CaydenDunn_hw1

2022-09-28

```
#get data
b_housing <- read.csv("/Users/cindydunn/Desktop/Grad_School/math540/hw1/data/BostonHousing.csv")
attach(b_housing)

## Include the functions required for data partitioning
source("/Users/cindydunn/Desktop/Grad_School/math540/Variable_Selection/myfunctions copy.r")

#remove the CAT.MDEV variable
b_housing <- b_housing[,-14]

# Create training validation test data
RNGkind (sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
set.seed(123) ## set seed so that you get same partition each time
p3 <- partition.3(b_housing, 0.6, 0.3) ## creating 60:30:10 partition
training.data <- p3$data.train # training data
validation.data <- p3$data.val # validation data
test.data <- p3$data.test # test data
```

Part 1

#q1 Why should the data be partitioned into training and test sets? What will the training set be used for? What will the test set be used for? The data should be partitioned into training and test sets to ensure that the model is not overfitting the data. The training set will be used to fit the model and the test set will be used to evaluate the model.

#q2 Fit a multiple linear regression model to the median house price (MEDV) as a function of CRIM, CHAS and RM. Write the equation for predicting the median house price using the predictors in the model.

```
fit1 <- lm(MEDV ~ CRIM + CHAS + RM, data = training.data)
summary(fit1)
```

```
##
## Call:
## lm(formula = MEDV ~ CRIM + CHAS + RM, data = training.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.657  -3.126  -0.502   2.539  38.234
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -27.31159    3.46259  -7.888 5.78e-14 ***
## CRIM         -0.25283    0.05114  -4.943 1.28e-06 ***
```

```
## CHAS          5.15182    1.56354   3.295   0.0011 **
## RM            8.11180    0.54081  14.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.394 on 300 degrees of freedom
## Multiple R-squared:  0.5455, Adjusted R-squared:  0.5409
## F-statistic:   120 on 3 and 300 DF,  p-value: < 2.2e-16
```

#q3 Using the estimated regression model, what median house price is predicted for a tract in the Boston area that does not bound the Charles river, has crime rate of 0.1, and where the average number of rooms per house is 6?

```
predict(fit1, data.frame(CRIM = 0.1, CHAS = 0, RM = 6))
```

```
##       1
## 21.3339
```

#q4 Fit a linear regression model with all 12 predictors.

```
fit2 <- lm(MEDV ~ ., data = training.data)
```

#sub1 Report the Root mean square error of this model on test data.

```
yhat_test = predict(fit2,test.data)
RMSE_test = sqrt(mean((test.data$MEDV - yhat_test)^2))
RMSE_test
```

```
## [1] 4.422687
```

Is multicollinearity a potential problem for this model?

```
library(car)
```

```
## Loading required package: carData
```

```
vif(fit2)
```

```
##     CRIM       ZN    INDUS     CHAS      NOX       RM      AGE      DIS
## 2.065763 2.205617 3.823512 1.056561 4.471157 1.924898 3.289721 4.074718
##      RAD      TAX  PTRATIO    LSTAT
## 7.462613 8.752755 1.866815 2.888292
```

Generally, VIF value > 4 is a matter of concern (VIF > 10 is definitely a matter of concern) There are several predictors with VIF > 4, so multicollinearity is a potential problem for this model.

