distribution of TotalIncome with its log version.

Variable Loan_by_TotalIncome

The next table shows a basic summary of Loan_by_TotalIncome. As we can see, this variable looks symmetric, so we will not take the log transformation.

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.001905 0.019231 0.024131 0.024076 0.028432 0.102273
```

Variable FamilyIncome

The next table shows a basic summary of FamilyIncome.

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
457.5  1763.2  2666.7  3582.7  4111.0 63337.0
```

Figure 14 shows the original distribution of FamilyIncome with its log version. We observe that after log transformation, the distribution looks much closer to normal and the effect of extreme values has been significantly subsided.

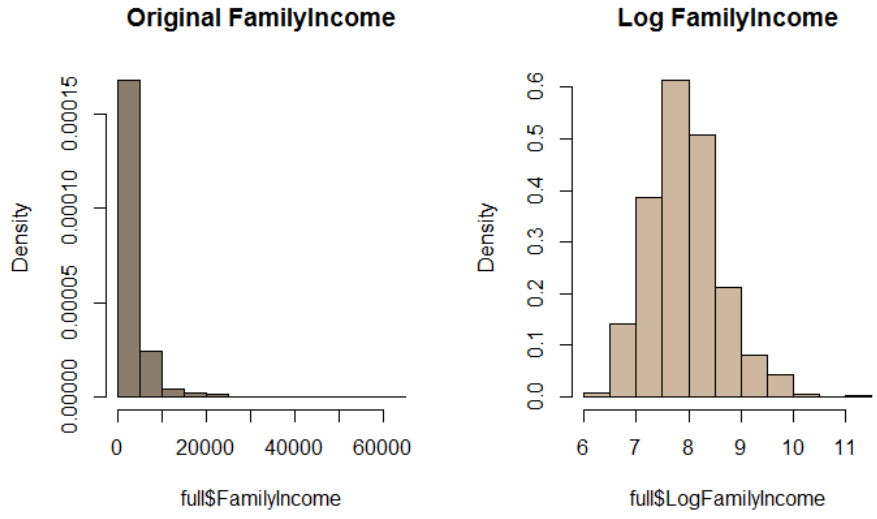


Figure 14: Comparison of the original distribution of FamilyIncome with its log version.

Variable Loan_by_FamilyIncome

The next table shows a basic summary of Loan_by_FamilyIncome.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00349 0.02936 0.04806 0.05634 0.07463 0.33085
```

Figure 15 shows the original distribution of Loan_by_FamilyIncome with its log version. As we can see, after log transformation, the distribution looks much closer to normal and the effect of extreme values has been significantly subsided.

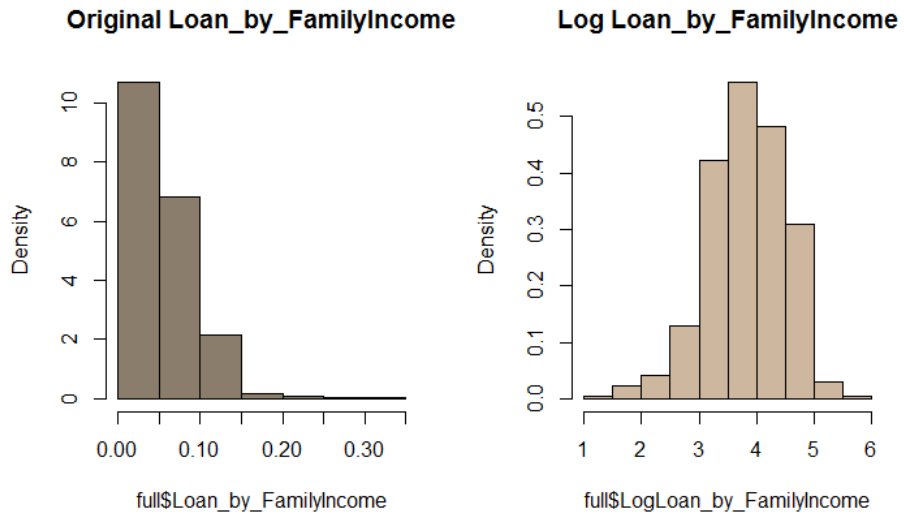


Figure 15: Comparison of the original distribution of Loan_by_FamilyIncome with its log version.

Variable LoanPerMonth

The next table shows a basic summary of LoanPerMonth.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0250  0.2889  0.3611  0.4903  0.5000 21.6667
```

