# INFX 573 Problem Set 8 - Classification

*Rajendran Seetharaman*

*Due: Thursday, December 7, 2017*

## Introduction

### Collaborators:

### Instructions:

2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. List all collaborators on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps7.Rmd`, knit a PDF and submit the PDF file on Canvas.

### Data

You will be using credit card application data (on canvas). This originates from a confidential source, and all variable names are removed. The only variable you have to know is A16: approval (+) or refusal (-). The data is downloaded from UCI Machine Learning Repo, more information is in the meta file.

Ans. The dataset consists of credit card application data. The only variable which can be interpreted is A16, which represents wether a credit card application was approved or rejected.
The data is downloaded from UCI Machine Learning Repo

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
#read the data
credit_card_info <- read.csv('credit_card_applications.csv.bz2')
#replace + and - with approved and rejected in A16
credit_card_info <- credit_card_info %>%
```

```r
  mutate(A16 =as.factor(ifelse(A16=='+',"Approved","Rejected")))
#convert A2 to numeric values and replace ? with NA's
credit_card_info$A2[credit_card_info$A2=='?']<-NA
credit_card_info$A2 <-
  as.numeric(as.character(credit_card_info$A2))
credit_card_info$A2[is.na(credit_card_info$A2)] <-0
#Summarize data
str(credit_card_info)
```

```
## 'data.frame':    690 obs. of  16 variables:
##  $ A1 : Factor w/ 3 levels "?","a","b": 3 2 2 3 3 3 3 2 3 3 ...
##  $ A2 : num  30.8 58.7 24.5 27.8 20.2 ...
##  $ A3 : num  0 4.46 0.5 1.54 5.62 ...
##  $ A4 : Factor w/ 4 levels "?","l","u","y": 3 3 3 3 3 3 3 3 4 4 ...
##  $ A5 : Factor w/ 4 levels "?","g","gg","p": 2 2 2 2 2 2 2 2 4 4 ...
##  $ A6 : Factor w/ 15 levels "?","aa","c","cc",..: 14 12 12 14 14 11 13 4 10 14 ...
##  $ A7 : Factor w/ 10 levels "?","bb","dd",..: 9 5 5 9 9 9 5 9 5 9 ...
##  $ A8 : num  1.25 3.04 1.5 3.75 1.71 ...
##  $ A9 : Factor w/ 2 levels "f","t": 2 2 2 2 2 2 2 2 2 2 ...
##  $ A10: Factor w/ 2 levels "f","t": 2 2 1 2 1 1 1 1 1 1 ...
##  $ A11: int  1 6 0 5 0 0 0 0 0 0 ...
##  $ A12: Factor w/ 2 levels "f","t": 1 1 1 2 1 2 2 1 1 2 ...
##  $ A13: Factor w/ 3 levels "g","p","s": 1 1 1 1 3 1 1 1 1 1 ...
##  $ A14: Factor w/ 171 levels "?","00000","00017",..: 70 13 98 33 39 117 56 25 64 17 ...
##  $ A15: int  0 560 824 3 0 0 31285 1349 314 1442 ...
##  $ A16: Factor w/ 2 levels "Approved","Rejected": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
head(credit_card_info)
```

