Assignment 1

INF 511

Muhammad

1 Provided

```
## Data, test hold-out, training data, validation data
data(prostate, package="faraway")
dim(prostate)
```

```
[1] 97  9
```

```
names(prostate)
```

```
[1] "lcavol"  "lweight" "age"     "lbph"    "svi"     "lcp"     "gleason"
[8] "pgg45"   "lpsa"
```

```
set.seed(20500 + 5150)
(hold.out<- sample(1:dim(prostate)[1],size=1)) ## 12th (`test') case held out
```

```
[1] 12
```

```
y<- prostate$lpsa[-hold.out] ## <-- outputs (minus 12th case)
X<- as.matrix(prostate[-hold.out,-9]) ## <-- inputs (minus 12th)
phold.out<- prostate[hold.out,,drop=FALSE] ## 12th case to `test' later
prostate<- cbind.data.frame(lpsa=y,X) ## same name! n=96 now

## Randomly choose n=72 training cases, with remaining n*=24 for
## validation.
set.seed(24601 + 711) ## Jean Valjean gets a Big Gulp
(ntot<- dim(prostate)[1])
```

```
[1] 96
```

```
(n<- ntot*0.75) ##<-- training set size
```

```
[1] 72
```

```
trainindx<- sample(x=1:ntot, size=n, replace=FALSE)
train.df<- prostate[trainindx,]
val.df<- prostate[-trainindx,]
(k<-dim(X)[2])
```

```
[1] 8
```

2 Problem 1

To solve this problem, we use the **regsubsets** function from the **leaps** library. This function fits all possible subsets of the input variables to the training data and returns the best models for each size.

Here is a step-by-step explanation of the code:

1. Load the **leaps** library:

```
library(leaps)
```

2. Fit all subsets model on the training data using the **regsubsets** function:

```
fit.full <- regsubsets(lpsa~., data=train.df, nvmax=k)
```

This line fits a linear regression model for all possible subsets of the input variables to the training data. The **lpsa** variable is the dependent variable and the **.** represents all the independent variables. The **nvmax** argument is the maximum number of variables that can be included in the model, which is set to **k**. The fitted models are stored in the **fit.full** object.

3. Initialize an empty vector to store the validation MSE values:

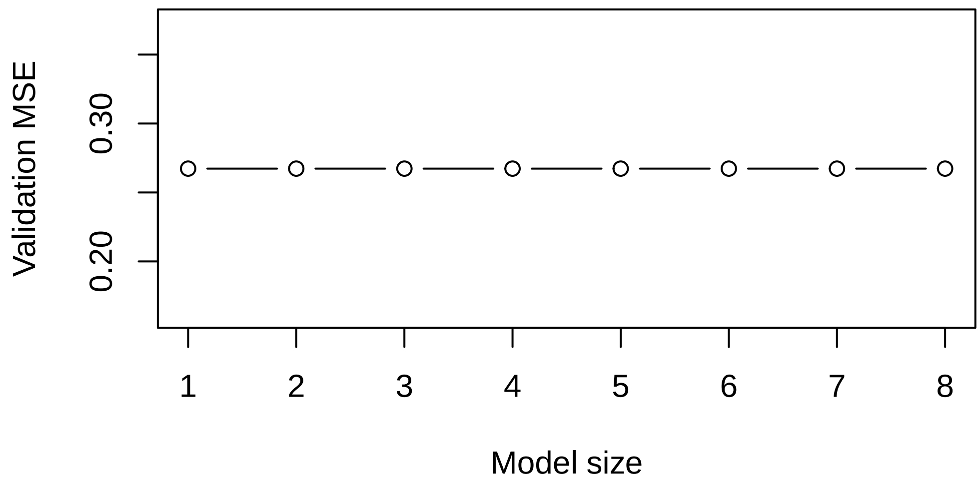
```
val.error <- rep(NA, k)
```

4. Loop over the size of the model, from 1 to **k**: In each iteration of the loop, fit a linear regression model to the validation data, using the **i**-th best subset of variables selected from the training data This line fits a linear regression model to the validation data using only the **i**-th best subset of variables selected from the training data. The **subset** argument is used to specify which variables should be included in the model.

```
for(i in 1:k){
     val.fit <- lm(lpsa~., data=val.df, subset=fit.full$which[i,])
      val.error[i] <- mean((val.fit$fitted.values - val.df$lpsa)^2)
}
```

5. This line plots the validation MSE values against the size of the model, with the size of the model on the x-axis and the validation MSE on the y-axis. The **type** argument is set to **"b"**, which means a line plot with points.

```
plot(val.error, xlab="Model size", ylab="Validation MSE", type="b")
```



6. Show the k=8 validation MSE (MSPR) values

```
val.error
```

```
[1] 0.2672908 0.2672908 0.2672908 0.2672908 0.2672908 0.2672908 0.2672908
[8] 0.2672908
```

3 Problem 2

1. Fit the best model to the entire data set

```
best_model <- regsubsets(lpsa ~ ., data = train.df, nvmax = k)
best_model_fit <- lm(lpsa ~ ., data = train.df, subset = best_model$which.min)
```

2. Create a 95% prediction interval for the hold out case

```
test_prediction <- predict(best_model_fit, newdata = phold.out, interval = "confidence", level = 0.
```

3. Output of the best model fit

```
summary(best_model_fit)
```

```
Call:
lm(formula = lpsa ~ ., data = train.df, subset = best_model$which.min)

Residuals:
     Min       1Q   Median       3Q      Max
-1.86868 -0.29416  0.04545  0.40152  1.35248

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.138222   1.514719  -0.091  0.92758
lcavol       0.678233   0.111276   6.095 7.31e-08 ***
lweight      0.373588   0.191451   1.951  0.05546 .
age         -0.022875   0.012936  -1.768  0.08185 .
lbph         0.150783   0.070842   2.128  0.03722 *
svi          0.864811   0.284875   3.036  0.00349 **
lcp         -0.142139   0.112897  -1.259  0.21267
gleason      0.221057   0.179441   1.232  0.22256
pgg45        0.002469   0.004829   0.511  0.61098
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6993 on 63 degrees of freedom
Multiple R-squared:  0.7257,    Adjusted R-squared:  0.6909
F-statistic: 20.83 on 8 and 63 DF,  p-value: 5e-15
```

4. Prediction interval

```
test_prediction
```

```
         fit        lwr      upr
12 0.5658714 0.04480372 1.086939
```