Craig Van Vliet & Ryan Thomas

ECO 7100 – Econometrics 1

April 3rd, 2019

Lab: Boosting, KNN, and Logistic Regression

Assignment Submission

1) This question uses the Caravan data set.

(a) Create a training set consisting of the first 1,000 observations, and a test set consisting of the remaining observations.

```
library("np")
library(gbm)
library(ISLR)
library(class)

set.seed(342)
summary(Caravan)
c<-Caravan
c$Purchase<-ifelse(c$Purchase=="Yes",1,0)
summary(c$Purchase)

train<-c[1:1000,]
test<-c[1001:5822,]
```

(b) Fit a boosting model to the training set with Purchase as the response and the other variables as predictors. Use 1,000 trees, and a shrinkage value of 0.01. Which predictors appear to be the most important?

```
boost.caravan=gbm(Purchase~.,data=train,n.trees=1000,shrinkage=.01)
summary(boost.caravan)
```

```
> summary(boost.caravan)
              var     rel.inf
PPERSAUT PPERSAUT 15.15534009
MKOOPKLA MKOOPKLA  9.23499526
MOPLHOOG MOPLHOOG  8.67017024
MBERMIDD MBERMIDD  5.39403655
MGODGE     MGODGE  5.03047673
PBRAND     PBRAND  4.83740038
MINK3045 MINK3045  3.94305387
ABRAND     ABRAND  3.69692919
MOSTYPE   MOSTYPE  3.38768960
PWAPART   PWAPART  2.51970169
MGODPR     MGODPR  2.43689096
MSKC         MSKC  2.34594774
```

The predictors PPERSAUT, MKOOPKLA, and MOPLHOOG are most important because they have the highest relative influence on prediction accuracy.

(c) Use the boosting model to predict the response on the test data. Predict that a person will make a purchase if the estimated probability of purchase is greater than 20 %. Form a confusion matrix. What fraction of the people predicted to make a purchase do in fact make one? How does this compare with the results obtained from applying KNN and logistic regression to this data set?

```
yhat.boost=predict(boost.caravan,newdata=test,distribution="bernoulli",type="response",n.
trees=1000)
summary(yhat.boost)
plot(yhat.boost)

boost.YesNo=rep("No",4822)
boost.YesNo[yhat.boost>=.2]="yes"
Original=rep("No",4822)
Original[test$Purchase==1]="yes"
table(boost.YesNo,Original)
```

```
           Original
boost.YesNo   No  yes
        No  4396  255
        yes  137   34
```

34/171 = 19.88% correctly predicted purchases using boosting

K-Nearest Neighbors

```
rm(list = ls())
attach(Caravan)

standardized.X=scale(Caravan[,-86])
var(Caravan[,1])
var(Caravan[,2])
var(standardized.X[,1])
var(standardized.X[,2])

test=1:1000
train.X=standardized.X[test,]
test.X=standardized.X[-test,]
train.Y=Purchase[test]
test.Y=Purchase[-test]
set.seed(342)
knn.pred=knn(train.X,test.X,train.Y,k=1)
mean(test.Y!=knn.pred)
mean(test.Y!="No")

table(knn.pred,test.Y)
```

```
        test.Y
knn.pred   No  Yes
     No  4249  252
     Yes  284   37
```

37/321 = 11.53% correctly predicted purchases using KNN with k=1

```
knn.pred=knn(train.X,test.X,train.Y,k=3)
table(knn.pred,test.Y)
```

```
        test.Y
knn.pred   No  Yes
     No  4437  264
     Yes   96   25
```

25/121 = 20.66% correctly predicted purchases using KNN with k=3

```
knn.pred=knn(train.X,test.X,train.Y,k=5)
table(knn.pred,test.Y)
```

```
        test.Y
knn.pred   No  Yes
     No  4506  279
     Yes   27   10
```

10/37 = 27.02% correctly predicted purchases using KNN with k=5

<u>Logistic Regression</u>

```
rm(list=ls())
attach(Caravan)

c<-Caravan
c$Purchase<-ifelse(c$Purchase=="Yes",1,0)
train<-c[1:1000,]
test<-c[1001:5822,]
Original=rep("No",4822)
Original[test$Purchase==1]="yes"

logitm<-glm(Purchase~.,family=binomial(link=logit),data=train)
yhat.logitm=predict(logitm,newdata=test,type="response")
summary(yhat.logitm)
log.YesNo=rep("No",4822)
log.YesNo[yhat.logitm>.2]="yes"
table(log.YesNo,Original)
```

```
         Original
log.YesNo   No  yes
      No  4183  231
      yes  350   58
```

58/408 = 14.22% correctly predicted purchases using logistic regression

Overall, boosting predicted purchases better than logistic regression and k-nearest neighbors with k=1, but did not perform quite as well as k-nearest neighbors with higher k values of 3 and 5.