Figure 16 shows the original distribution of LoanPerMonth with its log version. We observe that after log transformation, the distribution looks much closer to normal and the effect of extreme values has been significantly subsided.

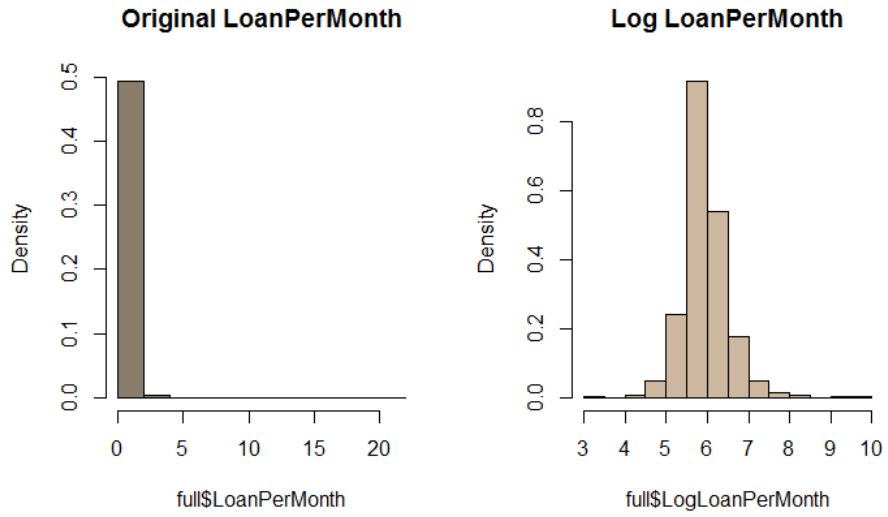


Figure 16: Comparison of the original distribution of LoanPerMonth with its log version.

Since log transformations of LoanAmount, TotalIncome, FamilyIncome, Loan_by_FamilyIncome, LoanPerMonth make the effect of extreme values less intense we will keep these variables and remove their corresponding original values from the database. We will also remove the ApplicantIncome and CoapplicantIncome since the new variable LogTotalIncome is a better predictor. Finally, we will remove the variable FamilySize, since we will keep its discretized version, DFamilySize.

Correlation of Continuous Variables

We will also examine the continuous variables separately, and remove any that are highly correlated. After computing their correlation, we see that there aren't any variables whose correlation is higher than 0.8, so we don't have to remove any variables for this reason.

Predictor Importance

Finally, we will compute again the predictor importance to evaluate the relative importance of all predictor variables. For categorical variables, the predictor importance is calculated as 1-Pval from Pearson Chi-square test and for continuous variables it is calculated as 1-Pval from ANOVA F-Test for Equity of Mean. The next table shows the predictor importance of the variables in descending (highest to lowest) order.

```
Credit_History          1
Property_Area           0.997864
Married                 0.9602594
Education               0.9569004
Loan_by_TotalIncome     0.956683
Coapplicant             0.9240891
Loan_Amount_Term        0.8694148
Dependents              0.7148474
DFamilySize             0.651103
LogLoanAmount           0.6320589
LogFamilyIncome         0.5625994
LogLoanPerMonth         0.5038477
LogTotalIncome          0.1420932
LogLoan_by_FamilyIncome 0.09906405
Gender                  0.04015181
Self_Employed           2.553513e-15
```

From the above univariate analysis, we see that Credit_History, Property_Area, Married and Education and the new features Loan_by_TotalIncome and Coapplicant are the most significant predictors.