Compute the correlation table for the numerical predictors and search for highly correlated pairs.

```
library(data.table)
corMatrix <- cor(matrix(rnorm(100), 5))
#look through the correlation table and find the highest pairwise correlation
corList <- setDT(melt(cor(training.data)))[order(value)]
```

```
## Warning in melt(cor(training.data)): The melt generic in data.table has
## been passed a matrix and will attempt to redirect to the relevant reshape2
## method; please note that reshape2 is deprecated, and this redirection is now
## deprecated as well. To continue using melt methods from reshape2 while both
## libraries are attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(cor(training.data)). In the next version, this warning will
## become an error.
```

```
#send an email to teacher what is cutoff value for high correlation
#make a new subset of the data with only the highly correlated predictors
highCorr <- corList[abs(value) > 0.6]
#remove any rows where the predictors are the same
#remove any rows where one of the predictors is MEDV
highCorr <- highCorr[Var1 != Var2]
highCorr <- highCorr[Var1 != "MEDV"]
#remove any rows where the column 'value' is the same as another row
highCorr <- highCorr[!duplicated(value)]
highCorr
```

```
##      Var1  Var2      value
##  1:   DIS   AGE -0.7788702
##  2:   DIS   NOX -0.7756054
##  3: LSTAT  MEDV -0.7380642
##  4:   DIS INDUS -0.7026881
##  5: LSTAT    RM -0.6299878
##  6: LSTAT   NOX  0.6064998
##  7:   RAD   NOX  0.6159477
##  8: LSTAT INDUS  0.6172093
##  9:   TAX  CRIM  0.6173081
## 10:   AGE INDUS  0.6456379
## 11:   DIS    ZN  0.6474437
## 12:   RAD  CRIM  0.6690403
## 13:   TAX   NOX  0.6744148
## 14:    RM  MEDV  0.7009853
## 15:   TAX INDUS  0.7052320
## 16:   AGE   NOX  0.7430835
## 17:   NOX INDUS  0.7456485
## 18:   TAX   RAD  0.9031300
```

#sub2 Use stepwise regression with cross validation approach to reduce the number of predictors. How many variables do you have in the final model? Which variables are dropped? Report the RMSE of this model on test data.

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
train_control <- trainControl(method = "cv", number = 5) # 5-fold cross validation
step_kcv <- train(MEDV ~ ., data = training.data, method = "lmStepAIC", trControl = train_control) # st
```

```
## Start:  AIC=811.83
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS +
##     RAD + TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - AGE      1      1.55 6168.4 809.89
## - INDUS    1     22.50 6189.4 810.72
## - CRIM     1     49.70 6216.5 811.78
## <none>                 6166.8 811.83
## - ZN       1     62.18 6229.0 812.27
## - TAX      1    128.49 6295.3 814.84
## - CHAS     1    240.90 6407.7 819.14
```

```
## - RAD      1    245.05 6411.9 819.30
## - NOX      1    263.03 6429.9 819.98
## - DIS      1    451.91 6618.8 827.02
## - PTRATIO  1    460.54 6627.4 827.33
## - RM       1    995.67 7162.5 846.20
## - LSTAT    1   1415.16 7582.0 860.03
##
## Step:  AIC=809.89
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + DIS + RAD +
##     TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - INDUS    1     21.81 6190.2 808.75
## - CRIM     1     49.90 6218.3 809.85
## <none>                 6168.4 809.89
## - ZN       1     60.80 6229.2 810.28
## - TAX      1    126.98 6295.4 812.85
## - CHAS     1    242.22 6410.6 817.25
## - RAD      1    243.75 6412.2 817.31
## - NOX      1    270.24 6438.6 818.31
## - PTRATIO  1    461.44 6629.8 825.42
## - DIS      1    545.63 6714.0 828.49
## - RM       1   1070.21 7238.6 846.77
## - LSTAT    1   1511.62 7680.0 861.16
##
## Step:  AIC=808.75
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
##     LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - ZN       1     48.84 6239.1 808.66
## <none>                 6190.2 808.75
## - CRIM     1     52.03 6242.2 808.79
## - TAX      1    107.46 6297.7 810.93
## - RAD      1    222.47 6412.7 815.33
## - NOX      1    248.77 6439.0 816.33
## - CHAS     1    257.96 6448.2 816.67
## - PTRATIO  1    442.14 6632.4 823.52
## - DIS      1    597.98 6788.2 829.16
## - RM       1   1054.93 7245.1 844.99
## - LSTAT    1   1491.21 7681.4 859.20
##
## Step:  AIC=808.66
## .outcome ~ CRIM + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
##     LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - CRIM     1     39.47 6278.5 808.19
## <none>                 6239.1 808.66
## - TAX      1     78.15 6317.2 809.69
## - RAD      1    195.11 6434.2 814.14
## - CHAS     1    271.39 6510.4 817.01
## - NOX      1    280.68 6519.7 817.36
## - DIS      1    582.93 6822.0 828.37
```

