HW4_Vikas Sanil

Vikas Sanil

Due 4/20 11:59 pm

GRADING

- Part I = 20 points;
- Part II = 80 points;

Part I: Review of basic concepts in statistical learning (20 points)

You will spend some time thinking of some real-life applications for statistical learning.

Question 1.

Describe three real-life applications in which classification might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

> Answer: Three real-life applications in which classification useful.
> 1. Type of project based on number of functional requirement, number of department involved, cost of project, duration estimated, estimated hour, estimated price, previous actual time taken, previous actual hour, previous actual price. This is an inference example as project type is inferred based on past experience in similar project.
> 2. Credit worthiness(response) classification based on age, demography, job type, income and credit score(predictors).This is an inference example where credit worthiness is inferred based on similar profile of other customers.
> 3. Malware(response) classifiction based on new/emerging malwares on the basis of comparable features like delivery system, data delivered, data compramised, commnication and system control(predictors). This ia an inference example as the malware is classified based on similarity between other existing malwares.

Question 2.

Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

> Answer: Three real-life applications in which regression anlysis useful.
> 1. Weather(response) forecasting based on above ground temperature, wind, water vapour density, below ground temp, sunlight, cloud density, landscape and ocean/river water level(predictors). The goal of this is to predict weather for the future based on past data.
> 2. Time(response) required to loose certain weight based on age, gender, body mass, calories intake, calories burnt and current weight. This is a prediction example where the time required

> to loose certain weight is predicted based on data available.
> 3. Women Ovulation period(response) based on women period cycle, age, pH level and body temperature(predictors). This is a prediction example as women ovulation period is predicted on the similar set of data.

Question 3.

Describe three real-life applications in which cluster analysis might be useful.

> Answer: Three real-life applications in which cluster analysis is useful.
> 1. Advertisement(response) placement in browser based on user age, gender, demography, and past 5 browsing topics(predictors). The goal of this example is to infer an advertisement a user may like based on past online activities.
> 2. Investment product(response) suggestion to a new investment banking client based on existing customer age, gender, demography, credit score, average balance(predictors), and preferred investment product. The goal of this application is to predict the best-suited investment product for an investment banking client based on existing customer details.
> 3. Restaurant(response) suggestion in Food app. based on member age, gender, demography, and past 5 cuisine selections(predictors). The goal of this application is to infer user restaurant interest based on past choices along with user details.

Question 4.

What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

> Answer:
> - Flexible approach provides more coverage of data. Less flexible approach provides easy to understand relationship. - When inference is the goal, there are clear advantages to using simple and relatively inflexible statistical learning methods. When interested in prediction and interpretability is not required the most flexible model works.

Part II: Multiple Linear Regression (80 points)

Load the `Boston` data set

```r
# import packages
library(MASS)
library(MLmetrics)
```

```
## Warning: package 'MLmetrics' was built under R version 4.1.3
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
library(AICcmodavg)
```

```
## Warning: package 'AICcmodavg' was built under R version 4.1.3
```

```
#load data
data(Boston)
```

Exploratory data analysis (10 points)

- Check the number of observations and features using `dim`

```
str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
dim(Boston)
```

```
## [1] 506  14
```

> Answer: There are 506 observations and 14 features in Boston data set.

- Check for missing values

```
which(is.na(Boston))
```

```
## integer(0)
```

```
sum(is.na(Boston))
```

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

> Answer: There are no missing values.

- Check for duplicated values

```r
sum(duplicated(Boston))
```

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

Answer: There are no duplicated values.