Prediction

In the begging of data processing we merged training and testing sets. Therefore, our first step now is to split the data back into the training and test sets.

Model Building

Now, we will build and fit some models to the training set and we will compute their accuracy on the test set. For this purpose, we will build seven different models: Decision Tree (CART), Random Forest (RF), Forest of conditional inference trees, Generalized Linear Model (GLM), Gradient Boosting Machine (GBM), Support Vector Machine (SVM) with Radial Basis Function Kernel and kNN (k-Nearest Neighbors). We reset the random number seed before each run to ensure that each algorithm will be evaluated using the same data partitions. This means that the results will be directly comparable.

Decision Trees

Decision Trees classify observations by sorting them down the tree from the root node to some leaf node which provides the classification of the observation. Each node in the tree specifies a test on a particular attribute (explanatory variable) of the observation, and each branch descending from that node corresponds to one of the possible values for that test [1]. Notice that decision trees are prone to overfitting which means that we should be very careful with how deep we grow them.

In this project, we will use the rpart (which stands for recursive partitioning) algorithm to build the tree. Also, note that all models will be tuned to find the optimal parameter values, i.e., minsplit (minimum number of observations in a node) and cp (complexity parameter).

First, we will build a decision tree to predict Loan_Status using only the variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome and Coapplicant. The next figure illustrates this decision tree.

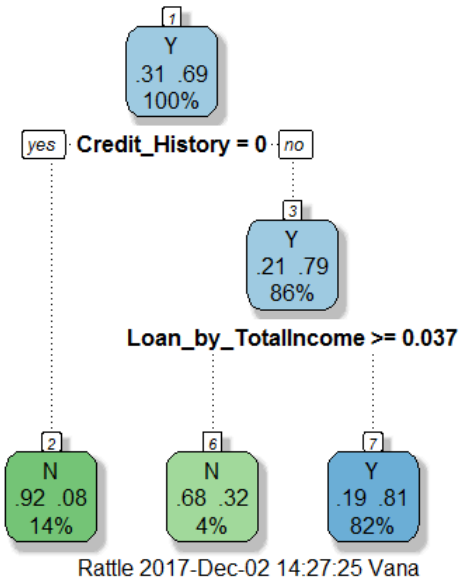


Figure 17: Visualization of the decision tree.

Credit_History governs the first decision due to the greedy nature of decision trees. From the top we can see that the node is voting Y, so at this level everyone would take the loan. Below we see that 31% of applicants don't take the loan, while 69% of take the loan (the most will take the loan here that's why the node is voting that everyone takes the loan). If we go down to the credit history variable, we see that 92% of those whose credit history doesn't meet the guidelines will not take the loan, while only 8% of them will take it. On the other hand, 79% of applicants whose credit history meets the guidelines will take the loan, while only 21% of them will not take it. Then, decisions for applicants whose credit history meets the guidelines have been made on the Loan_by_TotalIncome variable. 68% of applicants whose Loan_by_TotalIncome is higher than 0.037 will not take the loan, while only 32% of them will take it. Finally, 81% of applicants whose Loan_by_TotalIncome is not higher than 0.037 will take the loan, while only 19% of them will not take it.

We can now evaluate this model by computing its accuracy on the training set and the test set (by submitting a .csv file to Analytics Vidhya). The accuracy on the training set is 0.82247 and on the test set 0.78472.

Using different variables as predictors we can build new decision trees. Table 2 provides a summary of the model evaluation results. The best prediction for loan approval with decision trees was achieved using the predictor variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome, Coapplicant, Loan_Amount_Term, Dependents, DFamilySize, LogLoanAmount, LogFamilyIncome and LogLoanPerMonth. The accuracy of this model on the training and test set is 0.83061 and 0.79167, respectively. Figure 18 visualizes the decision tree that was built with these predictors. We observe that Credit_History and some of the new variables are governing the tree.

