Credit Card ML Analysis Report #2

Introduction

This is the second report for my Credit Card Acceptance Project. In this part of the project, I will be incorporating the comments I received on the first project while also running logistic regressions on my y-variable with several x-variables. I can observe the correct predictions by having a table of true positives, false positives, true negatives, and false negatives. This is because my Y, or cards, is a categorical/classification. I will try to see the effect of different combinations of X's and also observing the effect of taking some variables away. I will also be trying to run a K-nearest neighbors classification on the data as well.

2.1

```
library(ISLR)
library(AER)
library(ggplot2)
data("CreditCard")
CreditCard = data.frame(CreditCard)
names(CreditCard)
```

```
##  [1] "card"        "reports"     "age"         "income"      "share"
##  [6] "expenditure" "owner"       "selfemp"     "dependents"  "months"
## [11] "majorcards"  "active"
```

```
nrow(CreditCard)
```

```
## [1] 1319
```

I have 11 different variables for my y so I will choose 5 variables that I believe have the largest effect.

```
glm.fits1 <- glm(card~reports+share+selfemp+majorcards+active,data = CreditCard, fami
ly = binomial())
```

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

```
glm.probs1 = predict(glm.fits1,CreditCard,type="response")
glm.pred1 = rep(0,length(glm.probs1))
glm.pred1[glm.probs1>.5]=1
table1 = table(glm.pred1,CreditCard$card)
table1
```

```
##
## glm.pred1   no  yes
##         0  294   23
##         1    2 1000
```

```
prob1 = (1000+294)/1319
prob1
```

```
## [1] 0.9810462
```

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 .9810462 which is extremely high.

```
glm.fits2 <- glm(card~reports+share+selfemp+dependents+months,data = CreditCard, fami
ly = binomial())
```

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

```
glm.probs2 = predict(glm.fits2,CreditCard,type="response")
glm.pred2 = rep(0,length(glm.probs2))
glm.pred2[glm.probs2>.5]=1
table2 = table(glm.pred2,CreditCard$card)
table2
```

```
##
## glm.pred2  no yes
##         0 295  25
##         1   1 998
```

```
prob2 = (998+295)/1319
prob2
```

```
## [1] 0.9802881
```

Next, I decided to replace majorcards and active from Model 1 with dependents and months. I decided to run a logistic regression and it had an error rate of .9802881. This was lower than the first one so it might say that having majorcards and active were better at helping the model predict than dependents and months. This makes sense because I chose the variables from Model 1 based on my own intuition on having great influence on those variables.

```
glm.fits3 <- glm(card~reports+share+selfemp,data = CreditCard, family = binomial())
```

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

```
glm.probs3 = predict(glm.fits3,CreditCard,type="response")
glm.pred3 = rep(0,length(glm.probs3))
glm.pred3[glm.probs3>.5]=1
table3 = table(glm.pred3,CreditCard$card)
table3
```

```
##
## glm.pred3  no yes
##         0 295  25
##         1   1 998
```

```
prob3 = (998+295)/1319
prob3
```

```
## [1] 0.9802881
```

For Model 3, I wanted to check how taking away variables would affect my logistic regression. With this, I decided to see the effect of my first three variables that I chose(which were the same for Model1 and Model2). These variables were reports, share, and selfempl. When looking at this, I saw that the error rate is.9802881 which means that it correctly predicted it about 98% of the time, which is also the same as when I had the two extra variables in Model 2. This does not make sense since having more variables should help our model predict better, but it is less in this case.

```
glm.fits4 <- glm(card~reports,data = CreditCard, family = binomial())
glm.probs4 = predict(glm.fits4,CreditCard,type="response")
glm.pred4 = rep(0,length(glm.probs4))
glm.pred4[glm.probs4>.5]=1
table4 = table(glm.pred4,CreditCard$card)
table4
```

```
##
## glm.pred4   no  yes
##         0  104   18
##         1  192 1005
```

```
prob4 = (1005+104)/1319
prob4
```

```
## [1] 0.8407885
```

