STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Give Me Some Credit

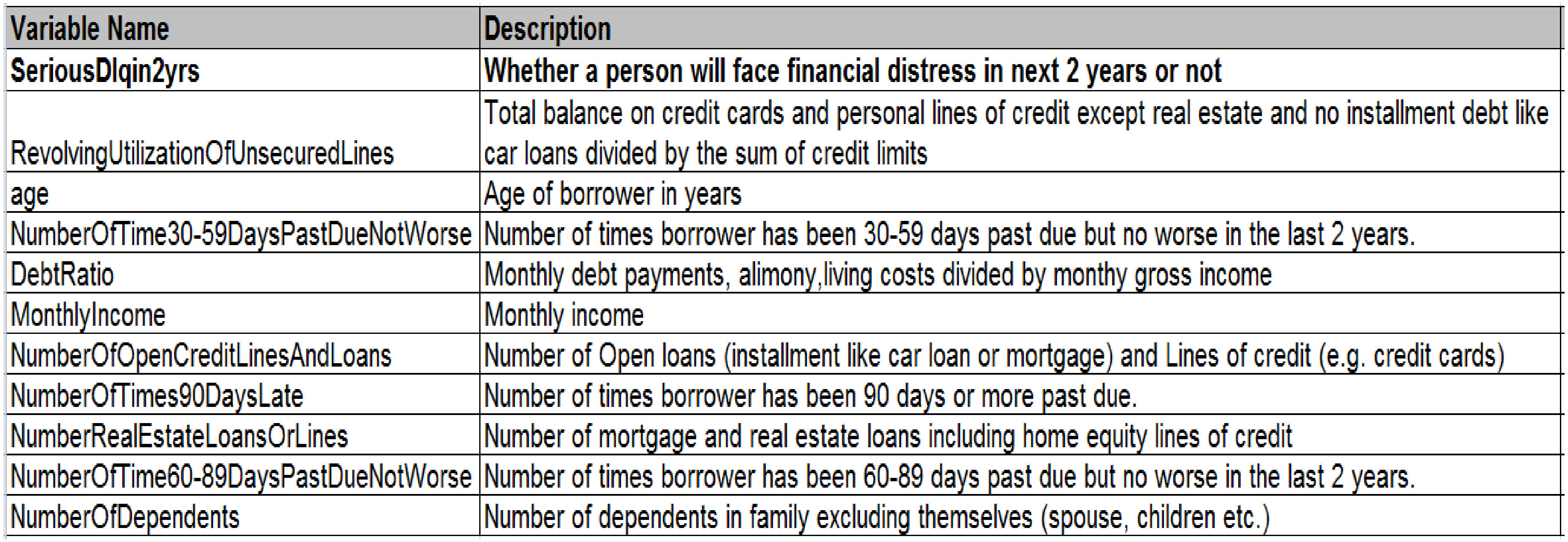
Classify whether or not somebody will experience financial distress in the next two years

Team:
Akash Chawla
Ana Parra Vera
Soham Mukherjee
Sweta Chowdary

Project Overview

- Based on an individual's credit data, classify whether or not somebody will experience financial distress in the next two years
- Credit scoring algorithms make a guess at the chances of default
- The purpose of building our model is to make it accessible to the borrowers
- Banks play a crucial role in market economies
- Data Source: https://www.kaggle.com/c/GiveMeSomeCredit/data