- checking correlation between variables

```r
res<-cor(Boston)
round(res,2)
```

```
##          crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio
## crim     1.00 -0.20  0.41 -0.06  0.42 -0.22  0.35 -0.38  0.63  0.58    0.29
## zn      -0.20  1.00 -0.53 -0.04 -0.52  0.31 -0.57  0.66 -0.31 -0.31   -0.39
## indus    0.41 -0.53  1.00  0.06  0.76 -0.39  0.64 -0.71  0.60  0.72    0.38
## chas    -0.06 -0.04  0.06  1.00  0.09  0.09  0.09 -0.10 -0.01 -0.04   -0.12
## nox      0.42 -0.52  0.76  0.09  1.00 -0.30  0.73 -0.77  0.61  0.67    0.19
## rm      -0.22  0.31 -0.39  0.09 -0.30  1.00 -0.24  0.21 -0.21 -0.29   -0.36
## age      0.35 -0.57  0.64  0.09  0.73 -0.24  1.00 -0.75  0.46  0.51    0.26
## dis     -0.38  0.66 -0.71 -0.10 -0.77  0.21 -0.75  1.00 -0.49 -0.53   -0.23
## rad      0.63 -0.31  0.60 -0.01  0.61 -0.21  0.46 -0.49  1.00  0.91    0.46
## tax      0.58 -0.31  0.72 -0.04  0.67 -0.29  0.51 -0.53  0.91  1.00    0.46
## ptratio  0.29 -0.39  0.38 -0.12  0.19 -0.36  0.26 -0.23  0.46  0.46    1.00
## black   -0.39  0.18 -0.36  0.05 -0.38  0.13 -0.27  0.29 -0.44 -0.44   -0.18
## lstat    0.46 -0.41  0.60 -0.05  0.59 -0.61  0.60 -0.50  0.49  0.54    0.37
## medv    -0.39  0.36 -0.48  0.18 -0.43  0.70 -0.38  0.25 -0.38 -0.47   -0.51
##         black lstat  medv
## crim    -0.39  0.46 -0.39
## zn       0.18 -0.41  0.36
## indus   -0.36  0.60 -0.48
## chas     0.05 -0.05  0.18
## nox     -0.38  0.59 -0.43
## rm       0.13 -0.61  0.70
## age     -0.27  0.60 -0.38
## dis      0.29 -0.50  0.25
## rad     -0.44  0.49 -0.38
## tax     -0.44  0.54 -0.47
## ptratio -0.18  0.37 -0.51
## black    1.00 -0.37  0.33
## lstat   -0.37  1.00 -0.74
## medv     0.33 -0.74  1.00
```

Answer: tax~ rad-> 0.91 has highest correlation between variables.

Split data set into 80:20 train and test data with name `BostonTraining` and `BostonTesting` respectively (10 points)

```r
i <- sample(2, nrow(Boston), replace=TRUE, prob=c(0.8, 0.2))
BostonTraining <- Boston[i==1,]
BostonTest <- Boston[i==2,]
```

Subset Selection Linear Regression Model

Forward Stepwise (25 points)