```
## - PTRATIO  1    632.58 6871.6 830.13
## - RM       1   1172.04 7411.1 848.49
## - LSTAT    1   1468.38 7707.4 858.02
##
## Step:  AIC=808.19
## .outcome ~ CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## <none>                 6278.5 808.19
## - TAX      1     74.03 6352.6 809.04
## - RAD      1    155.65 6434.2 812.15
## - NOX      1    271.83 6550.4 816.49
## - CHAS     1    285.44 6564.0 817.00
## - DIS      1    564.70 6843.2 827.12
## - PTRATIO  1    607.89 6886.4 828.65
## - RM       1   1232.02 7510.5 849.73
## - LSTAT    1   1607.72 7886.2 861.59
## Start:  AIC=771.96
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS +
##     RAD + TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - AGE      1      0.05 5278.7 769.96
## - INDUS    1      0.45 5279.1 769.98
## <none>                 5278.6 771.96
## - ZN       1     46.19 5324.8 772.07
## - CHAS     1     52.67 5331.3 772.36
## - CRIM     1     81.19 5359.8 773.65
## - TAX      1    105.77 5384.4 774.76
## - NOX      1    263.53 5542.1 781.75
## - RAD      1    263.72 5542.3 781.76
## - PTRATIO  1    425.50 5704.1 788.72
## - DIS      1    558.31 5836.9 794.29
## - RM       1   1062.74 6341.3 814.35
## - LSTAT    1   1485.23 6763.8 829.96
##
## Step:  AIC=769.96
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + DIS + RAD +
##     TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - INDUS    1      0.43 5279.1 767.98
## <none>                 5278.7 769.96
## - ZN       1     46.35 5325.0 770.08
## - CHAS     1     52.69 5331.3 770.37
## - CRIM     1     81.15 5359.8 771.65
## - TAX      1    106.08 5384.7 772.78
## - RAD      1    266.69 5545.3 779.89
## - NOX      1    293.60 5572.3 781.06
## - PTRATIO  1    436.49 5715.1 787.19
## - DIS      1    649.32 5928.0 796.04
## - RM       1   1100.54 6379.2 813.79
## - LSTAT    1   1623.02 6901.7 832.84
##
```

```
## Step:  AIC=767.98
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
##     LSTAT
##
##           Df Sum of Sq    RSS    AIC
## <none>                 5279.1 767.98
## - ZN       1     47.60 5326.7 768.15
## - CHAS     1     52.26 5331.3 768.37
## - CRIM     1     80.76 5359.9 769.66
## - TAX      1    134.98 5414.1 772.09
## - RAD      1    300.74 5579.8 779.39
## - NOX      1    325.15 5604.2 780.45
## - PTRATIO  1    454.51 5733.6 785.97
## - DIS      1    673.84 5952.9 795.05
## - RM       1   1115.00 6394.1 812.35
## - LSTAT    1   1662.02 6941.1 832.22
## Start:  AIC=814.79
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS +
##     RAD + TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - INDUS    1      0.05 6185.2 812.79
## - AGE      1      0.52 6185.6 812.81
## <none>                 6185.1 814.79
## - CRIM     1    104.05 6289.2 816.86
## - TAX      1    146.49 6331.6 818.50
## - ZN       1    174.40 6359.5 819.57
## - NOX      1    199.99 6385.1 820.55
## - CHAS     1    206.90 6392.0 820.82
## - RAD      1    305.00 6490.1 824.53
## - PTRATIO  1    551.33 6736.5 833.62
## - DIS      1    573.01 6758.1 834.41
## - RM       1    668.12 6853.2 837.82
## - LSTAT    1   1848.92 8034.0 876.60
##
## Step:  AIC=812.79
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + AGE + DIS + RAD + TAX +
##     PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - AGE      1      0.51 6185.7 810.81
## <none>                 6185.2 812.79
## - CRIM     1    104.08 6289.3 814.86
## - ZN       1    176.84 6362.0 817.67
## - TAX      1    180.69 6365.9 817.81
## - CHAS     1    207.56 6392.7 818.84
## - NOX      1    212.42 6397.6 819.03
## - RAD      1    336.34 6521.5 823.71
## - PTRATIO  1    561.20 6746.4 831.98
## - DIS      1    598.81 6784.0 833.34
## - RM       1    676.60 6861.8 836.12
## - LSTAT    1   1881.41 8066.6 875.59
##
## Step:  AIC=810.81
```