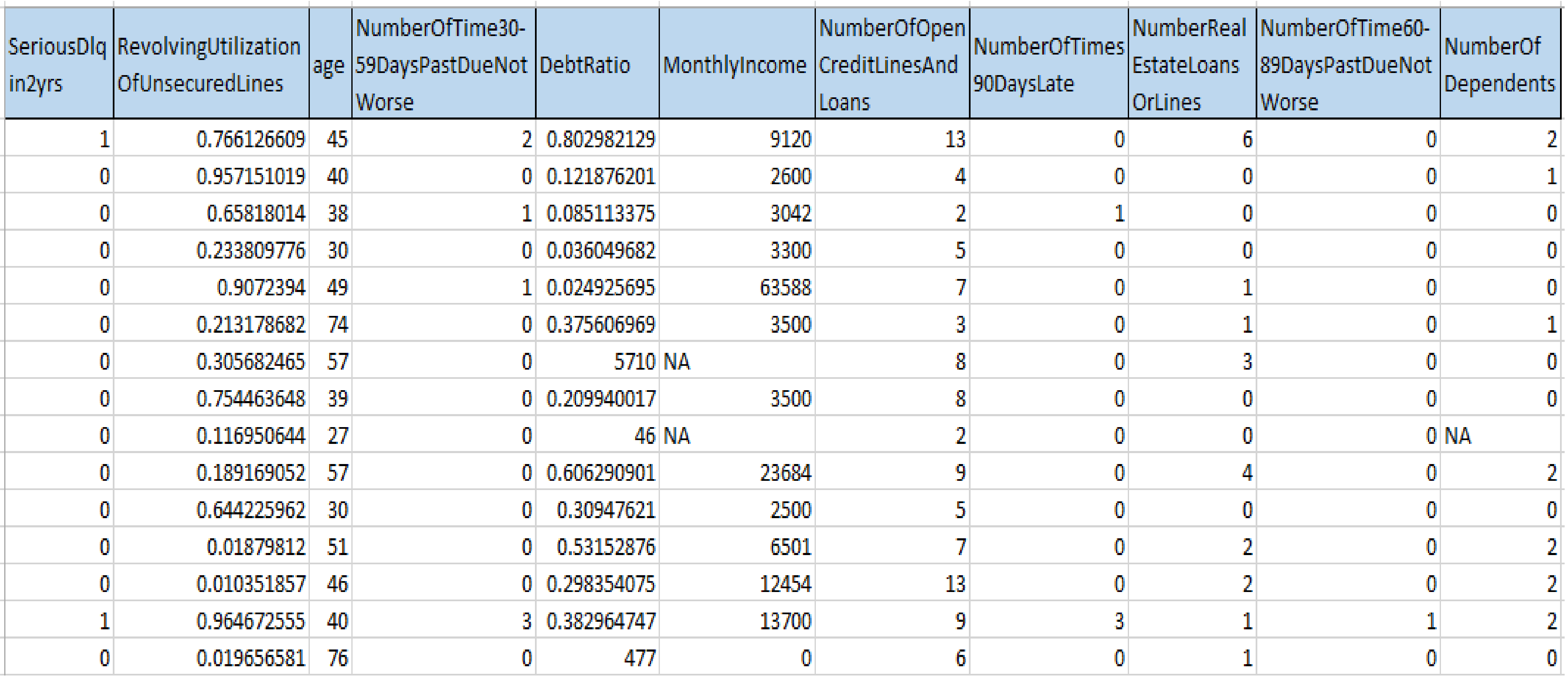
Data Fields



Variable Name	Description
SeriousDlqin2yrs	**Whether a person will face financial distress in next 2 years or not**
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits
age	Age of borrower in years
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.
DebtRatio	Monthly debt payments, alimony,living costs divided by monthy gross income
MonthlyIncome	Monthly income
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)

Snapshot of Data

150,000 records in dataset

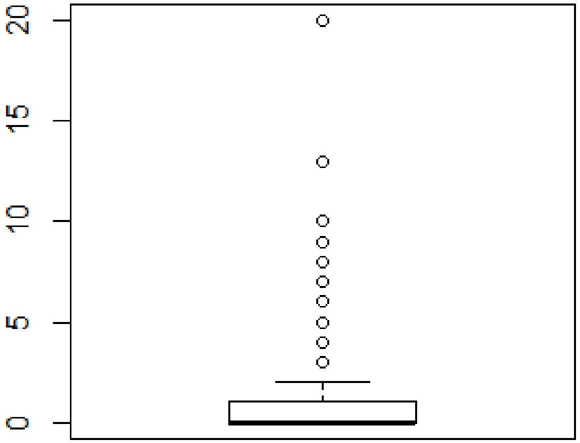


SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTime30-59DaysPastDueNotWorse	DebtRatio	MonthlyIncome	NumberOfOpenCreditLinesAndLoans	NumberOfTimes90DaysLate	NumberRealEstateLoansOrLines	NumberOfTime60-89DaysPastDueNotWorse	NumberOfDependents
1	0.766126609	45	2	0.802982129	9120	13	0	6	0	2
0	0.957151019	40	0	0.121876201	2600	4	0	0	0	1
0	0.65818014	38	1	0.085113375	3042	2	1	0	0	0
0	0.233809776	30	0	0.036049682	3300	5	0	0	0	0
0	0.9072394	49	1	0.024925695	63588	7	0	1	0	0
0	0.213178682	74	0	0.375606969	3500	3	0	1	0	1
0	0.305682465	57	0	5710	NA	8	0	3	0	0
0	0.754463648	39	0	0.209940017	3500	8	0	0	0	0
0	0.116950644	27	0	46	NA	2	0	0	0	NA
0	0.189169052	57	0	0.606290901	23684	9	0	4	0	2
0	0.644225962	30	0	0.30947621	2500	5	0	0	0	0
0	0.01879812	51	0	0.53152876	6501	7	0	2	0	2
0	0.010351857	46	0	0.298354075	12454	13	0	2	0	2
1	0.964672555	40	3	0.382964747	13700	9	3	1	1	2
0	0.019656581	76	0	477	0	6	0	1	0	0

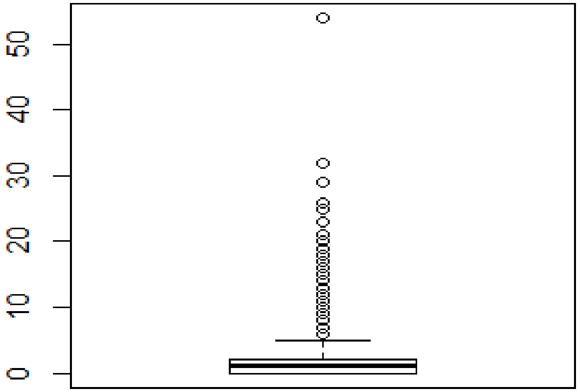
Data Exploration

Summary and Boxplot of Data Columns

```
summary(train$NumberOfDependents)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
0.000   0.000   0.000   0.757   1.000  20.000    3924 
```