- Please construct a forward stepwise regression with `BostonTraining`.

```
#null model
intercept_only<-lm(medv~ 1, data=BostonTraining)
#  full model
all<-lm(medv~., data = BostonTraining)
# forward set-wise regression
forward<- stepAIC(intercept_only, direction='forward', scope=formula(all))
```

```
## Start:  AIC=1801.63
## medv ~ 1
##
##           Df Sum of Sq   RSS    AIC
## + lstat    1   18719.5 15736 1486.2
## + rm       1   15403.5 19052 1563.7
## + ptratio  1    8485.0 25970 1689.1
## + indus    1    7989.7 26465 1696.8
## + tax      1    6974.6 27481 1712.0
## + nox      1    6446.5 28009 1719.7
## + age      1    5214.9 29240 1737.2
## + crim     1    5139.1 29316 1738.2
## + zn       1    5123.0 29332 1738.4
## + rad      1    4755.6 29699 1743.5
## + black    1    3762.0 30693 1756.8
## + dis      1    2459.2 31996 1773.6
## + chas     1    1234.3 33221 1788.8
## <none>                 34455 1801.6
##
## Step:  AIC=1486.22
## medv ~ lstat
##
##           Df Sum of Sq   RSS    AIC
## + rm       1   2794.18 12941 1409.0
## + ptratio  1   2128.82 13607 1429.3
## + chas     1    728.34 15007 1469.0
## + dis      1    609.51 15126 1472.2
## + age      1    324.40 15411 1479.8
## + zn       1    236.18 15499 1482.1
## + tax      1    198.58 15537 1483.1
## + black    1    153.73 15582 1484.2
## + crim     1    150.49 15585 1484.3
## <none>                 15736 1486.2
## + indus    1     71.54 15664 1486.4
## + rad      1     22.02 15714 1487.7
## + nox      1      5.53 15730 1488.1
##
## Step:  AIC=1409.04
## medv ~ lstat + rm
##
```

```
##           Df Sum of Sq   RSS    AIC
## + ptratio  1   1366.10 11575 1365.9
## + chas     1    552.49 12389 1393.4
## + dis      1    340.12 12601 1400.2
## + black    1    296.27 12645 1401.7
## + tax      1    282.60 12659 1402.1
## + crim     1    270.54 12671 1402.5
## + rad      1    112.62 12829 1407.5
## + zn       1     86.25 12855 1408.3
## <none>                 12941 1409.0
## + age      1     59.53 12882 1409.2
## + indus    1     29.87 12912 1410.1
## + nox      1      2.80 12939 1411.0
##
## Step:  AIC=1365.86
## medv ~ lstat + rm + ptratio
##
##         Df Sum of Sq   RSS    AIC
## + dis    1    533.31 11042 1348.8
## + chas   1    376.53 11199 1354.5
## + black  1    219.63 11356 1360.1
## + age    1    155.07 11420 1362.4
## + crim   1    100.04 11475 1364.3
## <none>               11575 1365.9
## + rad    1     22.08 11553 1367.1
## + indus  1     12.17 11563 1367.4
## + tax    1     10.17 11565 1367.5
## + zn     1      3.76 11572 1367.7
## + nox    1      1.68 11574 1367.8
##
## Step:  AIC=1348.76
## medv ~ lstat + rm + ptratio + dis
##
##         Df Sum of Sq   RSS    AIC
## + nox    1    534.47 10508 1330.7
## + black  1    299.70 10742 1339.6
## + chas   1    254.89 10787 1341.3
## + zn     1    207.54 10834 1343.1
## + crim   1    203.40 10839 1343.2
## + indus  1    181.04 10861 1344.1
## + tax    1    140.90 10901 1345.6
## <none>               11042 1348.8
## + age    1      8.77 11033 1350.4
## + rad    1      6.97 11035 1350.5
##
## Step:  AIC=1330.66
## medv ~ lstat + rm + ptratio + dis + nox
##
##         Df Sum of Sq   RSS    AIC
## + chas   1   270.293 10237 1322.1
## + zn     1   205.451 10302 1324.7
## + black  1   174.916 10333 1325.9
## + crim   1   136.579 10371 1327.4
## + rad    1    55.347 10452 1330.5
```

```