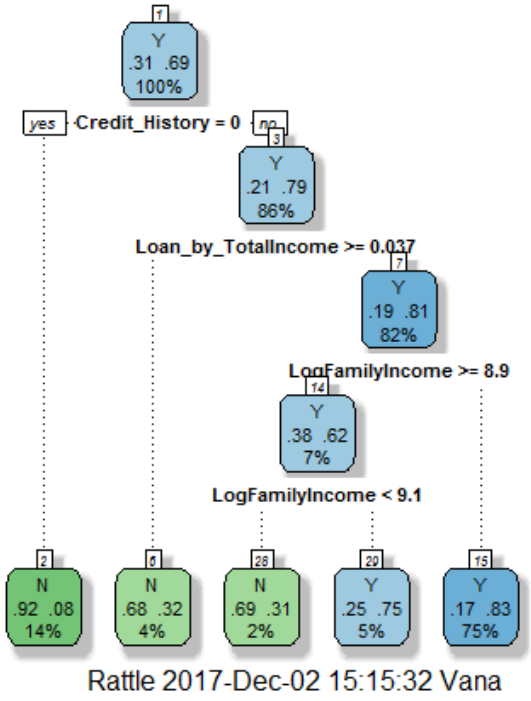


Figure 18: Visualization of the decision tree that achieved the highest accuracy on the test set.

Formula	Accuracy (Training Set)	Accuracy (Test Set) – Analytics Vidhya Score
1. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant	0.82247	0.78472
2. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant + Loan_Amount_Term + Dependents + DFamilySize + LogLoanAmount + LogFamilyIncome + LogLoanPerMonth	0.83061	0.79167
3. Loan_Status ~ Gender + Married + Dependents + Education + Self_Employed + Loan_Amount_Term + Credit_History + Property_Area + Loan_by_TotalIncome + Coapplicant + DFamilySize + LogLoanAmount + LogTotalIncome + LogFamilyIncome + LogLoan_by_FamilyIncome + LogLoanPerMonth	0.82410	0.75694

Table 2: Performance summary of decision tree models.

Random Forest

Random Forest (RF) is an algorithm that fits many classification (or regression) tree models to random subsets of the input data and averages the predictions from such simple trees for prediction [2]. Random forests correct for decision trees' habit of overfitting to their training set [3].

Also, note that all models will be tuned to find the optimal parameter values, i.e., mtry (number of variables used at each split of the tree) and ntree (number of trees).

First, we will build a random forest model to predict Loan_Status using only the variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome and Coapplicant. It is well-known that random forests do not waste the out-of-bag (OOB) observations, since they use them to examine how well each tree performs on unseen data. The mean decrease in Gini measures how pure the nodes are at the end of the tree. It actually tests how worse the model would perform if each variable was taken out, and a high score means that the variable was important. The next figure shows the relative variable importance by plotting the mean decrease in Gini calculated across all trees. In this model, we see that the most important predictor variable is Credit_History. Another variable of high importance is our variable Loan_by_TotalIncome.