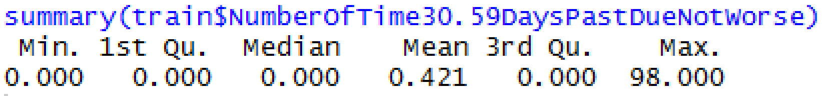


```
summary(train$NumberRealEstateLoansOrLines)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.000   0.000   1.000   1.018   2.000  54.000 
```

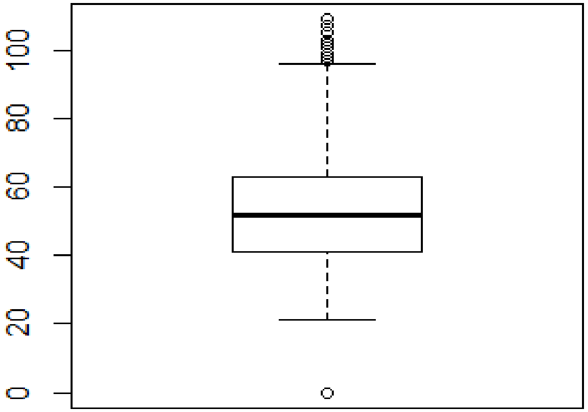


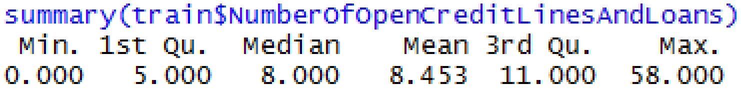
Data Exploration

Summary and Boxplot of Data Columns

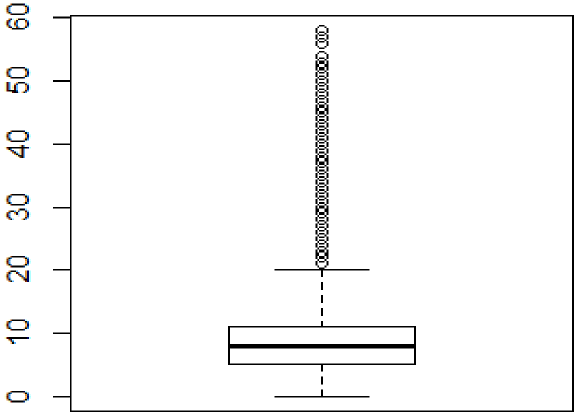


```
summary(train$NumberOfTime30.59DaysPastDueNotWorse)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.000   0.000   0.000   0.421   0.000  98.000
```





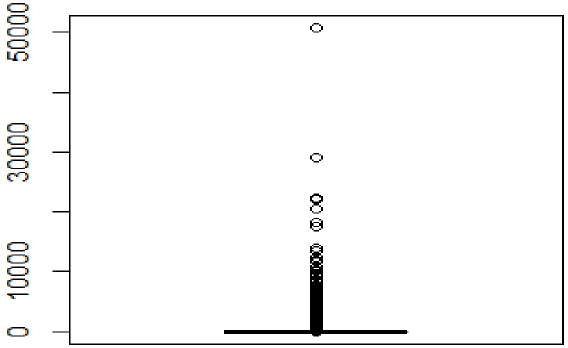
```
summary(train$NumberOfOpenCreditLinesAndLoans)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.000   5.000   8.000   8.453  11.000  58.000
```



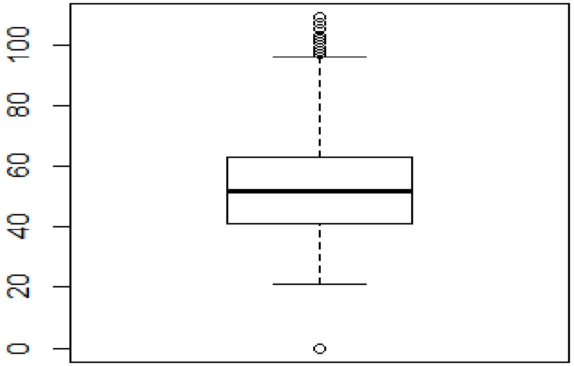
Data Exploration

Summary and Boxplot of Data Columns

```
summary(train$RevolvingUtilizationOfUnsecuredLines)
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
   0.00     0.03     0.15     6.05     0.56 50710.00 
```