```
##   A1    A2    A3 A4 A5 A6 A7   A8 A9 A10 A11 A12 A13   A14 A15      A16
## 1  b 30.83 0.000  u  g  w  v 1.25  t   t   1   f   g 00202   0 Approved
## 2  a 58.67 4.460  u  g  q  h 3.04  t   t   6   f   g 00043 560 Approved
## 3  a 24.50 0.500  u  g  q  h 1.50  t   f   0   f   g 00280 824 Approved
## 4  b 27.83 1.540  u  g  w  v 3.75  t   t   5   t   g 00100   3 Approved
## 5  b 20.17 5.625  u  g  w  v 1.71  t   f   0   f   s 00120   0 Approved
## 6  b 32.08 4.000  u  g  m  v 2.50  t   f   0   t   g 00360   0 Approved
```

## Task

Your task is to predict the approval or disapproval using logistic regression and decision trees, and compare the performance of these methods.

### 1. Select variables

Select some variables. As we don't know the meaning of the variables, you have just to use cross-tables, scatter plots, trial-and-error to find good predictors of A16.

Ans. For categorical variables, I have used cross tabulations and chi squared tests to determine good predictors for approval. For quantitative variables, I have used boxplots and t tests to determine good predictors for credit card approval.

For categorical variables, A7, A9, A10 seem to have a strong relationship with getting approved or rejected as seen from the chi square test results (p values significantly lower than critical value of 0.05). A2, A4, A5, A6 seems to have a moderately strong relationship with A16 looking at the chi squared test results. A13 seems to have a weak relationship with getting approved or rejected.

For quantitative variables - A8 and A11 seems to have a strong relationship with getting approved and rejected as indicated by the extremely low p values from the t-test, indicating that these results might not have occured due to chance. A3 and A15 seem to have a moderately strong relationship with being approved or rejected.

I am not using A14 as even though it is numerical data, it seems to be categorical data (Something like serial numbers) which I feel might not be a good predictor for card approvals.

So the variables that I am selecting for the model are A2,A4,A5,A6,A7,A9,A10,A13,A3,A8,A11, and A15.

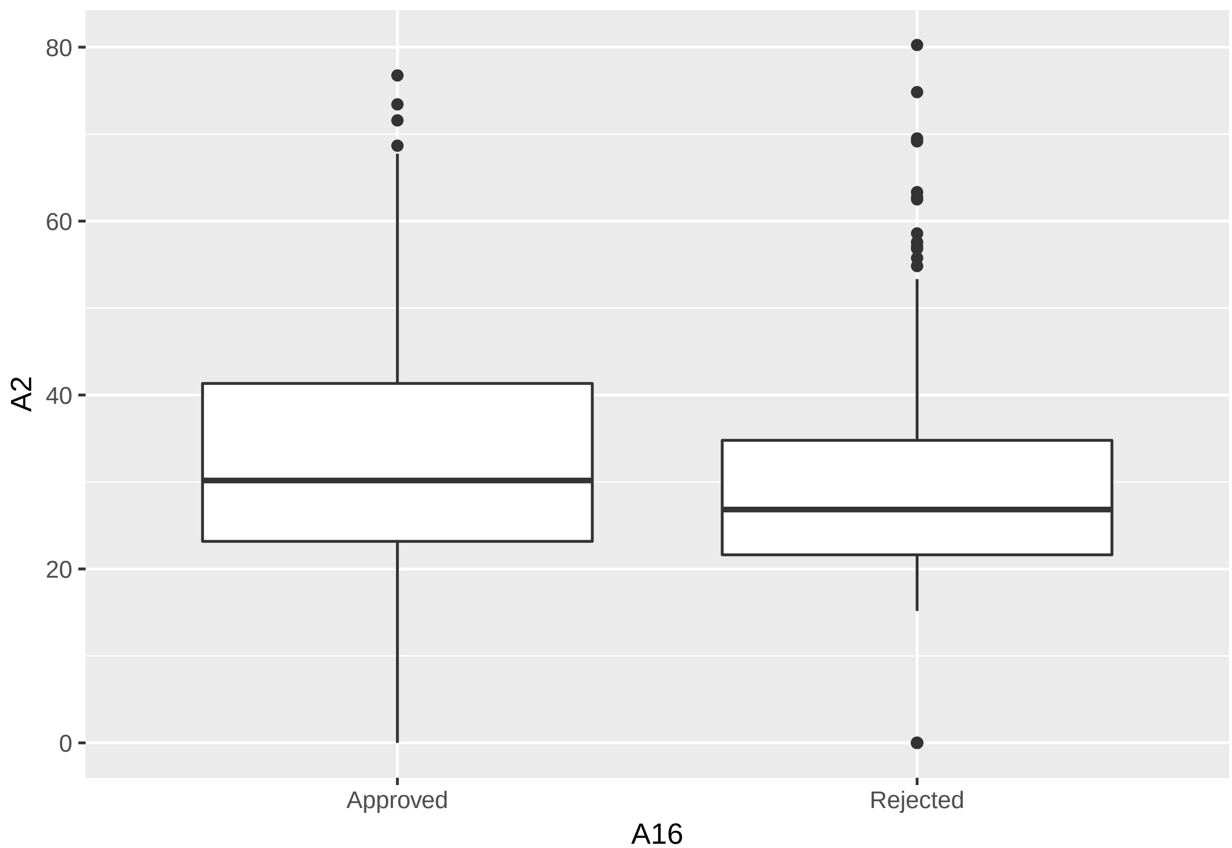
```r
#testing A1
table(credit_card_info$A1,credit_card_info$A16)
```

```
##
##     Approved Rejected
##   ?        3        9
##   a       98      112
##   b      206      262
```