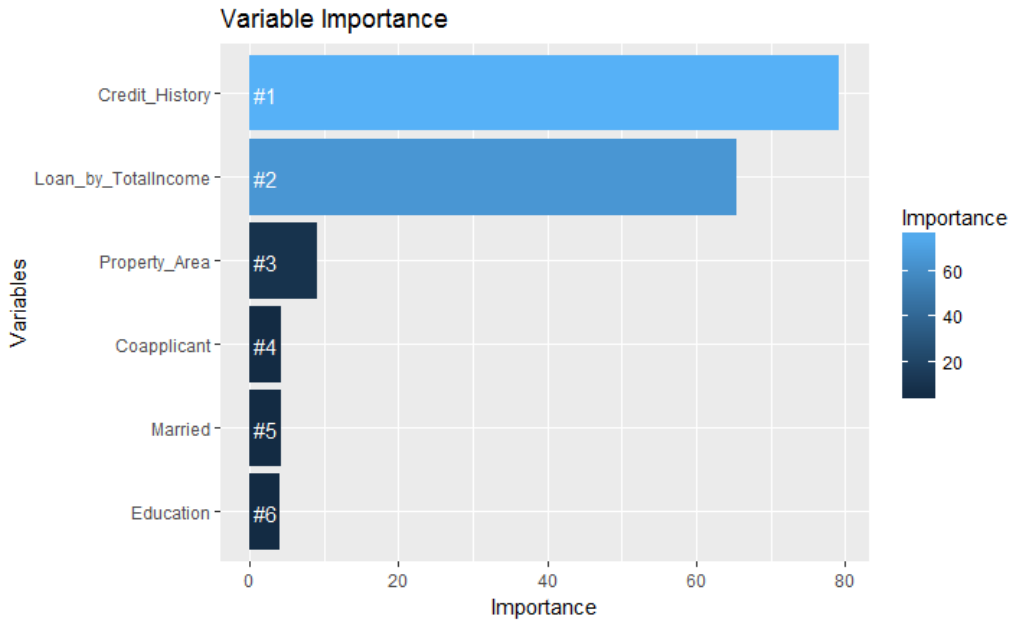


Figure 19: Variable importance for the developed random forest model.

We can now evaluate this model by computing its accuracy on the training set and the test set (by submitting a .csv file to Analytics Vidhya). The accuracy on the training set is 0.85016 and on the test set 0.79167.

Formula	Accuracy (Training Set)	Accuracy (Test Set) – Analytics Vidhya Score
1. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant	0.85016	0.79167
2. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant + Loan_Amount_Term + Dependents + DFamilySize + LogLoanAmount + LogFamilyIncome + LogLoanPerMonth	0.85667	0.78472
3. Loan_Status ~ Gender + Married + Dependents + Education + Self_Employed + Loan_Amount_Term + Credit_History + Property_Area + Loan_by_TotalIncome + Coapplicant + DFamilySize + LogLoanAmount + LogTotalIncome + LogFamilyIncome + LogLoan_by_FamilyIncome + LogLoanPerMonth	0.82410	0.75694

Table 3: Performance summary of random forest models.

Using different variables as predictors we can build new random forests models. Table 3 provides a summary of the model evaluation results. The best prediction for loan approval with random forests models was achieved using the predictor variables of either the first or the third model. Figure 20 visualizes the relative variable importance of the predictors used in the third model. We observe that although Credit_History is the most important variable, many of our created variables have significant importance.