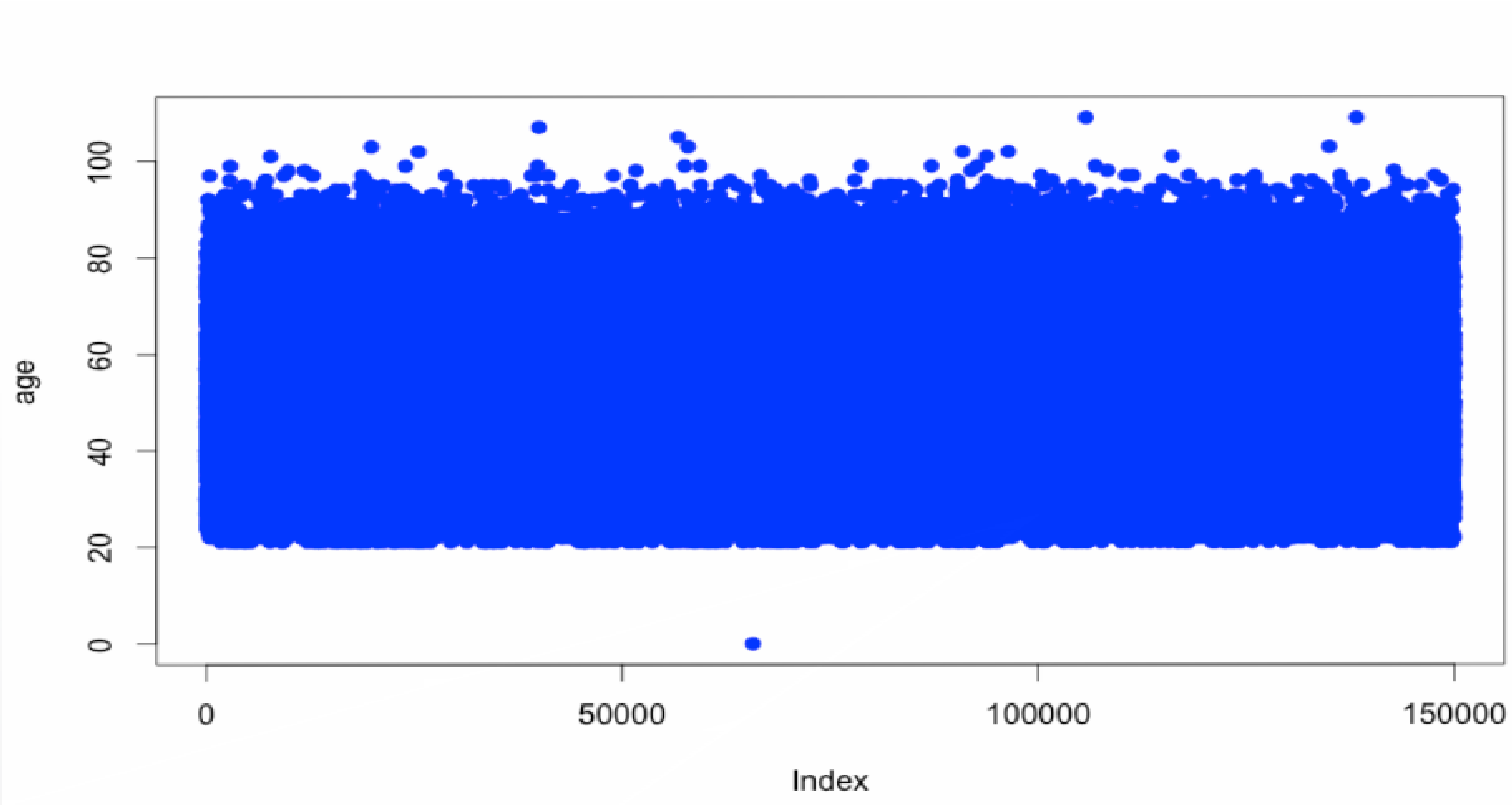


```
summary(train$age)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.0    41.0    52.0    52.3    63.0   109.0 
```