```r
chisq.test(credit_card_info$A1,credit_card_info$A16)
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit_card_info$A1 and credit_card_info$A16
## X-squared = 2.291, df = 2, p-value = 0.3181
```

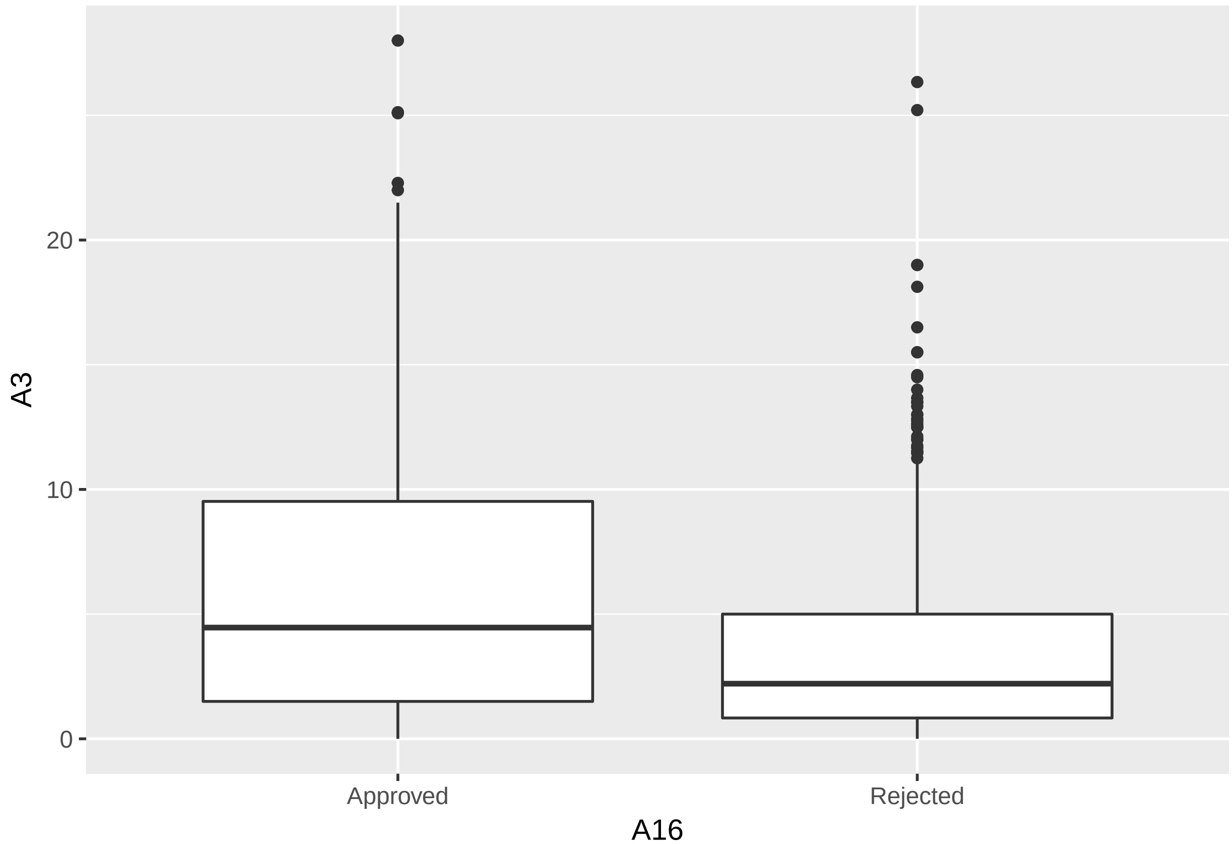
```r
#testing A2
ggplot(data=credit_card_info,aes(x=A16,y=A2))+geom_boxplot()
```




```r
t.test(credit_card_info$A2~credit_card_info$A16)
```

```
##
##  Welch Two Sample t-test
##
## data:  credit_card_info$A2 by credit_card_info$A16
## t = 4.6679, df = 623.17, p-value = 3.727e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.589996 6.351737
## sample estimates:
## mean in group Approved mean in group Rejected
##               33.50081               29.02995
```

```r
#testing A3
ggplot(data=credit_card_info,aes(x=A16,y=A3))+geom_boxplot()
```




```
t.test(credit_card_info$A3~credit_card_info$A16)
```

```
##
##  Welch Two Sample t-test
##
## data:  credit_card_info$A3 by credit_card_info$A16
## t = 5.3925, df = 575.06, p-value = 1.016e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.312876 2.817130
## sample estimates:
## mean in group Approved mean in group Rejected
##               5.904951               3.839948
```

```
#testing A4
table(credit_card_info$A4,credit_card_info$A16)
```

```
##
##     Approved Rejected
##   ?        4        2
##   l        2        0
##   u      256      263
##   y       45      118
```

```
chisq.test(credit_card_info$A4,credit_card_info$A16)
```

```
## Warning in chisq.test(credit_card_info$A4, credit_card_info$A16): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit_card_info$A4 and credit_card_info$A16
## X-squared = 27.416, df = 3, p-value = 4.816e-06
```