## <none>              10508 1330.7
## + indus  1    18.387 10489 1332.0
## + age    1    10.950 10497 1332.2
## + tax    1     2.326 10505 1332.6
##
## Step:  AIC=1322.11
## medv ~ lstat + rm + ptratio + dis + nox + chas
##
##         Df Sum of Sq   RSS    AIC
## + zn     1   214.100 10023 1315.5
## + black  1   144.557 10093 1318.3
## + crim   1   116.183 10121 1319.5
## + rad    1    57.572 10180 1321.8
## <none>              10237 1322.1
## + indus  1    24.419 10213 1323.1
## + age    1     4.857 10232 1323.9
## + tax    1     0.224 10237 1324.1
##
## Step:  AIC=1315.55
## medv ~ lstat + rm + ptratio + dis + nox + chas + zn
##
##         Df Sum of Sq     RSS    AIC
## + crim   1   178.103  9845.0 1310.3
## + black  1   168.994  9854.1 1310.7
## <none>              10023.1 1315.5
## + indus  1    28.293  9994.8 1316.4
## + rad    1    24.162  9999.0 1316.6
## + tax    1    20.914 10002.2 1316.7
## + age    1    18.584 10004.5 1316.8
##
## Step:  AIC=1310.29
## medv ~ lstat + rm + ptratio + dis + nox + chas + zn + crim
##
##         Df Sum of Sq    RSS    AIC
## + rad    1   140.262 9704.8 1306.5
## + black  1   107.983 9737.0 1307.8
## <none>              9845.0 1310.3
## + indus  1    28.624 9816.4 1311.1
## + age    1    13.468 9831.6 1311.7
## + tax    1     0.004 9845.0 1312.3
##
## Step:  AIC=1306.48
## medv ~ lstat + rm + ptratio + dis + nox + chas + zn + crim +
##     rad
##
##         Df Sum of Sq    RSS    AIC
## + tax    1   242.367 9462.4 1298.2
## + black  1   158.737 9546.0 1301.8
## + indus  1    50.372 9654.4 1306.4
## <none>              9704.8 1306.5
## + age    1    23.665 9681.1 1307.5
##
## Step:  AIC=1298.23
## medv ~ lstat + rm + ptratio + dis + nox + chas + zn + crim +
```

```
##     rad + tax
##
##         Df Sum of Sq    RSS    AIC
## + black  1   150.473 9311.9 1293.7
## <none>               9462.4 1298.2
## + age    1    27.321 9435.1 1299.1
## + indus  1     1.640 9460.8 1300.2
##
## Step:  AIC=1293.74
## medv ~ lstat + rm + ptratio + dis + nox + chas + zn + crim +
##     rad + tax + black
##
##         Df Sum of Sq    RSS    AIC
## <none>               9311.9 1293.7
## + age    1   20.8137 9291.1 1294.8
## + indus  1    0.8559 9311.1 1295.7
```

```
#results
forward$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## medv ~ 1
##
## Final Model:
## medv ~ lstat + rm + ptratio + dis + nox + chas + zn + crim +
##     rad + tax + black
##
##
##         Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                                404  34455.110 1801.628
## 2    + lstat  1 18719.5313       403  15735.579 1486.216
## 3       + rm  1  2794.1833       402  12941.395 1409.041
## 4  + ptratio  1  1366.0974       401  11575.298 1365.860
## 5      + dis  1   533.3059       400  11041.992 1348.757
## 6      + nox  1   534.4668       399  10507.525 1330.664
## 7     + chas  1   270.2925       398  10237.233 1322.109
## 8       + zn  1   214.0998       397  10023.133 1315.549
## 9     + crim  1   178.1031       396   9845.030 1310.288
## 10     + rad  1   140.2624       395   9704.767 1306.477
## 11     + tax  1   242.3672       394   9462.400 1298.234
## 12   + black  1   150.4725       393   9311.928 1293.742
```

```
summary(forward)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm + ptratio + dis + nox + chas +
##     zn + crim + rad + tax + black, data = BostonTraining)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -14.8934  -2.9053  -0.6372   1.7724  24.6095
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.793496   5.746110   7.099 5.90e-12 ***
## lstat        -0.594622   0.053393 -11.137  < 2e-16 ***
## rm            3.189183   0.452213   7.052 7.97e-12 ***
## ptratio      -0.926650   0.151362  -6.122 2.24e-09 ***
## dis          -1.581055   0.212191  -7.451 5.94e-13 ***
## nox         -15.874519   4.054231  -3.916 0.000106 ***
## chas          2.678040   0.984937   2.719 0.006838 **
## zn            0.058085   0.015846   3.666 0.000281 ***
## crim         -0.114696   0.034981  -3.279 0.001135 **
## rad           0.308432   0.073009   4.225 2.98e-05 ***
## tax          -0.012243   0.003895  -3.143 0.001798 **
## black         0.007739   0.003071   2.520 0.012130 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.868 on 393 degrees of freedom
## Multiple R-squared:  0.7297, Adjusted R-squared:  0.7222
## F-statistic: 96.47 on 11 and 393 DF,  p-value: < 2.2e-16
```