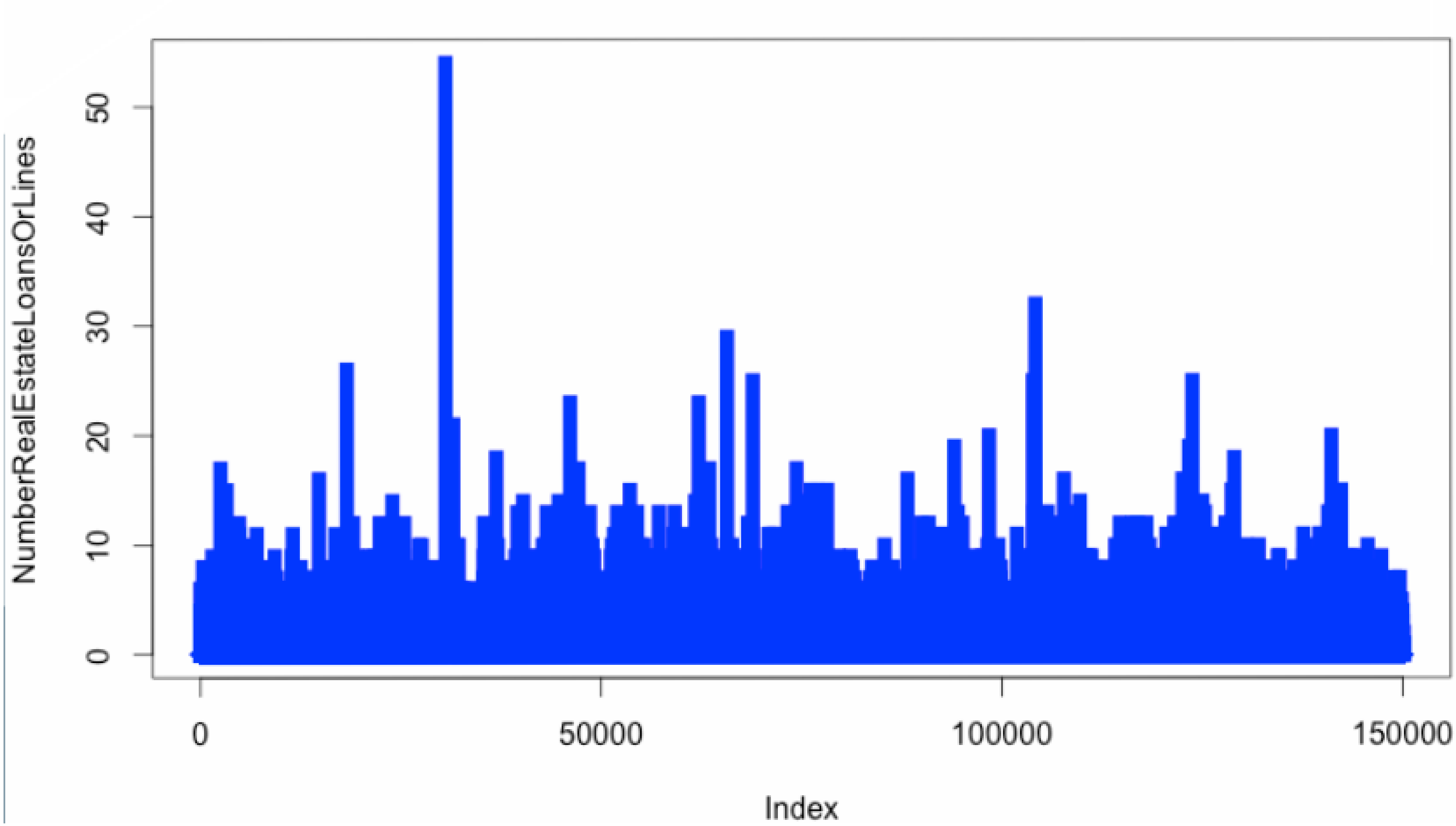
Data Exploration

Scatter Plot of Age



Data Exploration

Histogram of Number Real Estate Loans Or Lines



Outliers

Finding Outliers Numerically

- The Interquartile range (IQR) is a measure of variability that represents the spread of the middle 50% of the data

IQR= Q3 - Q1

- A data value is an outlier if:

It is located 1.5 (IQR) or more below Q1, or

It is located 1.5 (IQR) or more above Q3

Outliers

- Age =0 (1 record) → Replaced with Median
- Revolving Utilization >3 (292 records) → Deleted
- Monthly Income=0 and Debt Ratio =0 (97 Records) → Deleted
- Debt Ratio = 0 (4016 records) → Deleted
- Number of Real Estate Loans =54 (1 record) → Replaced with Median

- NumberOfTime60.89DaysPastDue, NumberOfTimes90DaysLate, NumberOfTime30.59DaysPastDue =98 (269 rows) → No Action

Missing Data

- 29,731 Records with Monthly Income missing
- 3,924 Records with Number of Dependents missing

Handling Missing Data

- Delete rows with missing values
- Predict missing values using KNN
- Replace missing values by Mean or Median

- For Number of Dependents, we replaced the Missing values with Median

Prediction of Missing Values

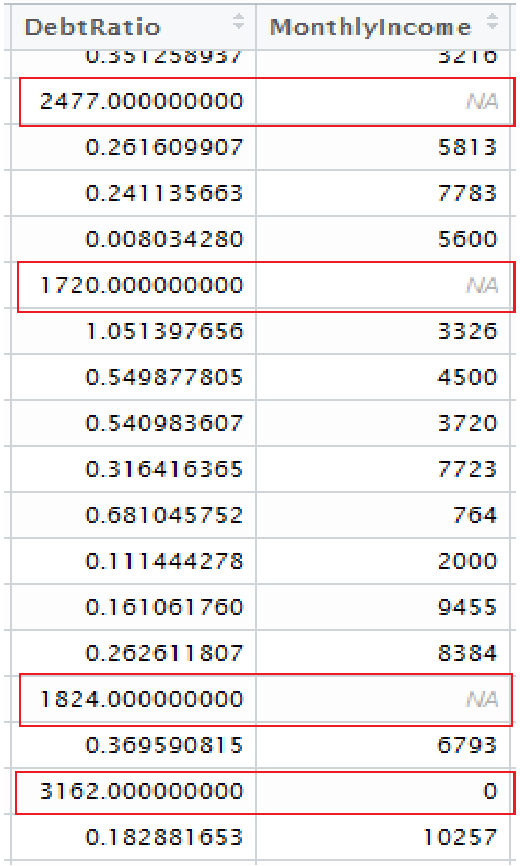
- Implemented KNN to predict Monthly Income
    - ➢ Using K=5, Error Rate= 73.09%
    - ➢ Using K=10, Error Rate= 71.43%
    - ➢ Using K=20, Error Rate= 69.59%
    - ➢ Using K=30, Error Rate= 67.63%
    - ➢ Using K=50, Error Rate= 65.27%
    - ➢ Using K=100, Error Rate= 64.30%

- Replacing Missing values with **Median** resulted in 72.95% error rate
- Replacing Missing values with **Mean** resulted in 94.70% error rate