```r
#testing A5
table(credit_card_info$A5,credit_card_info$A16)
```

```
##
##      Approved Rejected
##   ?         4        2
##   g       256      263
##   gg        2        0
##   p        45      118
```

```r
chisq.test(credit_card_info$A5,credit_card_info$A16)
```

```
## Warning in chisq.test(credit_card_info$A5, credit_card_info$A16): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit_card_info$A5 and credit_card_info$A16
## X-squared = 27.416, df = 3, p-value = 4.816e-06
```

```r
#testing A6
table(credit_card_info$A6,credit_card_info$A16)
```

```
##
##      Approved Rejected
##   ?         4        5
##   aa       19       35
##   c        62       75
##   cc       29       12
##   d         7       23
##   e        14       11
##   ff        7       46
##   i        14       45
##   j         3        7
##   k        14       37
##   m        16       22
##   q        51       27
##   r         2        1
##   w        33       31
##   x        32        6
```

```r
chisq.test(credit_card_info$A6,credit_card_info$A16)
```

```
## Warning in chisq.test(credit_card_info$A6, credit_card_info$A16): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit_card_info$A6 and credit_card_info$A16
## X-squared = 98.325, df = 14, p-value = 9.921e-15
```

```r
#testing A7
table(credit_card_info$A7,credit_card_info$A16)
```

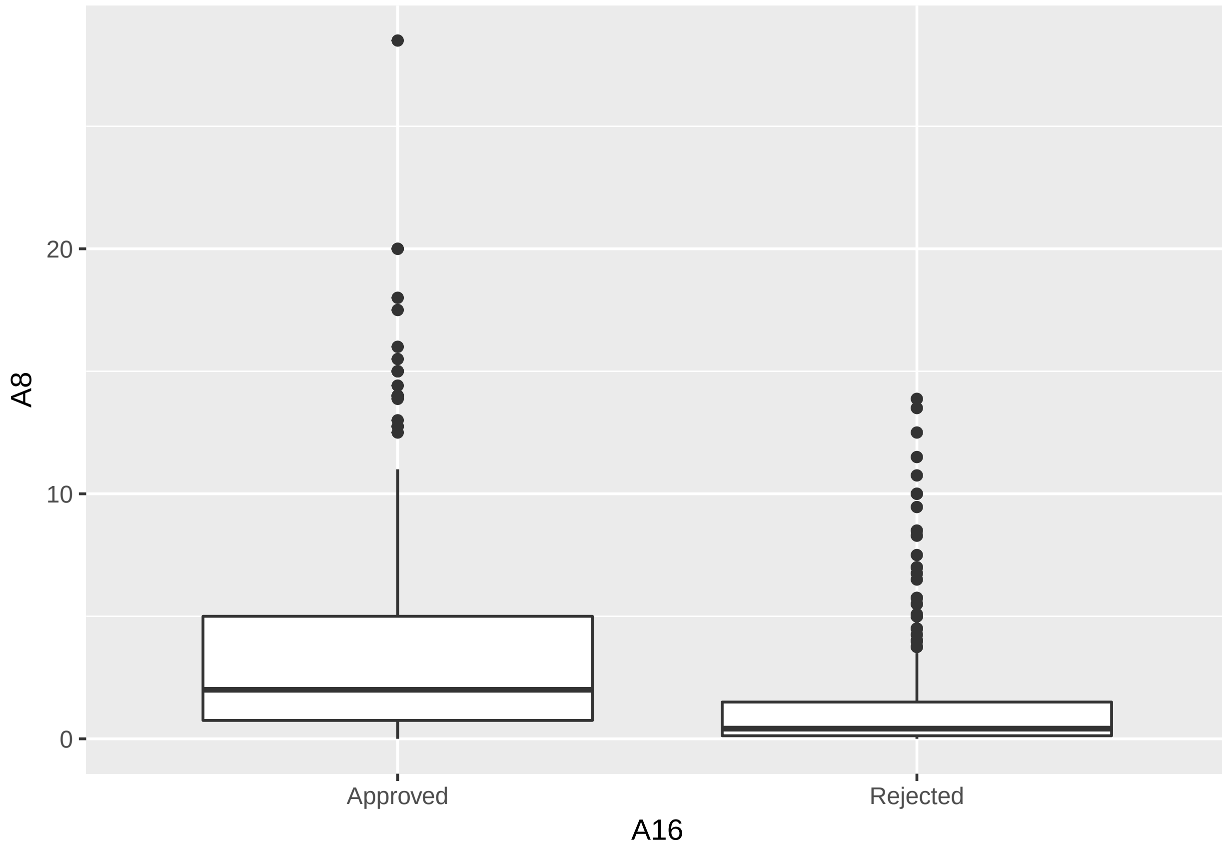
```
##
##      Approved Rejected
##   ?         4        5
##   bb       25       34
##   dd        2        4
##   ff        8       49
##   h        87       51
##   j         3        5
##   n         2        2
##   o         1        1
##   v       169      230
##   z         6        2
```