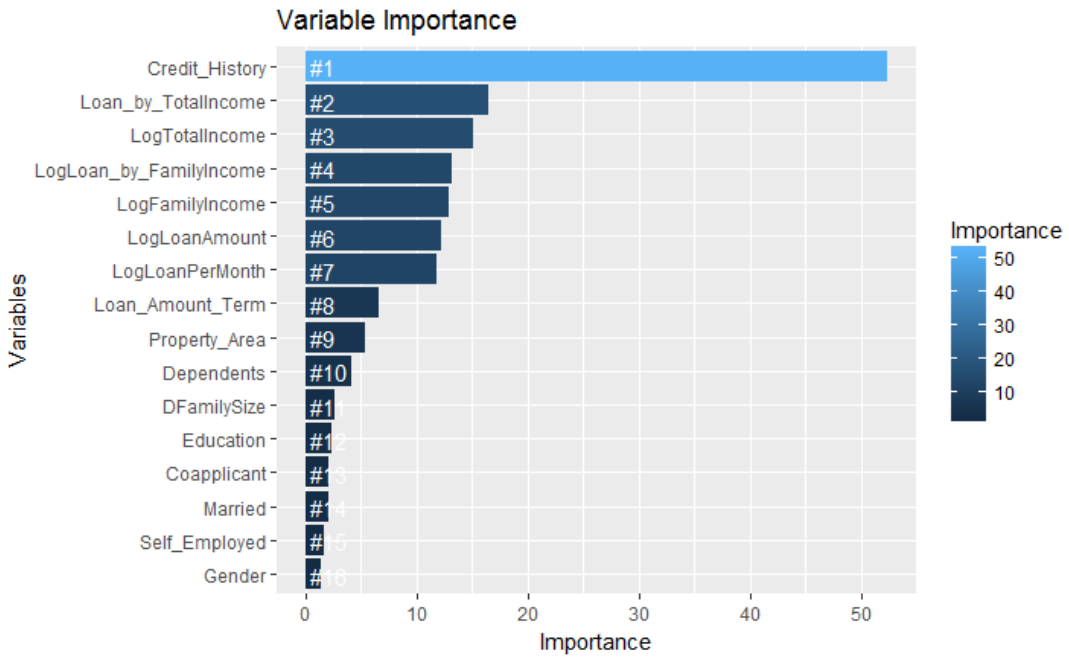


Figure 20: Variable importance for the third developed random forest model.

Forest of conditional inference trees

Conditional inference trees are a popular tree-based classification method. Similar to traditional decision trees, conditional inference trees also recursively partition the data by performing a univariate split on the dependent variable. The difference between conditional inference trees and traditional decision trees is that conditional inference trees adapt the significance test procedures to select variables rather than selecting variables by maximizing information measures [4].

Also, note that all models will be tuned to find the optimal parameter values, i.e., mtry (number of variables used at each split of the tree).

First, we will build a forest of conditional inference trees to predict Loan_Status using only the variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome and Coapplicant. We can now evaluate this model by computing its accuracy on the training set and the test set (by submitting a .csv file to Analytics Vidhya). The accuracy on the training set is 0.80944 and on the test set 0.77778.

Using different variables as predictors we can build new forests of conditional inference trees. Table 4 provides a summary of the model evaluation results. All models have the same performance on training and test sets.

Formula	Accuracy (Training Set)	Accuracy (Test Set) – Analytics Vidhya Score
1. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant	0.80944	0.77778
2. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant + Loan_Amount_Term + Dependents + DFamilySize + LogLoanAmount + LogFamilyIncome + LogLoanPerMonth	0.80944	0.77778
3. Loan_Status ~ Gender + Married + Dependents + Education + Self_Employed + Loan_Amount_Term + Credit_History + Property_Area + Loan_by_TotalIncome + Coapplicant + DFamilySize + LogLoanAmount + LogTotalIncome + LogFamilyIncome + LogLoan_by_FamilyIncome + LogLoanPerMonth	0.80944	0.77778

Table 4: Performance summary of forests of conditional inference trees.

Generalized Linear Models (GLMs)

Generalized Linear Models estimate regression models for outcomes following exponential distributions. In addition to the Gaussian (i.e. normal) distribution, these include Poisson, binomial and gamma distributions [5]. In this project we do a binomial regression (classification).

First, we will build a generalized linear model to predict Loan_Status using only the variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome and Coapplicant. We can now evaluate this model by computing its accuracy on the training set and the test set (by submitting a .csv file to Analytics Vidhya). The accuracy on the training set is 0.80944 and on the test set 0.77778.

Using different variables as predictors we can build new generalized linear models. Table 5 provides a summary of the model evaluation results. The best prediction for loan approval with generalized linear models was achieved using the predictor variables of the first model.

Formula	Accuracy (Training Set)	Accuracy (Test Set) – Analytics Vidhya Score
1. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant	0.80944	0.77778
2. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant + Loan_Amount_Term + Dependents + DFamilySize + LogLoanAmount + LogFamilyIncome + LogLoanPerMonth	0.81758	0.77083
3. Loan_Status ~ Gender + Married + Dependents + Education + Self_Employed + Loan_Amount_Term + Credit_History + Property_Area + Loan_by_TotalIncome + Coapplicant + DFamilySize + LogLoanAmount + LogTotalIncome + LogFamilyIncome + LogLoan_by_FamilyIncome + LogLoanPerMonth	0.81596	0.77083

Table 5: Performance summary of generalized linear models.