Addition of a new field

Expenditure = Debt Ratio * Monthly Income

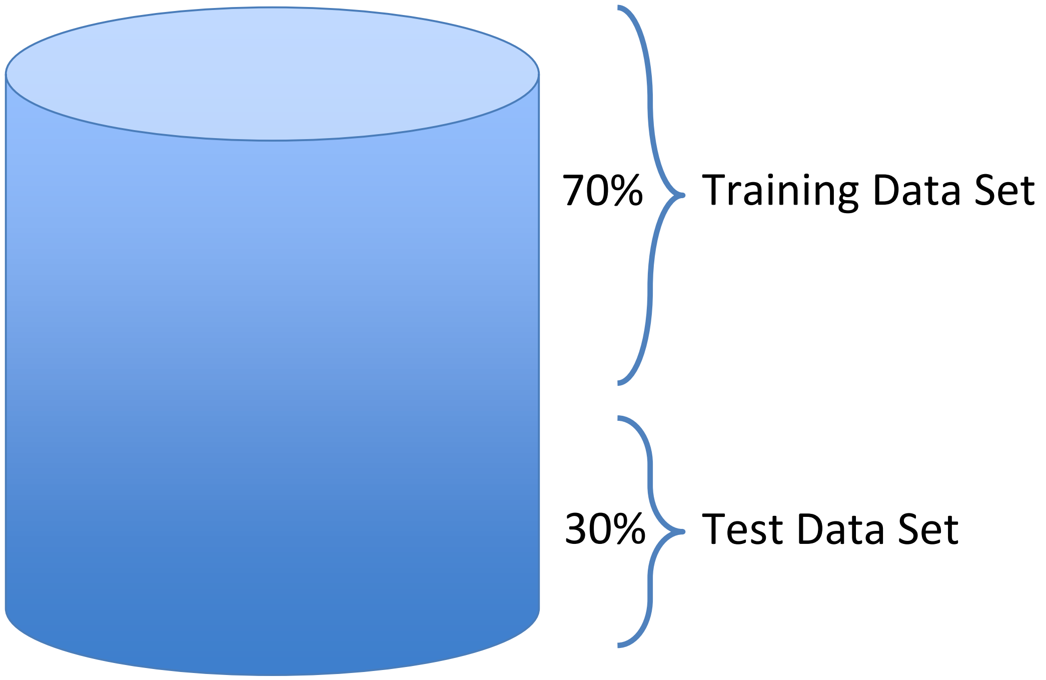
- Not a good idea to predict or even replace monthly income with median
- Large values of Debt Ratio where Monthly Income is not available
- To implement the above formula correctly, replace 'NA' and '0' values for Monthly Income with '1'
- Removed Debt Ratio and Monthly Income columns from our dataset
- Added Expenditure column to our dataset



DebtRatio	MonthlyIncome
0.351258937	[illegible]
2477.000000000	NA
0.261609907	5813
0.241135663	7783
0.008034280	5600
1720.000000000	NA
1.051397656	3326
0.549877805	4500
0.540983607	3720
0.316416365	7723
0.681045752	764
0.111444278	2000
0.161061760	9455
0.262611807	8384
1824.000000000	NA
0.369590815	6793
3162.000000000	0
0.182881653	10257

Split Data into Train and Test sets

- Uniform Division of Data



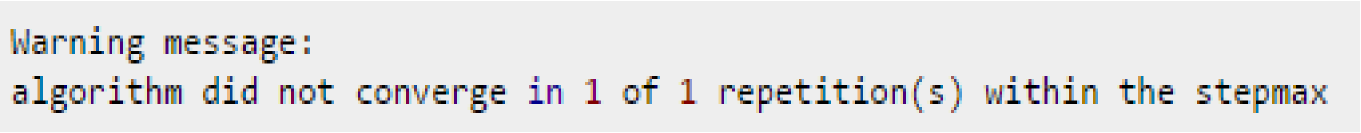
70% Training Data Set

30% Test Data Set

Classification using ANN

- We tried to implement the Artificial Neural Network.
- Runtime: 15-18 hours for 101916 records

Error:



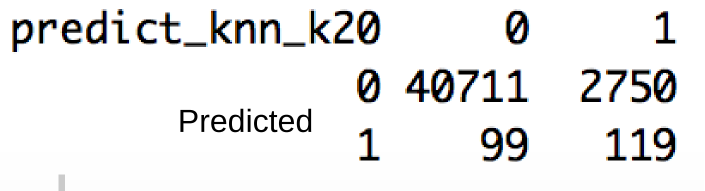
```
Warning message:
algorithm did not converge in 1 of 1 repetition(s) within the stepmax
```

```
plot(net)
Error in plot.nn(net) : weights were not calculated
```

Classification using KNN

- ➢ K=20

Actual



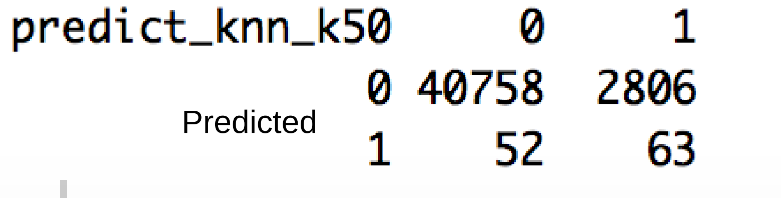
```
predict_knn_k20     0     1
              0 40711  2750
              1    99   119
```

Predicted

- Error Rate: 6.52%

- ➢ K=50

Actual



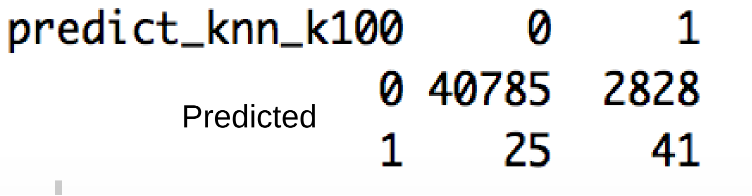
```
predict_knn_k50     0     1
              0 40758  2806
              1    52    63
```