```r
chisq.test(credit_card_info$A7,credit_card_info$A16)
```

```
## Warning in chisq.test(credit_card_info$A7, credit_card_info$A16): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit_card_info$A7 and credit_card_info$A16
## X-squared = 45.034, df = 9, p-value = 9.093e-07
```

```r
#testing A8
ggplot(data=credit_card_info,aes(x=A16,y=A8))+geom_boxplot()
```




```r
t.test(credit_card_info$A8~credit_card_info$A16)
```

```
##
##  Welch Two Sample t-test
##
## data:  credit_card_info$A8 by credit_card_info$A16
## t = 8.3801, df = 434.02, p-value = 7.425e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.661033 2.678917
## sample estimates:
## mean in group Approved mean in group Rejected
##               3.427899               1.257924
```

```r
#testing A9
table(credit_card_info$A9,credit_card_info$A16)
```

```
##
##     Approved Rejected
##   f       23      306
##   t      284       77
```

```r
chisq.test(credit_card_info$A9,credit_card_info$A16)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit_card_info$A9 and credit_card_info$A16
```

```
## X-squared = 355.2, df = 1, p-value < 2.2e-16
```

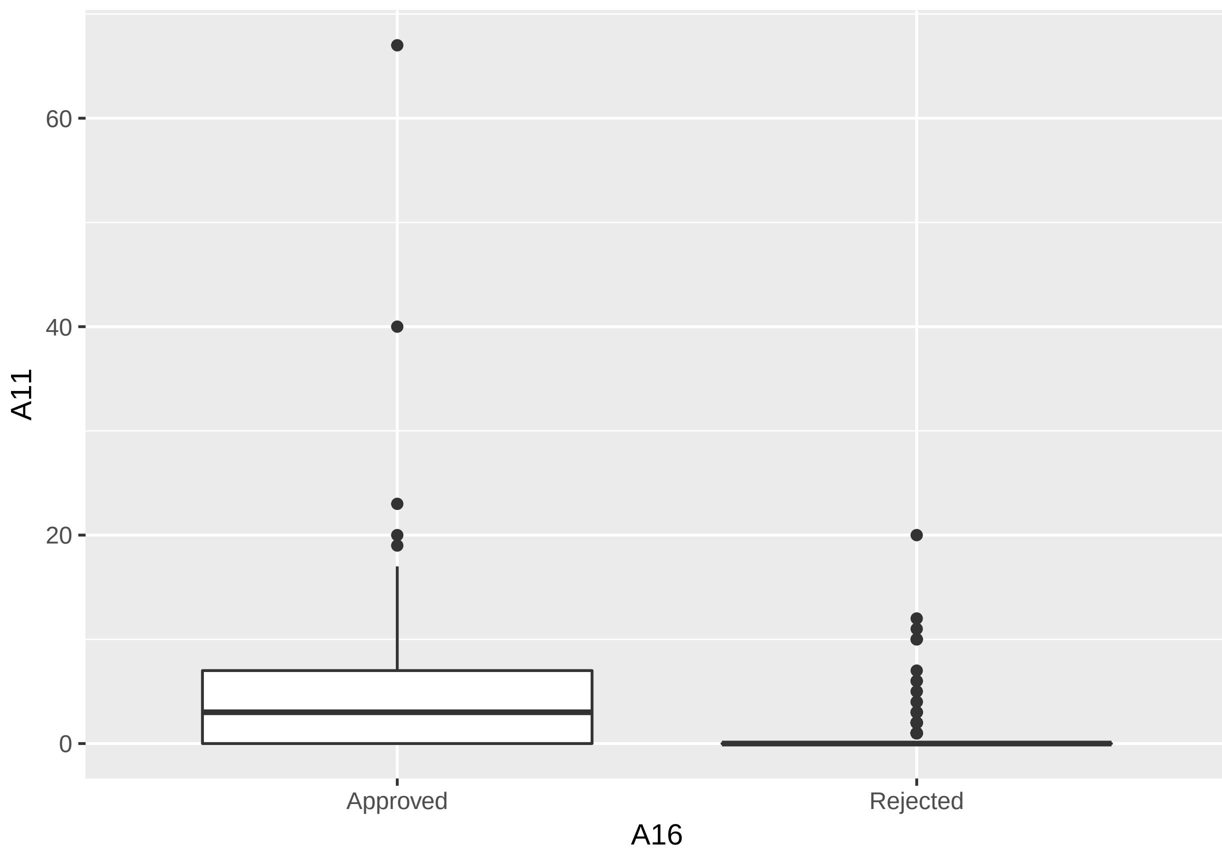
```r
#testing A10
table(credit_card_info$A10,credit_card_info$A16)
```