Gradient Boosting Machine (GBM)

The way the Gradient Boosting Machine (GBM) works is very similar to the Random Forest, since both approaches combine weaker models (typically decision trees) to predict the class. Their difference is that Random Forest trains the trees from different random subsets of the input data, while Gradient Boosting takes the error from the previous tree and uses it to improve the next one [6].

Also, note that all models will be tuned to find the optimal parameter values, i.e., n.trees (number of trees) and interaction.depth (maximum nodes per tree).

We will build a gradient boosting machine to predict Loan_Status using only the variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome and Coapplicant. Figure 21 visualizes the relative variable importance of the predictors. We see that the most important predictor variable is Credit_History. Another variable of high importance is our variable Loan_by_TotalIncome.

We can now evaluate this model by computing its accuracy on the training set and the test set (by submitting a .csv file to Analytics Vidhya). The accuracy on the training set is 0.70684 and on the test set 0.74306. Note that since this model performs relatively poor compared with the other models, we will not train other GBM models using different predictors.

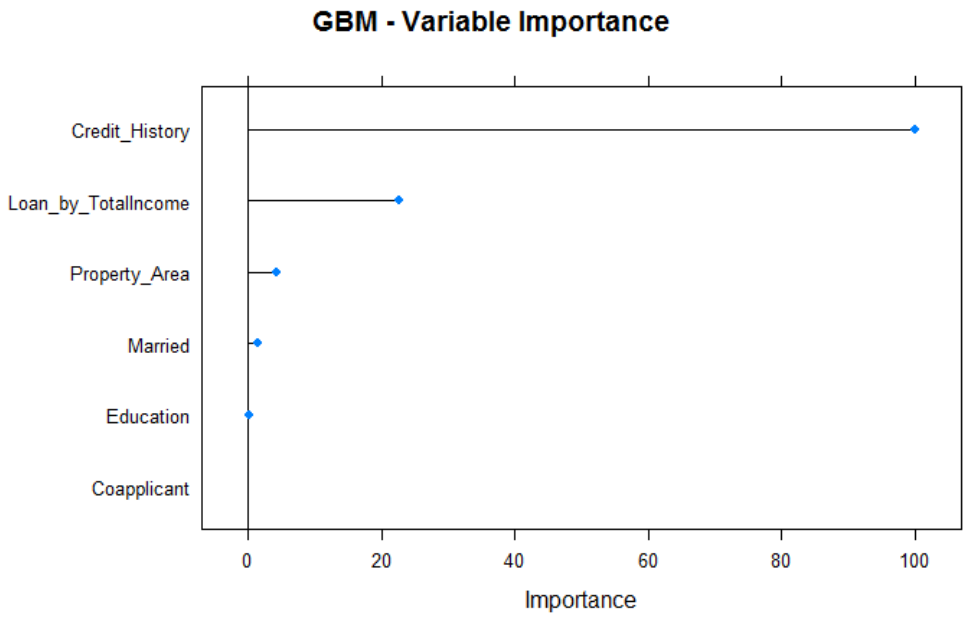


Figure 21: Variable importance for the gradient boosting machine.

Support Vector Machines (SVM)

We also consider Support Vector Machines (SVM) with Radial Basis Function (RBF) Kernel. Basically, the SVM classifier maps the input space into a new space by a kernel transformation, and then finds the optimal separating hyperplane and the margin of separations in that space. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new observations [7].

We are going to tune the SVM models to find the optimal parameter values, i.e., the gamma and cost values. The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close', while the cost parameter is a regularization term that controls the complexity of the model. A high cost value will force the SVM model to create a complex enough prediction function to missclassify as few training points as possible, while a lower cost parameter will lead to a simpler prediction function [8].

First, we will build a SVM model to predict Loan_Status using only the variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome and Coapplicant. We can now evaluate this model by computing its accuracy on the training set and the test set (by submitting a .csv file to Analytics Vidhya). The accuracy on the training set is 0.85179 and on the test set 0.79861.

Using different variables as predictors we can build new SVM models. Table 6 provides a summary of the model evaluation results. The best prediction for loan approval with SVM models was achieved using the predictor variables of the first model.