Predicted

- Error Rate: 6.54%

- ➢ K=100

Actual

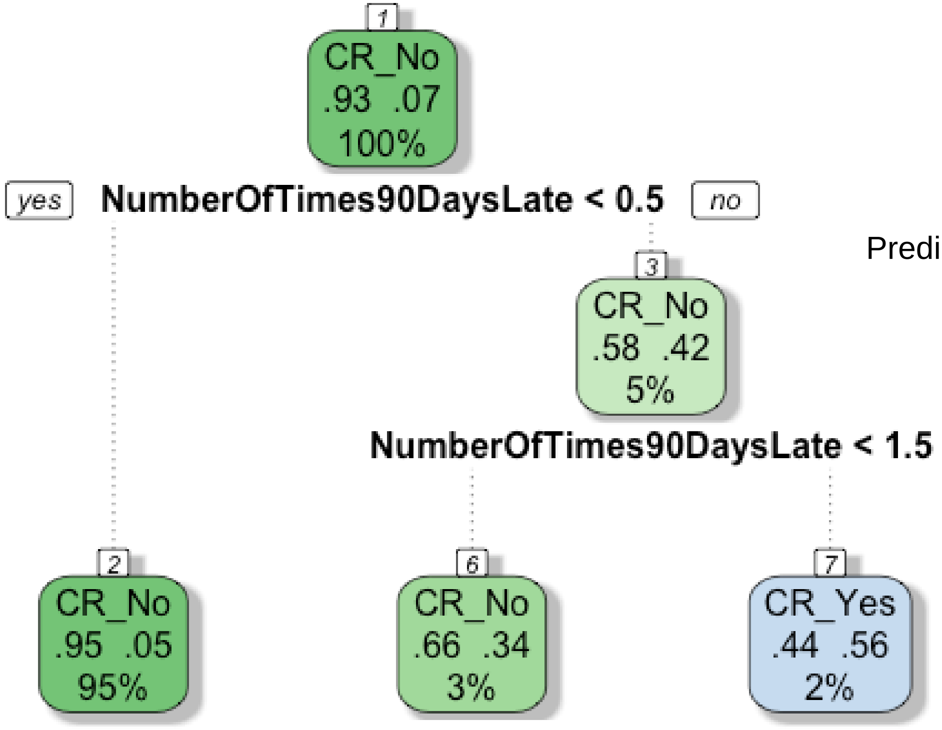


```
predict_knn_k100     0     1
               0 40785  2828
               1    25    41
```

Predicted

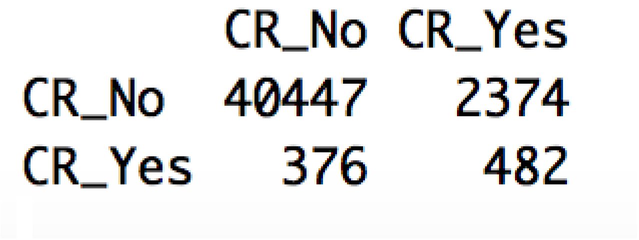
- Error Rate: 6.53%

Classification using CART



Actual

Predicted

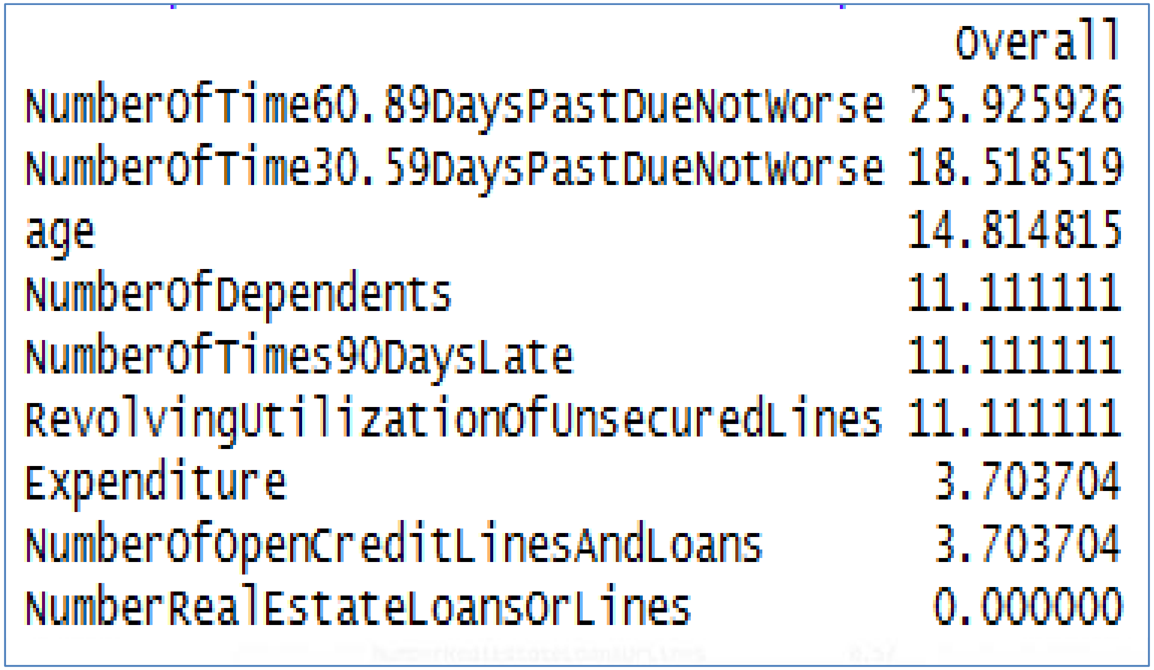


```
        CR_No CR_Yes
CR_No   40447   2374
CR_Yes    376    482
```