```
##
##     Approved Rejected
##   f       98      297
##   t      209       86
```

```r
chisq.test(credit_card_info$A10,credit_card_info$A16)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit_card_info$A10 and credit_card_info$A16
## X-squared = 143.07, df = 1, p-value < 2.2e-16
```

```r
#testing A11
ggplot(data=credit_card_info,aes(x=A16,y=A11))+geom_boxplot()
```




```r
t.test(credit_card_info$A11~credit_card_info$A16)
```

```
##
##  Welch Two Sample t-test
##
## data:  credit_card_info$A11 by credit_card_info$A16
## t = 10.638, df = 350.47, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  3.239323 4.708696
## sample estimates:
## mean in group Approved mean in group Rejected
##              4.6058632              0.6318538
```

```r
#testing A12
table(credit_card_info$A12,credit_card_info$A16)
```

```
##
##     Approved Rejected
##   f      161      213
##   t      146      170
```

```r
chisq.test(credit_card_info$A12,credit_card_info$A16)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  credit_card_info$A12 and credit_card_info$A16
## X-squared = 0.56827, df = 1, p-value = 0.4509
```

```r
#testing A13
table(credit_card_info$A13,credit_card_info$A16)
```

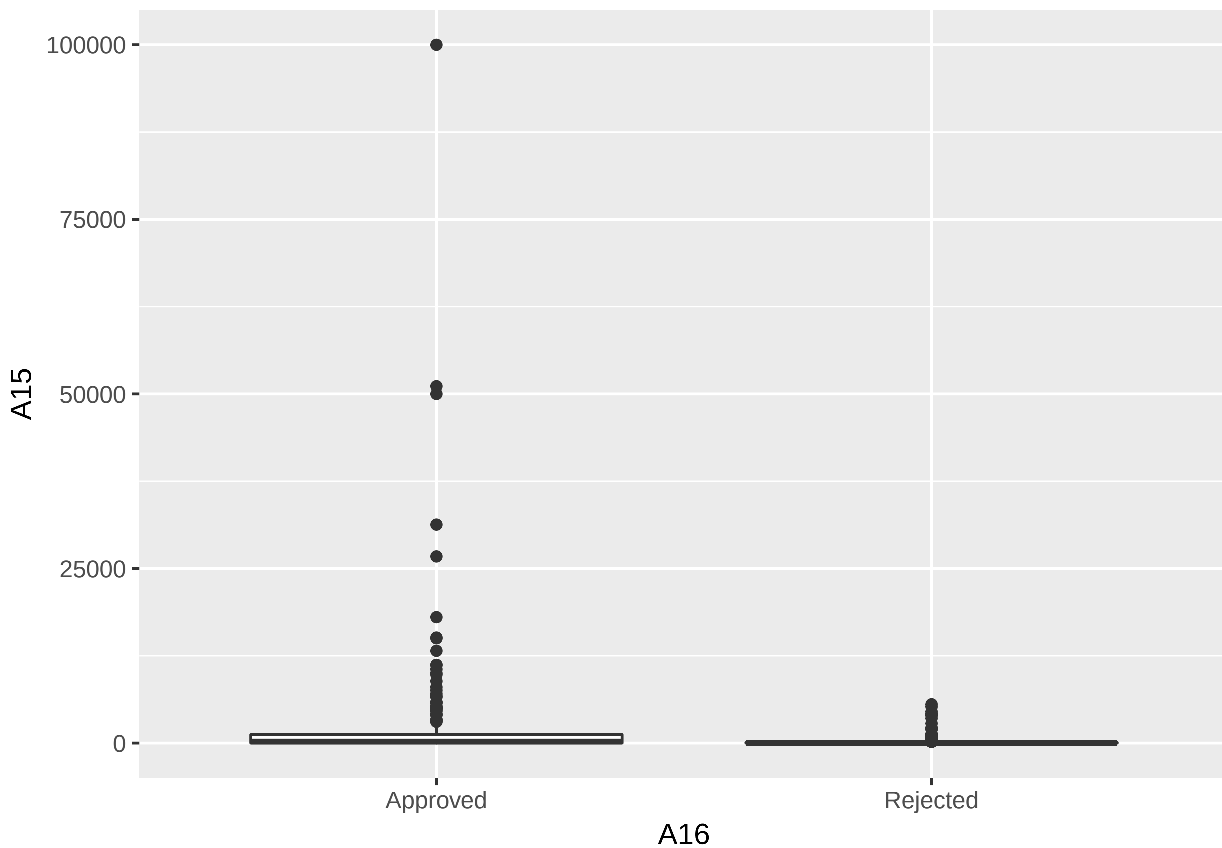
```
##
##     Approved Rejected
##   g      287      338
##   p        5        3
##   s       15       42
```

