Data Mining - Lab

EXERCIZE I (3 points)

We will now perform cross-validation on a simulated data set. Generate a simulated data set as follows:

```
set.seed(1)
y=rnorm(100)
x=rnorm(100)
y=1+x+rnorm(100)
```

(a) Report in **output** the LOOCV errors that result from fitting the following four models using least squares:

i. $Y = \beta_0 + \epsilon$

ii. $Y = \beta_0 + \beta_1 X + \epsilon$

iii. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$

iv. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$

```
train = data.frame(y,x)
library(boot)
LOOCV <- vector()
LOOCV[1] <- cv.glm(train, glm(y ~ 1))$delta[1]
for (d in 1:3){
LOOCV[d+1] = cv.glm(train, glm(y ~ poly(x,d))  )$delta[1]
}
names(LOOCV) <- c("i","ii","iii","iv")
LOOCV
```

```
##        i       ii      iii       iv
## 1.908168 1.096903 1.086596 1.102585
```

(b) Which of the models in (a) had the smallest LOOCV error? Is this what you expected? Explain your answer.

EXERCIZE II (4 points)

Load the Boston data set. The response variable is medv.

```
library(MASS)
```

(a) Split the data set into a training set containing the first 300 observations and a test set containing the last 206 observations.

```
train = Boston[1:300,]
test = Boston[301:506,]
p = ncol(train)-1
n = nrow(train)
```

(b) Perform best subset selection on the training set, and report in **output** the training set MSE associated with the best model of each size.

```
library(leaps)
fit <- regsubsets(medv ~ .,train, nvmax=p)
MSE.tr = summary(fit)$rss/n
names(MSE.tr) <- 1:p
MSE.tr
```

```
##         1         2         3         4         5         6         7
## 15.234872 13.195994 12.221624 11.227568 10.483982 10.135679  9.995821
##         8         9        10        11        12        13
##  9.895926  9.759636  9.718253  9.674324  9.642789  9.633874
```

(c) Report in **output** the test set MSE associated with the best model of each size. For which model size does the test set MSE take on its minimum value?

```
predict.regsubsets =function(object, newdata ,id ,...){
form=as.formula(object$call[[2]])
mat=model.matrix(form, newdata)
coefi =coef(object, id=id)
xvars =names(coefi)
mat[,xvars]%*%coefi
}

MSE.te = vector()
for (k in 1:p){
yhat = predict.regsubsets(fit, newdata=test, id=k)
MSE.te[k] = mean( (yhat - test$medv)^2 )
}
names(MSE.te)<-1:p
MSE.te
```

```
##         1         2         3         4         5         6         7
##  95.71748  83.64248  78.82010  79.90052  73.61831  70.17474  66.34272
##         8         9        10        11        12        13
## 314.04493 328.55578 314.40536 391.39819 372.92974 366.06553
```

```
which.min(MSE.te)
```

```
## 7
## 7
```

EXERCIZE III (2 points)

Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of P(Class is Red|X):

$$0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75.$$

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches? Explain your answer.

```
hatp = c(0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75)
ifelse(sum(hatp>.5)>5,"red","green")
```

```
## [1] "red"
```

```
ifelse(mean(hatp)>.5,"red","green")
```

```
## [1] "green"
```