- Error Rate: 6.29%

Classification using C5.0

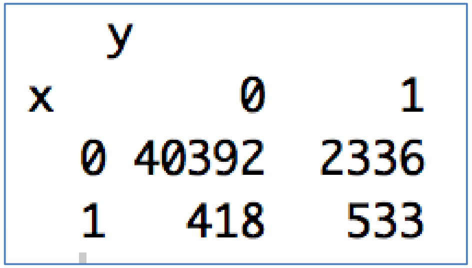
- Importance of Variables



```
                                        Overall
NumberOfTime60.89DaysPastDueNotWorse 25.925926
NumberOfTime30.59DaysPastDueNotWorse 18.518519
age                                  14.814815
NumberOfDependents                   11.111111
NumberOfTimes90DaysLate              11.111111
RevolvingUtilizationOfUnsecuredLines 11.111111
Expenditure                           3.703704
NumberOfOpenCreditLinesAndLoans       3.703704
NumberRealEstateLoansOrLines          0.000000
```

Actual

Predicted



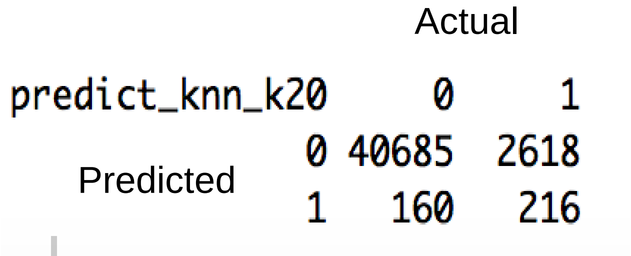
```
   y
x       0     1
  0 40392  2336
  1   418   533
```

- Error Rate: 6.30%

Classification

After using the most significant variables in our model

- **KNN, k=20**



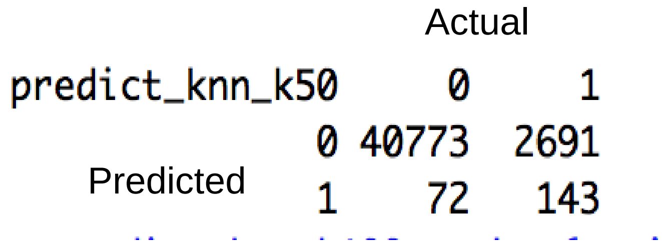
Actual

```
predict_knn_k20     0     1
              0 40685  2618
              1   160   216
```

Predicted

- Error Rate: 6.36%

- **KNN, k=50**



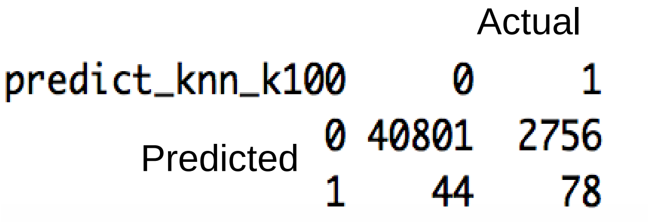
Actual

```
predict_knn_k50     0     1
              0 40773  2691
              1    72   143
```

Predicted

- Error Rate: 6.32%

- **KNN, k=100**



Actual

```
predict_knn_k100     0     1
               0 40801  2756
               1    44    78
```

Predicted

- Error Rate: 6.41%

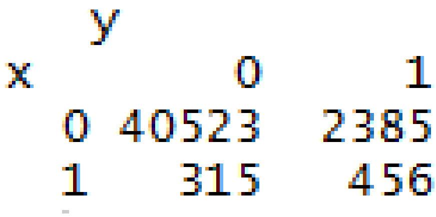
Classification

After using the most significant variables in our model

- **C5.0**

Actual

Predicted



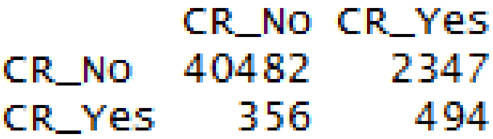
```
     y
x         0     1
  0 40523  2385
  1   315   456
```

- Error Rate: 6.18%

- **CART**

Actual

Predicted



```
        CR_No CR_Yes
CR_No   40482   2347
CR_Yes    356    494
```

- Error Rate: 6.19%

Conclusions

- Best method for our problem: C5.0

Least percent error

Easy to implement

No need to normalize

- Cleaning the data takes usually takes the most time (at least 60% of the time)

Our team spent at least 80% of the time cleaning the data

More effort than implementing the algorithms

- Outliers must be analyzed (not just deleted or replaced)

Outliers may be relevant to the dataset