In this model, I decided to see the effect of one variable on the model. I decided to do a logistic regression based on the model with the X, reports. I see that the it predicted it correctly .8407885, or ~84%. This makes sense because if we decrease the X to one variable, it would be harder to correctly predict the Y.

```
glm.fits5 <- glm(card~reports+income+owner+share+selfemp+dependents+majorcards+active
,data = CreditCard, family = binomial())
```

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

```
glm.probs5= predict(glm.fits5,CreditCard,type="response")
glm.pred5 = rep(0,length(glm.probs5))
glm.pred5[glm.probs5>.5]=1
table5 = table(glm.pred5,CreditCard$card)
table5
```

```
##
## glm.pred5   no  yes
##         0  293   23
##         1    3 1000
```

```
prob5 = (1000+293)/1319
prob5
```

```
## [1] 0.9802881
```

In this fifth model, I decided to increase the number of variables to 8 variables. The error rate became .9802881, which is less than my first model. This does not make sense because my first model's X were present in this model and this model included more variables. Since we had more variables to predict, it should have been a higher number but it was not. This means that the variables I added dependents, majorcards, and active, do not help us predict the model too well.

2.2

```
summary(glm.fits1)
```

```
##
## Call:
## glm(formula = card ~ reports + share + selfemp + majorcards +
##     active, family = binomial(), data = CreditCard)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.406   0.000   0.000   0.000   2.900
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.07873    0.60417  -6.751 1.47e-11 ***
## reports       -2.57145    0.96748  -2.658  0.00786 **
## share       2610.28045  482.96805   5.405 6.49e-08 ***
## selfempyes     0.46171    0.65609   0.704  0.48160
## majorcards     0.52302    0.53461   0.978  0.32791
## active         0.10532    0.03125   3.371  0.00075 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1404.6  on 1318  degrees of freedom
## Residual deviance:  146.5  on 1313  degrees of freedom
## AIC: 158.5
##
## Number of Fisher Scoring iterations: 16
```

