Credit Card ML Analysis Report #2

Second Report

2a. Logistic Regression
I have 11 different variables for my y so I will choose 5 variables that I believe have the largest effect - reports, share, selfemp, majorcards,active.

Importing Data, Packages, and Libraries

```
library(AER)
```

```
## Loading required package: car

## Loading required package: carData

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```
data(CreditCard)
CreditCard$card01 <- ifelse(CreditCard$card=="yes", 1, 0)
```

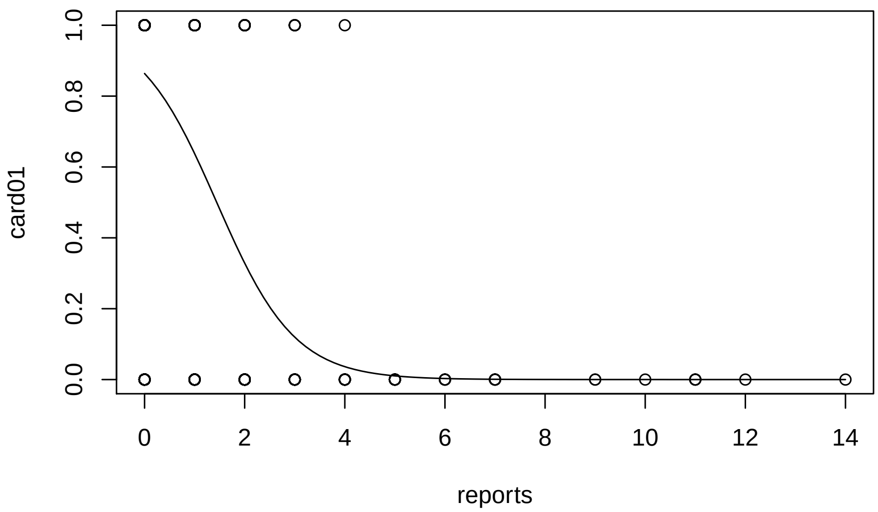
```
attach(CreditCard)
glm.fits1 <- glm(card~reports+share+selfemp+majorcards+active,
                 data = CreditCard, family = binomial)
glm.probs1 = predict(glm.fits1,CreditCard,type="response")
glm.pred1 = rep(0,length(glm.probs1))
glm.pred1[glm.probs1>.99]=1
table1 = table(glm.pred1,CreditCard$card)
logperf = (sum(diag(table1)))/sum(table1)
table1
```

```
##
## glm.pred1  no yes
##         0 296  46
##         1   0 977
```

```
logperf
```

```
## [1] 0.9651251
```

```
plot(reports,card01,xlab="reports",ylab="card01")
g=glm(card~reports,family=binomial,data = CreditCard)
curve(predict(g,data.frame(reports=x),type="resp"),add=TRUE)
```



The logistic regression graph above shows how the number of derogatory reports looks like when plotted against card.

The logistic regression used with the variables reports, share, +selfemp, majorcards, active had a 97.4981% classification rate. The five variables that I think most correctly predicted my model with logistic regressions are reports, share, selfemp, majorcards, active, data. I think those most logically predict my data because of the fact that negative affects of the X would negatively affect the Y. For example, having a high number of derogatory reports would cause someone to not be accepted for a credit card. While observing the correct predictions, true negatives and true positives, I see that the error rate is .974981 which is extremely high.

I will see the performance of logistic regression using all variables.

```
data("CreditCard")
library(class)
library(dplyr)
data("CreditCard")
CreditCard$card <- ifelse(CreditCard$card=="yes", 1, 0)
CreditCard$owner <- ifelse(CreditCard$owner=="yes", 1, 0)
CreditCard$selfemp <- ifelse(CreditCard$selfemp=="yes", 1, 0)
set.seed(123)
train = CreditCard %>% sample_frac(.7)
test = CreditCard %>% setdiff(train)
```

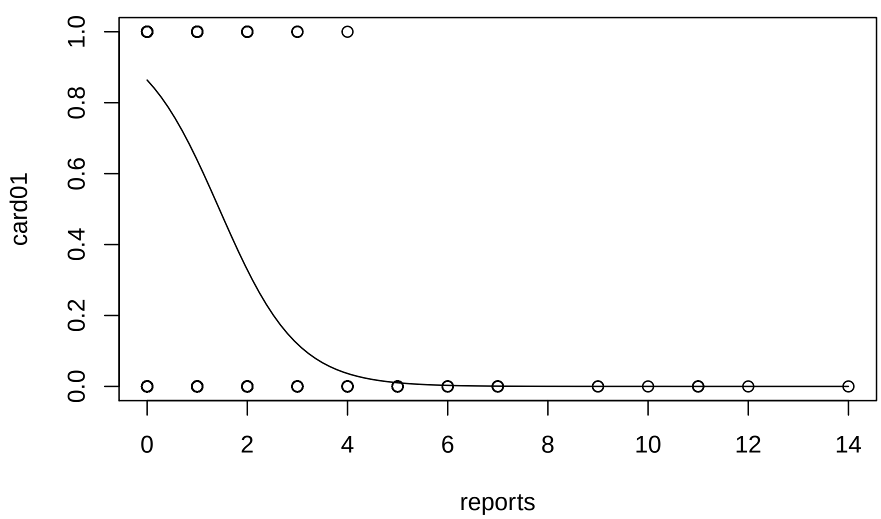
```
glm.fits1 <- glm(card~.,
                 data = train, family = "binomial")
logitmodel = predict(glm.fits1,test,type="response")
logpred = rep(0,length(logitmodel))
logpred[logitmodel>.1]=1
logit_matrix = table(logpred,test$card)
logperf = (sum(diag(logit_matrix)))/sum(logit_matrix)
logit_matrix
```

```
##
## logpred   0   1
##       0  67   3
##       1  17 309
```

```
logperf
```

```
## [1] 0.9494949
```

```
plot(reports,card01,xlab="reports",ylab="card01")
g=glm(card~reports,family=binomial,data = CreditCard)
curve(predict(g,data.frame(reports=x),type="resp"),add=TRUE)
```