```r
chisq.test(credit_card_info$A13,credit_card_info$A16)
```

```
## Warning in chisq.test(credit_card_info$A13, credit_card_info$A16): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  credit_card_info$A13 and credit_card_info$A16
## X-squared = 9.1916, df = 2, p-value = 0.01009
```

```r
#testing A15
ggplot(data=credit_card_info,aes(x=A16,y=A15))+geom_boxplot()
```




```r
t.test(credit_card_info$A15~credit_card_info$A16)
```

```
##
##  Welch Two Sample t-test
##
## data:  credit_card_info$A15 by credit_card_info$A16
## t = 4.1966, df = 309.77, p-value = 3.543e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   977.4179 2703.0905
## sample estimates:
## mean in group Approved mean in group Rejected
##              2038.8599               198.6057
```

### 2. Estimate logistic regression

Use these variables to estimate logistic regression models. You may use the function `glm` in the base package, or any other implementation of logistic regression. Use this model to predict the outcome. Make a cross-table of actual/predicted outcomes. Which percentage did you get right? Ans. Using the model, the percent correct predictions are 88.11594%.

```r
#fit model to variables
m <- glm(A16 ~ A2+A4+A5+A6+A7+A9+A10+A13+A3+A8+A11+A15, data=credit_card_info, family
         = binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
#predict outcome i.e approved/rejected
lm <- predict(m,type="response")>0.5
#cross-table of actual/predicted outcomes
pred_table <- table(lm,credit_card_info$A16)
print(pred_table)
```

```
##
## lm      Approved Rejected
##   FALSE      276       51
##   TRUE        31      332
```

```r
#compute percent correct predictions
print('Percent correct predictions:')
```

```
## [1] "Percent correct predictions:"
```

```r
print(sum(diag(pred_table))*100/sum(pred_table))
```

```
## [1] 88.11594
```

### 3. Estimate decision trees.

Use exactly the same variables to compute decision tree models. You may use function `rpart` in the `rpart` package, or any other decision tree implementations in R. As above, predict the result, make a cross-table, and find the correct percentage.

Ans. Using the decision tree model, the percent correct predictions are 88.4058%.

```r
library(rpart)
#estimate decision tree
mod <- rpart::rpart(A16 ~ A2+A4+A5+A6+A7+A9+A10+A13+A3+A8+
                      A11+A15,data=credit_card_info)
#predict survival using model
approvalpred <-predict(mod,type="class")
#create pivot table of obuserved/predicted survival
predtable <-table(credit_card_info$A16,approvalpred)
print(predtable)
```

```
##           approvalpred
##            Approved Rejected
##   Approved      269       38
##   Rejected       42      341
```

```r
#compute percent of correct predictions
print("Percent of correct predictions-")
```

```
## [1] "Percent of correct predictions-"
```

```r
print(sum(diag(predtable)*100/sum(predtable)))
```

```
## [1] 88.4058
```

### 4. Repeat the process

Repeat steps 1,2,3 with 3 different sets of variables. Feel free to do feature engineering.

Ans. For case 1, my logistic model had an accuracy rate of 86.08696%, while the corresponsing decision tree using the same predictors to predict approvals had an accuracy rate of 85.50725%

For case 2, my logistic model had an accuracy rate of 93.76812%, while the corresponsing decision tree using the same predictors to predict approvals had an accuracy rate of 92.89855%

For case 3, my logistic model had an accuracy rate of 88.11594%, while the corresponsing decision tree using the same predictors to predict approvals had an accuracy rate of 88.4058%

```
#case 1
#logistic model 1
#fit model to variables (using only variables which
#seem to have a strong relationship to approval)
m1<- glm(A16 ~ A7+A9+A10+A8+A11, data=credit_card_info, family =
           binomial(link = "logit"))
#predict outcome i.e approved/rejected
lm1 <- predict(m1,type="response")>0.5
#cross-table of actual/predicted outcomes
pred_table1 <- table(lm1,credit_card_info$A16)
print(pred_table1)
```

```
##
## lm1     Approved Rejected
##   FALSE      277       66
##   TRUE        30      317
```

```
#compute percent correct predictions
print('Percent correct predictions:')
```