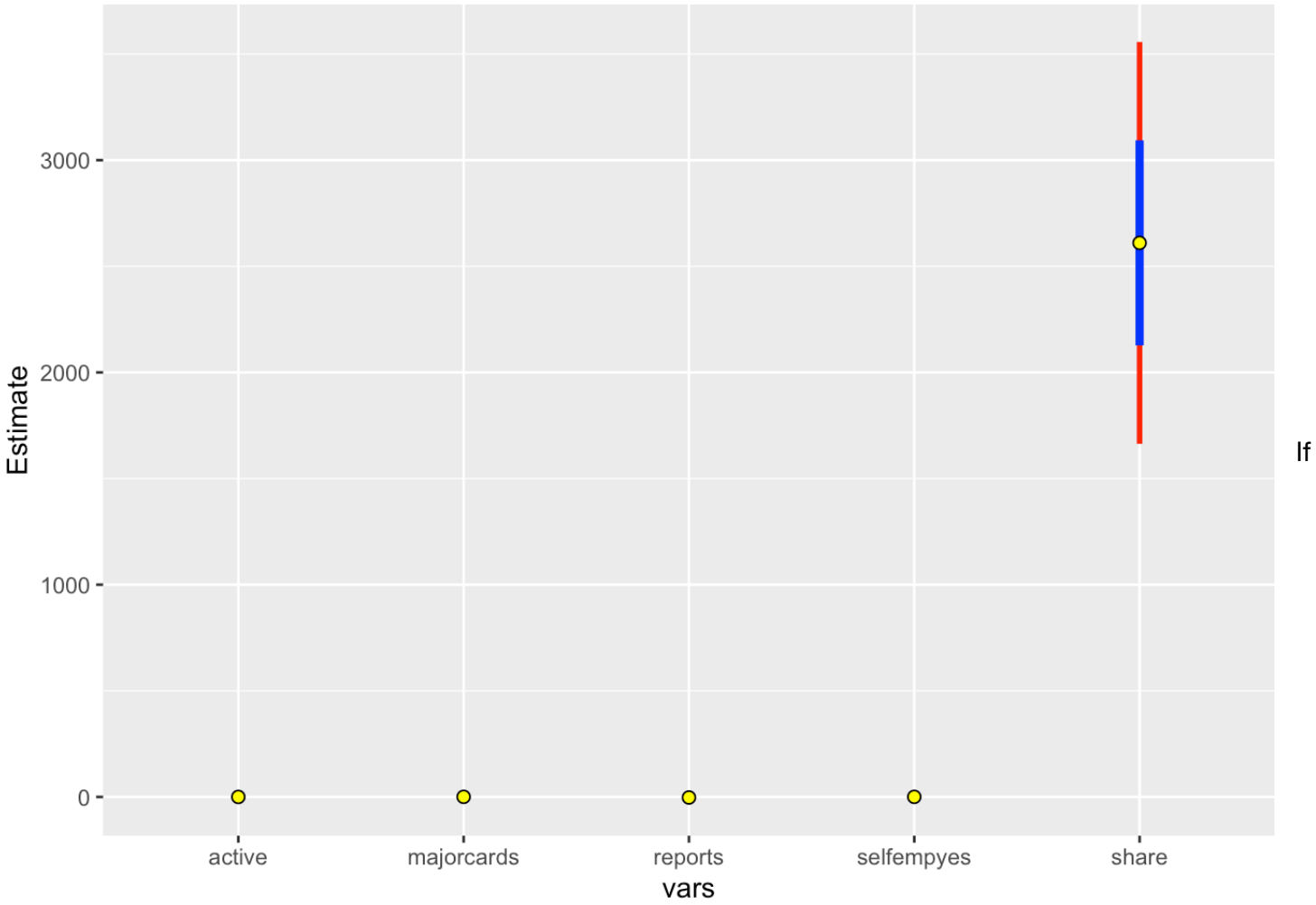
Model 1 is my best regression with the following X: reports, share, selfempl, majorcards, and active. The effect, or the values of their estimates in summary, is -4.07873, -2.57145, 2610, 0.46171, .52302, and .10532, respectively. The negative coefficient, reports, for this predictor suggests that if the reports suggest that the reports were higher for one person, they were more likely to be rejected for a credit card. This definitely makes sense because as I stated before, someone who has a higher number of derogatory reports should not be accepted for a credit card as much as someone who does not. Additionally, when looking at the P-values, I see that the p-value for reports is .00786 which indicates that there is some association between reports and credit card acceptances. Additionally, the other coefficients indicate the effect on that variable on the output, y. The coefficients are different from 0 however, selfempyes, majorcards, and active are not too significantly different from 0.

2.3

Model 1 is my best model with a correct prediction 98.10462% of the time. The true positive rate is 1000/1023 yes' and for the false positives, it is 23/1000. It tells me that the model is very reliable when predicting the classifications for y.

2.4

```
glm.fits1summary=summary(glm.fits1)
coefs=as.data.frame(glm.fits1summary$coefficients[-1,1:2])
names(coefs)[2]="se"
coefs$vars=rownames(coefs)
ggplot(coefs, aes(vars,Estimate)) +
  geom_errorbar(aes(ymin=Estimate-1.96*se,ymax=Estimate+1.96*se),lwd=1, colour="red",
width=0)+
  geom_errorbar(aes(ymin=Estimate - se,ymax=Estimate+se),lwd=1.5,colour="blue",width=
0)+
  geom_point(size=2,pch=21,fill="yellow")
```



If the graphs intersect 0 they're not significant and if they do then that variable is significant. We see that for share,

PROBIT REGRESSION

```
glm.fits=glm(card~reports+share+selfemp+majorcards+active,data=CreditCard,family = bi
nomial(link = "probit"))
```

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

```
glm.probs=predict(glm.fits,CreditCard,type="response")
glm.pred=rep("0",length(glm.probs))
glm.pred[glm.probs>.5]="1"
table(glm.pred,CreditCard$card)
```

```
##
## glm.pred  no yes
##        0 294  24
##        1   2 999
```

```
mean(glm.pred==CreditCard$card)
```

```
## [1] 0
```

```
logitProb = (999+294)/1319
logitProb
```

```
## [1] 0.9802881
```

When running the probit regression, I see that the standard error is .9802881. Compared to my Logistic regression, which had .981. The logistic regression predicted my model better than my probit regression. The true positives and true negatives were 999 and 294 respectively.