> Answer: Final forward model with optimal set of features is lm(medv~lstat+rm+ptratio+dis+nox+chas+black+zn+crim
> data=BostonTraining)

- Use this model to predict `medv` in `BostonTesting` and calculate `MAE` and `MSE`.

```r
ypred_forward<-predict(object = forward, newdata=BostonTest)
MAE(y_pred = ypred_forward, y_true = BostonTest$medv)
```

```
## [1] 3.050796
```

```r
MSE(y_pred = ypred_forward, y_true = BostonTest$medv)
```

```
## [1] 18.38295
```

> Answer:
> - The MAE is 3.28 and MSE is 17.40 for forward model on BostonTraining data set.
> - The forecast distance between true value will be 3.28. - The model is not perfect.

Backward Stepwise (25 points)

- Please construct a backward stepwise regression with `BostonTraining`.

```r
# backward set-wise regression
backward<- stepAIC(all, direction='backward')
```

```
## Start:  AIC=1296.8
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq     RSS    AIC
## - indus    1      0.88  9291.1 1294.8
## - age      1     20.84  9311.1 1295.7
## <none>                  9290.2 1296.8
## - black    1    143.19  9433.4 1301.0
## - chas     1    167.74  9458.0 1302.0
## - tax      1    193.23  9483.5 1303.1
## - crim     1    256.56  9546.8 1305.8
## - zn       1    329.03  9619.3 1308.9
## - nox      1    349.98  9640.2 1309.8
## - rad      1    403.53  9693.8 1312.0
## - ptratio  1    881.98 10172.2 1331.5
## - rm       1   1035.17 10325.4 1337.6
## - dis      1   1042.81 10333.0 1337.9
## - lstat    1   2674.56 11964.8 1397.3
##
## Step:  AIC=1294.84
## medv ~ crim + zn + chas + nox + rm + age + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq     RSS    AIC
## - age      1     20.81  9311.9 1293.7
## <none>                  9291.1 1294.8
## - black    1    143.97  9435.1 1299.1
## - chas     1    166.88  9458.0 1300.0
## - tax      1    237.38  9528.5 1303.0
## - crim     1    255.68  9546.8 1303.8
## - zn       1    333.55  9624.7 1307.1
## - nox      1    383.28  9674.4 1309.2
## - rad      1    434.48  9725.6 1311.3
## - ptratio  1    905.89 10197.0 1330.5
## - rm       1   1054.70 10345.8 1336.4
## - dis      1   1096.67 10387.8 1338.0
## - lstat    1   2699.14 11990.3 1396.1
##
## Step:  AIC=1293.74
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq     RSS    AIC
## <none>                  9311.9 1293.7
## - black    1    150.47  9462.4 1298.2
## - chas     1    175.17  9487.1 1299.3
## - tax      1    234.10  9546.0 1301.8
## - crim     1    254.73  9566.7 1302.7
## - zn       1    318.37  9630.3 1305.4
## - nox      1    363.27  9675.2 1307.2
## - rad      1    422.88  9734.8 1309.7
## - ptratio  1    888.07 10200.0 1328.6
## - rm       1   1178.47 10490.4 1340.0
```

```
## - dis      1   1315.49 10627.4 1345.3
## - lstat    1   2938.70 12250.6 1402.8
```

```r
#results
backward$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
## Final Model:
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##
##      Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                             391   9290.234 1296.797
## 2 - indus  1  0.8802079       392   9291.114 1294.835
## 3   - age  1 20.8137268       393   9311.928 1293.742
```

```r
summary(backward)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat, data = BostonTraining)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.8934  -2.9053  -0.6372   1.7724  24.6095
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.793496   5.746110   7.099 5.90e-12 ***
## crim         -0.114696   0.034981  -3.279 0.001135 **
## zn            0.058085   0.015846   3.666 0.000281 ***
## chas          2.678040   0.984937   2.719 0.006838 **
## nox         -15.874519   4.054231  -3.916 0.000106 ***
## rm            3.189183   0.452213   7.052 7.97e-12 ***
## dis          -1.581055   0.212191  -7.451 5.94e-13 ***
## rad           0.308432   0.073009   4.225 2.98e-05 ***
## tax          -0.012243   0.003895  -3.143 0.001798 **
## ptratio      -0.926650   0.151362  -6.122 2.24e-09 ***
## black         0.007739   0.003071   2.520 0.012130 *
## lstat        -0.594622   0.053393 -11.137  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.868 on 393 degrees of freedom
## Multiple R-squared:  0.7297, Adjusted R-squared:  0.7222
## F-statistic: 96.47 on 11 and 393 DF,  p-value: < 2.2e-16
```