```
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
##     LSTAT
##
##           Df Sum of Sq    RSS    AIC
## <none>                 6185.7 810.81
## - CRIM     1    103.89 6289.6 812.87
## - ZN       1    179.34 6365.0 815.78
## - TAX      1    182.75 6368.4 815.91
## - CHAS     1    207.71 6393.4 816.87
## - NOX      1    229.44 6415.1 817.70
## - RAD      1    340.49 6526.2 821.88
## - PTRATIO  1    563.90 6749.6 830.10
## - RM       1    694.91 6880.6 834.79
## - DIS      1    705.17 6890.9 835.15
## - LSTAT    1   2035.07 8220.8 878.21
## Start:  AIC=795.06
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS +
##     RAD + TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - AGE      1      0.74 5705.6 793.10
## - INDUS    1     10.02 5714.9 793.49
## <none>                 5704.8 795.06
## - CRIM     1     51.91 5756.7 795.27
## - ZN       1    104.49 5809.3 797.49
## - TAX      1    113.27 5818.1 797.86
## - RAD      1    193.27 5898.1 801.19
## - NOX      1    285.14 5990.0 804.96
## - CHAS     1    413.95 6118.8 810.16
## - PTRATIO  1    483.13 6188.0 812.90
## - DIS      1    628.47 6333.3 818.56
## - RM       1   1137.19 6842.0 837.42
## - LSTAT    1   1216.65 6921.5 840.23
##
## Step:  AIC=793.1
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + DIS + RAD +
##     TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - INDUS    1      9.85 5715.4 791.52
## <none>                 5705.6 793.10
## - CRIM     1     51.72 5757.3 793.30
## - ZN       1    109.55 5815.1 795.74
## - TAX      1    116.21 5821.8 796.01
## - RAD      1    198.45 5904.0 799.44
## - NOX      1    309.17 6014.7 803.97
## - CHAS     1    413.23 6118.8 808.16
## - PTRATIO  1    486.00 6191.6 811.04
## - DIS      1    693.35 6398.9 819.08
## - RM       1   1191.81 6897.4 837.38
## - LSTAT    1   1343.31 7048.9 842.68
##
## Step:  AIC=791.52
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
```

```
##     LSTAT
##
##           Df Sum of Sq    RSS    AIC
## <none>                 5715.4 791.52
## - CRIM     1     54.35 5769.8 791.83
## - ZN       1    102.11 5817.5 793.84
## - TAX      1    112.49 5827.9 794.27
## - RAD      1    192.50 5907.9 797.60
## - NOX      1    300.87 6016.3 802.03
## - CHAS     1    418.30 6133.7 806.75
## - PTRATIO  1    476.71 6192.1 809.06
## - DIS      1    751.33 6466.8 819.65
## - RM       1   1183.77 6899.2 835.45
## - LSTAT    1   1341.36 7056.8 840.96
## Start:  AIC=808.78
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS +
##     RAD + TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - INDUS    1      0.01 6089.9 806.78
## - AGE      1     16.41 6106.3 807.44
## <none>                 6089.9 808.78
## - CRIM     1     54.64 6144.5 808.95
## - TAX      1     83.52 6173.4 810.09
## - ZN       1    159.55 6249.5 813.07
## - NOX      1    223.71 6313.6 815.55
## - RAD      1    239.57 6329.5 816.16
## - CHAS     1    353.66 6443.6 820.50
## - PTRATIO  1    511.70 6601.6 826.39
## - DIS      1    515.09 6605.0 826.51
## - RM       1    735.27 6825.2 834.48
## - LSTAT    1   1928.65 8018.6 873.64
##
## Step:  AIC=806.78
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + AGE + DIS + RAD + TAX +
##     PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS    AIC
## - AGE      1     16.41 6106.3 805.44
## <none>                 6089.9 806.78
## - CRIM     1     54.85 6144.8 806.96
## - TAX      1     99.50 6189.4 808.72
## - ZN       1    163.62 6253.5 811.23
## - NOX      1    238.37 6328.3 814.11
## - RAD      1    255.41 6345.3 814.77
## - CHAS     1    357.95 6447.9 818.66
## - PTRATIO  1    517.93 6607.8 824.62
## - DIS      1    531.93 6621.8 825.13
## - RM       1    739.46 6829.4 832.63
## - LSTAT    1   1949.37 8039.3 872.27
##
## Step:  AIC=805.44
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
##     LSTAT
```