The logit matrix above represents how the classification changes as the number of reports increases. This is a logical graph because as the number of derogatory reports a person has increases, they should start getting denied for a credit card application.

```
logit_matrix = table(logpred,test$card)
logperf = (sum(diag(logit_matrix)))/sum(logit_matrix)
logit_matrix
```

```
##
## logpred   0   1
##       0  67   3
##       1  17 309
```

```r
logperf
```

```
## [1] 0.9494949
```

When running logistic regression on all the available variables, I see that I get an accuracy of 94.94%. **2b. K Nearest Neighbors**

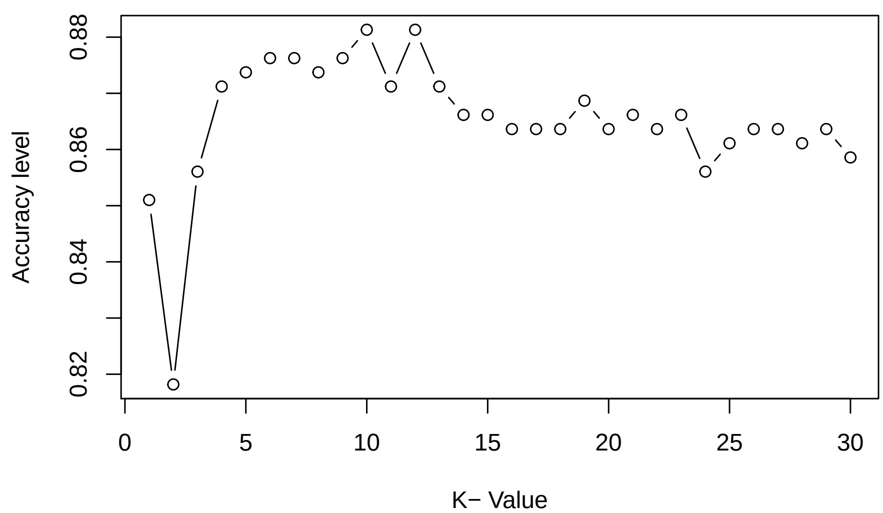
```r
library(class)
library(dplyr)
data("CreditCard")
CreditCard$card <- ifelse(CreditCard$card=="yes", 1, 0)
CreditCard$owner <- ifelse(CreditCard$owner=="yes", 1, 0)
CreditCard$selfemp <- ifelse(CreditCard$selfemp=="yes", 1, 0)
set.seed(123)
train = CreditCard %>% sample_frac(.7)
test = CreditCard %>% setdiff(train)

X_card_trn = train[, -1]
Y_card_trn = train$card

# testing data
X_card_tst = test[, -1]
Y_card_tst = test$card
set.seed(123)
card_pred = knn(train = scale(X_card_trn),
                test  = scale(X_card_tst),
                cl    = Y_card_trn,
                k     = 12,
                prob  = TRUE)

set.seed(123)
i=1
k.optm=1
bestk = 0;
besti = 0;
for (i in 1:30){
  set.seed(123)
  knn.mod <- knn(train=scale(X_card_trn), test=scale(X_card_tst), cl=Y_card_trn,
                 k=i, prob = TRUE)
  k.optm[i] <- sum(Y_card_tst == knn.mod)/NROW(Y_card_tst)
  k=i
   if(k.optm[i]>bestk) {
     bestk = k.optm[i]
     besti = k
   }
}
bestk
```

```
## [1] 0.8813131
```

```
besti
```

```
## [1] 10
```

```
plot(k.optm, type="b", xlab="K- Value",ylab="Accuracy level")
```




```
knn_matrix = table(card_pred, Y_card_tst)
knnperf = (sum(diag(knn_matrix)))/sum(knn_matrix)
knn_matrix
```

```
##          Y_card_tst
## card_pred   0   1
##         0  46   9
##         1  38 303
```

```
knnperf
```

```
## [1] 0.8813131
```

This K nearest neighbors output correctly classifies 88.1% of my testing data. Also we see that the optimal value of K is 10.

```
mlperformance<-matrix(c(logperf, knnperf),ncol=1,byrow=TRUE)
rownames(mlperformance)<-c("Logistic Regression", "K Nearest Neighbors")
colnames(mlperformance)<-c("Performance")
mlperformance <- as.table(mlperformance)
mlperformance
```

```
##                     Performance
## Logistic Regression   0.9494949
## K Nearest Neighbors   0.8813131
```