Answer: Final forward model with optimal set of features is lm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+
data=BostonTraining)

- Use this model to predict `medv` in `BostonTesting` and calculate `MAE` and `MSE`.

```
ypred_backward<-predict(object = backward, newdata=BostonTest)
MAE(y_pred = ypred_backward, y_true = BostonTest$medv)
```

```
## [1] 3.050796
```

```
MSE(y_pred = ypred_backward, y_true = BostonTest$medv)
```

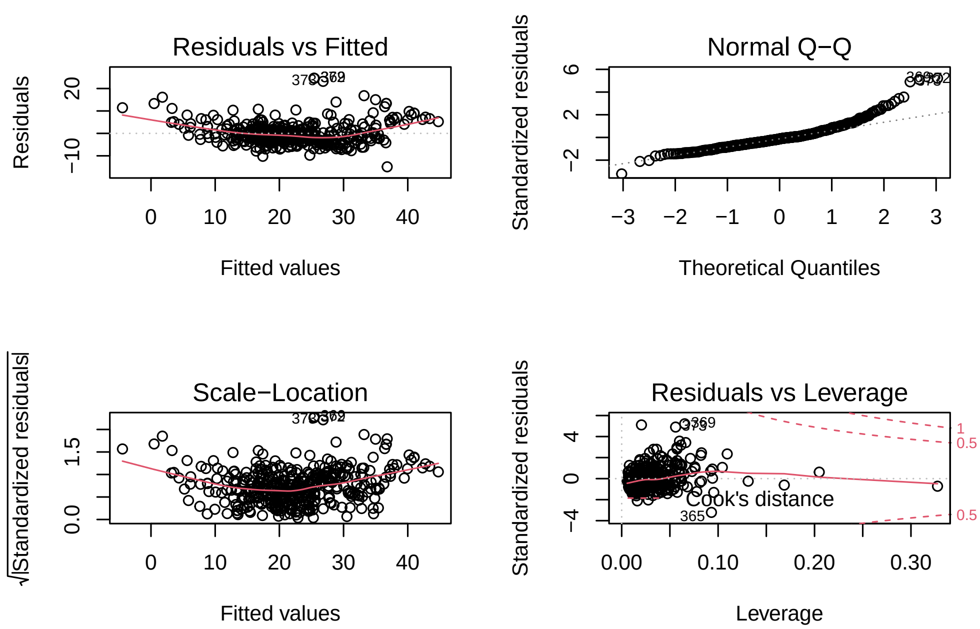
```
## [1] 18.38295
```

Answer:
- The MAE is 3.28 and MSE is 17.40 for forward model on BostonTraining data set.
- The forecast distance between true value will be 3.28. - The model is not perfect.

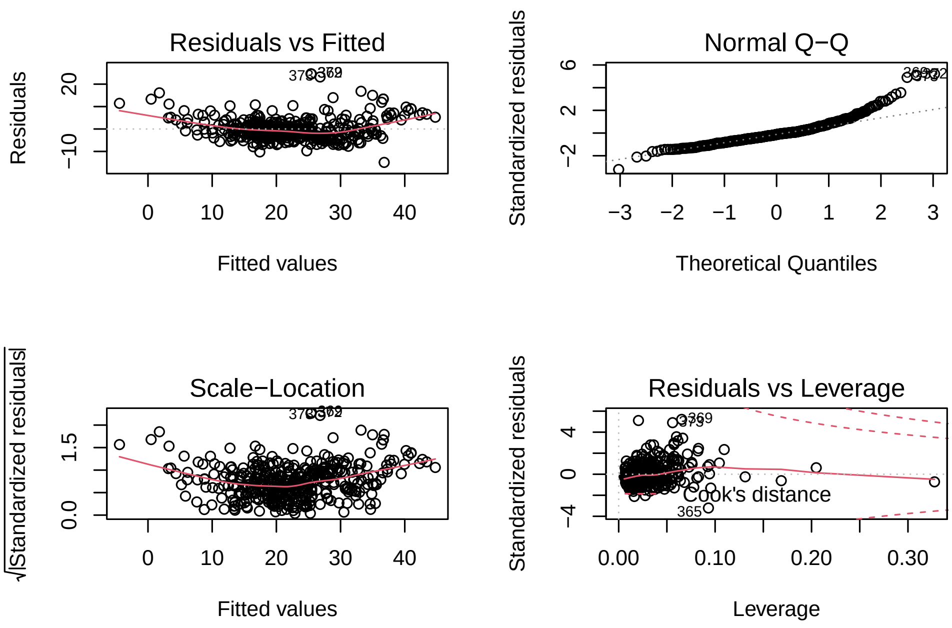
Model Assessment (10 points)