Formula	Accuracy (Training Set)	Accuracy (Test Set) – Analytics Vidhya Score
1. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant	0.85179	0.79861
2. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant + Loan_Amount_Term + Dependents + DFamilySize + LogLoanAmount + LogFamilyIncome + LogLoanPerMonth	0.82410	0.78472
3. Loan_Status ~ Gender + Married + Dependents + Education + Self_Employed + Loan_Amount_Term + Credit_History + Property_Area + Loan_by_TotalIncome + Coapplicant + DFamilySize + LogLoanAmount + LogTotalIncome + LogFamilyIncome + LogLoan_by_FamilyIncome + LogLoanPerMonth	0.81107	0.77778

Table 6: Performance summary of SVM models.

k-Nearest Neighbor (kNN)

In the kth-Nearest Neighbor (kNN) algorithm, an observation is classified by a majority vote of its k closest points (neighbors), determined by a predefined distance function. The observation is then assigned to the class most common amongst its k nearest neighbors. The number “k” is the nearest neighbors we wish to take vote from and is typically a small positive integer [9].

The kNN classifier requires all variables to be numeric. In order to be able to use kNN with categorical variables, we will use the knncat algorithm. In this algorithm, continuous variables are permitted too.

First, we will build a kNN model to predict Loan_Status using only the variables Credit_History, Property_Area, Married, Education, Loan_by_TotalIncome and Coapplicant. We can now evaluate this model by computing its accuracy on the training set and the test set (by submitting a .csv file to Analytics Vidhya). The accuracy on the training set is 0.80618 and on the test set 0.79167.

Using different variables as predictors we can build new kNN models. Table 7 provides a summary of the model evaluation results. The best prediction for loan approval with kNN models was achieved using the predictor variables of the first model.

Formula	Accuracy (Training Set)	Accuracy (Test Set) – Analytics Vidhya Score
1. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant	0.80618	0.79167
2. Loan_Status ~ Credit_History + Property_Area + Married + Education + Loan_by_TotalIncome + Coapplicant + Loan_Amount_Term + Dependents + DFamilySize + LogLoanAmount + LogFamilyIncome + LogLoanPerMonth	0.80944	0.77778
3. Loan_Status ~ Gender + Married + Dependents + Education + Self_Employed + Loan_Amount_Term + Credit_History + Property_Area + Loan_by_TotalIncome + Coapplicant + DFamilySize + LogLoanAmount + LogTotalIncome + LogFamilyIncome + LogLoan_by_FamilyIncome + LogLoanPerMonth	0.81107	0.77778

Table 7: Performance summary of kNN models.

References

[1] T. Mitchell, *Machine Learning*, McGraw Hill, 1997

[2] L. Breiman, “Random forests”, Machine Learning, Vol. 45, No. 1, pp 5–32, 2001

[3] “Random forest”, [Online]. Available: https://en.wikipedia.org/wiki/Random_forest, [Accessed: 02- Dec- 2017]

[4] W. Yu and D. Chiu, Machine Learning with R Cookbook, 1st Edition, Chapter 5. Classification (I) – Tree, Lazy, and Probabilistic, p 166, Packt Publishing, 2015

[5] “Generalized Linear Model (GLM)”, [Online]. Available: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glm.html, [Accessed: 02- Dec- 2017]

[6] T. Drabas, Practical Data Analysis Cookbook, Chapter 3. *Classification Techniques*, p 91, Packt Publishing, 2011

[7] S. Wan and M.W. Mak, *Machine Learning for Protein Subcellular Localization Prediction*, p 167, De Gruyter, 2015

[8] A. Karatzoglou, D. Meyer and K. Hornik, Support Vector Machines in R, Journal of Statistical Software, Vol. 15, No. 9, 2006

[9] M.H. Namaki, K. Sasani, Y. Wu and A.H. Gebremedhin, Performance Prediction for Graph Queries, In SIGMOD, 2017