```
## [1] "Percent correct predictions:"
```

```
print(sum(diag(pred_table1))*100/sum(pred_table1))
```

```
## [1] 86.08696
```

```
#tree model 1
#estimate decision tree
mod1 <- rpart::rpart(A16 ~ A7+A9+
                       A10+A8+A11,data=credit_card_info)
#predict survival using model
approvalpred1 <-predict(mod1,type="class")
#create pivot table of obuserved/predicted survival
predtable1 <-table(credit_card_info$A16,approvalpred1)
print(predtable1)
```

```
##           approvalpred1
##            Approved Rejected
##   Approved      284       23
##   Rejected       77      306
```

```
#compute percent of correct predictions
print("Percent of correct predictions-")
```

```
## [1] "Percent of correct predictions-"
```

```
print(sum(diag(predtable1)*100/sum(predtable1)))
```

```
## [1] 85.50725
```

```r
#logistic model 2 (Using all variables)
#fit model to variables
m2<- glm(A16 ~ A2+A4+A5+A6+A7+A9+A10+A12+A13+A14+
           A15+A3+A8+A11, data=credit_card_info, family =
           binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
#predict outcome i.e approved/rejected
lm2 <- predict(m2,type="response")>0.5
#cross-table of actual/predicted outcomes
pred_table2 <- table(lm2,credit_card_info$A16)
print(pred_table2)
```

```
##
## lm2     Approved Rejected
##   FALSE      291       27
##   TRUE        16      356
```

```r
#compute percent correct predictions
print('Percent correct predictions:')
```

```
## [1] "Percent correct predictions:"
```

```r
print(sum(diag(pred_table2))*100/sum(pred_table2))
```

```
## [1] 93.76812
```

```r
#tree model 2
#estimate decision tree
mod2 <- rpart::rpart(A16 ~ A2+A4+A5+A6+A7+A9+A10+A12+A13+A14+
                       A15+A3+A8+A11,data=credit_card_info)
#predict survival using model
approvalpred2 <-predict(mod2,type="class")
#create pivot table of obuserved/predicted survival
predtable2 <-table(credit_card_info$A16,approvalpred2)
print(predtable2)
```

```
##           approvalpred2
##            Approved Rejected
##   Approved      291       16
##   Rejected       33      350
```

```r
#compute percent of correct predictions
print("Percent of correct predictions-")
```

```
## [1] "Percent of correct predictions-"
```

```r
print(sum(diag(predtable2)*100/sum(predtable2)))
```

```
## [1] 92.89855
```

```r
#logistic model 3 (Removing A13 from model which seemed to have weak relationship with A16)
#fit model to variables
m3<- glm(A16 ~ A2+A4+A5+A6+A7+A9+A10+A3+A8+A11+A15,
         data=credit_card_info, family = binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
#predict outcome i.e approved/rejected
lm3 <- predict(m3,type="response")>0.5
#cross-table of actual/predicted outcomes
pred_table3 <- table(lm3,credit_card_info$A16)
print(pred_table3)
```

```
##
## lm3     Approved Rejected
##   FALSE      276       51
##   TRUE        31      332
```

```r
#compute percent correct predictions
print('Percent correct predictions:')
```

```
## [1] "Percent correct predictions:"
```

```r
print(sum(diag(pred_table3))*100/sum(pred_table3))
```

```
## [1] 88.11594
```

```r
#tree model 3
#estimate decision tree
mod3 <- rpart::rpart(A16 ~ A2+A4+A5+A6+A7+A9+A10+
                       A3+A8+A11+A15,data=credit_card_info)
#predict survival using model
approvalpred3 <-predict(mod3,type="class")
#create pivot table of obuserved/predicted survival
predtable3 <-table(credit_card_info$A16,approvalpred3)
print(predtable3)
```

```
##           approvalpred3
##            Approved Rejected
##   Approved      269       38
##   Rejected       42      341
```

```r
#compute percent of correct predictions
print("Percent of correct predictions-")
```

```
## [1] "Percent of correct predictions-"
```

```r
print(sum(diag(predtable3)*100/sum(predtable3)))
```

```
## [1] 88.4058
```

### 5. Compare the models

Which model performed best overall? Did logistic regression or decision trees perform better generally?

Ans.

The models in which I considered all the variables as predictors for credit card approval performed better than any other model on an overall level. My logistic model 2 which considered all variables performed the best with an approval prediction accuracy rate of 93.76812%.

Generally speaking, out of 4 cases, in 2 cases the logistic model performed better, and 2 two cases the decision tree performed better. This suggests that in some cases the logitic model would berform better and in some cases the decision tree might perform better.