Compare the forward and backward stepwise linear regression models. You can use plots, assessment measures (R^2, RSS, MAE, MSE,etc.) Which one is better? Explain your answer.

```
par(mfrow=c(2,2))
plot(forward)
```



```r
par(mfrow=c(2,2))
plot(backward)
```




```r
par(mfrow=c(1,2))
summary(forward)$coefficients
```

```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)  40.793496168 5.746110297   7.099324 5.904287e-12
## lstat        -0.594622375 0.053393314 -11.136645 3.192959e-25
## rm            3.189182549 0.452212765   7.052394 7.971428e-12
## ptratio      -0.926649725 0.151361937  -6.122079 2.239230e-09
## dis          -1.581054699 0.212190647  -7.451105 5.941305e-13
## nox         -15.874518672 4.054230637  -3.915544 1.063254e-04
## chas          2.678039869 0.984936514   2.718997 6.838244e-03
## zn            0.058084572 0.015845854   3.665601 2.806545e-04
## crim         -0.114696039 0.034980859  -3.278823 1.135284e-03
## rad           0.308431834 0.073008522   4.224600 2.978085e-05
## tax          -0.012242999 0.003895009  -3.143253 1.797613e-03
## black         0.007739315 0.003071125   2.520026 1.212999e-02
```

```r
summary(backward)$coefficients
```

```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)  40.793496168 5.746110297   7.099324 5.904287e-12
## crim         -0.114696039 0.034980859  -3.278823 1.135284e-03
## zn            0.058084572 0.015845854   3.665601 2.806545e-04
## chas          2.678039869 0.984936514   2.718997 6.838244e-03
## nox         -15.874518672 4.054230637  -3.915544 1.063254e-04
## rm            3.189182549 0.452212765   7.052394 7.971428e-12
```

```
## dis          -1.581054699 0.212190647  -7.451105 5.941305e-13
## rad           0.308431834 0.073008522   4.224600 2.978085e-05
## tax          -0.012242999 0.003895009  -3.143253 1.797613e-03
## ptratio      -0.926649725 0.151361937  -6.122079 2.239230e-09
## black         0.007739315 0.003071125   2.520026 1.212999e-02
## lstat        -0.594622375 0.053393314 -11.136645 3.192959e-25
```

```r
anova(forward,backward)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat + rm + ptratio + dis + nox + chas + zn + crim +
##     rad + tax + black
## Model 2: medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##   Res.Df    RSS Df  Sum of Sq F Pr(>F)
## 1    393 9311.9
## 2    393 9311.9  0 -1.819e-12
```

```r
modelset<-list(forward,backward)
aictab(modelset)
```

```
## Warning in aictab.AIClm(modelset):
## Model names have been supplied automatically in the table
```

```
## Warning in aictab.AIClm(modelset):
## Check model structure carefully as some models may be redundant
```

```
##
## Model selection based on AICc:
##
##       K    AICc Delta_AICc AICcWt Cum.Wt       LL
## Mod1 13 2446.01          0    0.5    0.5 -1209.54
## Mod2 13 2446.01          0    0.5    1.0 -1209.54
```

Answer: Both model have same result. Hence both are better.