TIM 209

Assignment 3B

Shilpi Jaiswal

1. P(x)=(e^(-6+0.05(X1)+X2))/(1+ e^(-6+0.05(X1)+X2))

a).

X1=40

X2=3.5

P(x)=0.607/1.607=0.3774

b).

P(x)=0.5

X2=3.5

Ln(0.5/0.5)=-6+0.05(X1)+3.5

0=-6+0.05(X1)+3.5

X1=50

2.

Pk(4)=(0.8\*e^((-1/2)\*(4-10)^2)/36)/( (0.8\*e^((-1/2)\*(4-10)^2)/36)+ (0.2\*e^((-1/2)\*(4-0)^2)/36))

=0.4852/(0.4852+0.1601)

=0.7519

3.

a).

*library(MASS)*

*library(ISLR)*

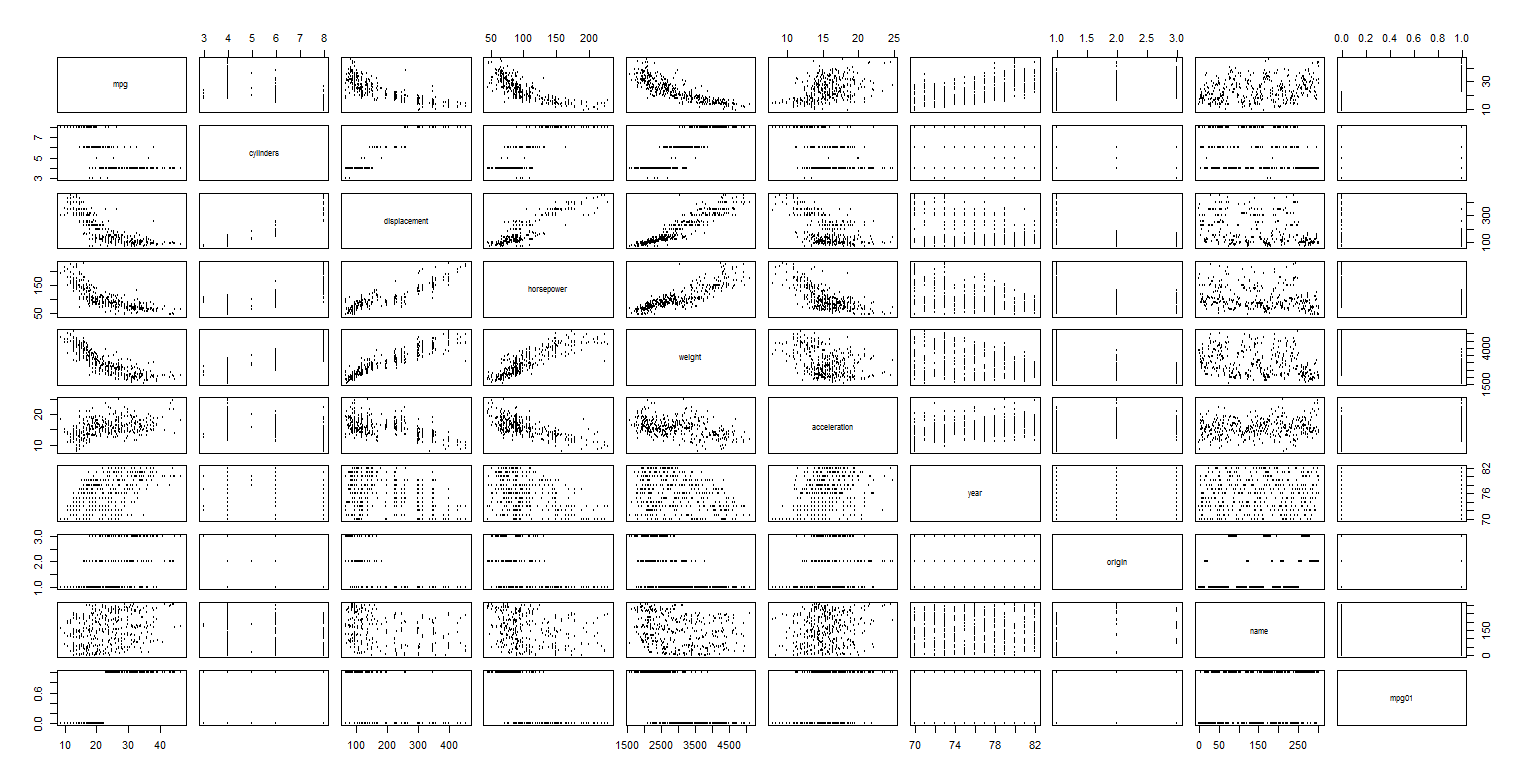
*library(class)*

*Auto<-Auto*

*Auto$mpg01<-ifelse(Auto$mpg>median(Auto$mpg),1,0)*

b).

*pairs(Auto, cex=1)*



Horsepower, displacement, weight, and acceleration seem have more direct correlations with mpg01

c).

*test.ind<-sample(nrow(Auto), 100)*

*testing<-Auto[test.ind,]*

*training<-Auto[-test.ind,]*

d).

*lda.fit<-lda(mpg01∼displacement+horsepower+weight+acceleration ,data =training)*

*lda.fit*

*lda.test<-predict(lda.fit,testing)$class*

*mean(lda.test!=testing$mpg01)*

Test error is 0.13

e).

*qda.fit<-qda(mpg01∼displacement+horsepower+weight+acceleration ,data =training)*

*qda.fit*

*qda.test<-predict(qda.fit,testing)$class*

*mean(qda.test!=testing$mpg01)*

Test error is 0.11

f).

*log.fit<-glm(mpg01∼displacement+horsepower+weight+acceleration ,family=binomial ,data =training)*

*log.fit*

*log.prob<-predict(log.fit, testing, type="response")*

*log.test<-ifelse(log.prob>0.5,1,0)*

*mean(log.test!=testing$mpg01)*

Test error is 0.11

g).

*KNN.training<-training[,c("displacement", "horsepower", "weight", "acceleration")]*

*KNN.testing<-testing[,c("displacement", "horsepower", "weight", "acceleration")]*

*KNN.test1<-knn(KNN.training, KNN.testing, training$mpg01,k=1)*

*mean(KNN.test1!=testing$mpg01)*

*KNN.test2<-knn(KNN.training, KNN.testing, training$mpg01,k=2)*

*mean(KNN.test2!=testing$mpg01)*

*KNN.test3<-knn(KNN.training, KNN.testing, training$mpg01,k=3)*

*mean(KNN.test3!=testing$mpg01)*

*KNN.test4<-knn(KNN.training, KNN.testing, training$mpg01,k=4)*

*mean(KNN.test4!=testing$mpg01)*

*KNN.test5<-knn(KNN.training, KNN.testing, training$mpg01,k=5)*

*mean(KNN.test5!=testing$mpg01)*

*KNN.test10<-knn(KNN.training, KNN.testing, training$mpg01,k=10)*

*mean(KNN.test10!=testing$mpg01)*

K=4 yields the highest accuracy of 0.08 test error