```
##
##           Df Sum of Sq    RSS    AIC
## <none>                  6106.3 805.44
## - CRIM     1     55.03 6161.4 805.62
## - TAX      1     94.60 6200.9 807.17
## - ZN       1    156.48 6262.8 809.58
## - NOX      1    222.04 6328.4 812.12
## - RAD      1    244.60 6350.9 812.98
## - CHAS     1    362.71 6469.0 817.46
## - PTRATIO  1    501.94 6608.3 822.63
## - DIS      1    733.54 6839.9 831.00
## - RM       1    839.28 6945.6 834.73
## - LSTAT    1   2015.89 8122.2 872.76
## Start:  AIC=998.36
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS +
##     RAD + TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS     AIC
## - AGE      1      0.60 7447.7  996.39
## - INDUS    1      2.50 7449.6  996.46
## <none>                  7447.1  998.36
## - CRIM     1     82.80 7529.9  999.72
## - ZN       1    131.15 7578.3 1001.67
## - TAX      1    145.57 7592.7 1002.25
## - CHAS     1    306.16 7753.3 1008.61
## - NOX      1    311.06 7758.2 1008.80
## - RAD      1    318.10 7765.2 1009.08
## - PTRATIO  1    604.93 8052.1 1020.10
## - DIS      1    687.55 8134.7 1023.21
## - RM       1   1160.05 8607.2 1040.37
## - LSTAT    1   1984.27 9431.4 1068.17
##
## Step:  AIC=996.39
## .outcome ~ CRIM + ZN + INDUS + CHAS + NOX + RM + DIS + RAD +
##     TAX + PTRATIO + LSTAT
##
##           Df Sum of Sq    RSS     AIC
## - INDUS    1      2.44 7450.2  994.49
## <none>                  7447.7  996.39
## - CRIM     1     83.05 7530.8  997.76
## - ZN       1    130.60 7578.3  999.67
## - TAX      1    144.98 7592.7 1000.25
## - CHAS     1    307.16 7754.9 1006.67
## - RAD      1    318.53 7766.3 1007.12
## - NOX      1    325.24 7773.0 1007.38
## - PTRATIO  1    607.65 8055.4 1018.23
## - DIS      1    823.45 8271.2 1026.27
## - RM       1   1228.42 8676.2 1040.80
## - LSTAT    1   2131.71 9579.5 1070.91
##
## Step:  AIC=994.49
## .outcome ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD + TAX + PTRATIO +
##     LSTAT
##
```

```
##           Df Sum of Sq    RSS     AIC
## <none>                 7450.2  994.49
## - CRIM     1     84.63 7534.8  995.92
## - ZN       1    128.16 7578.4  997.67
## - TAX      1    161.56 7611.8  999.01
## - CHAS     1    313.52 7763.7 1005.02
## - NOX      1    332.01 7782.2 1005.74
## - RAD      1    333.50 7783.7 1005.80
## - PTRATIO  1    608.19 8058.4 1016.34
## - DIS      1    876.34 8326.5 1026.29
## - RM       1   1227.82 8678.0 1038.86
## - LSTAT    1   2146.13 9596.3 1069.44
```

```
print(step_kcv)
```

```
## Linear Regression with Stepwise Selection
##
## 304 samples
##  12 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 243, 242, 244, 244, 243
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   5.192851  0.7001766  3.627941
```

```
#this is the final model
step_kcv$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ CRIM + ZN + CHAS + NOX + RM + DIS + RAD +
##     TAX + PTRATIO + LSTAT, data = dat)
##
## Coefficients:
## (Intercept)         CRIM           ZN         CHAS          NOX           RM
##    39.91340     -0.09989      0.04068      4.40706    -17.95096      3.76516
##         DIS          RAD          TAX      PTRATIO        LSTAT
##    -1.56068      0.30874     -0.01113     -0.87767     -0.59446
```

```
#how many variables do you have in the final model?
#There are 10 variables in the final model.
#Which variables are dropped?
# DIS, AGE are dropped.

# prediction on test data
yhat.kcv = predict(step_kcv$finalModel, newdata=data.frame(test.data))
# RMSE for test data
error.test.kcv <- yhat.kcv - test.data$MEDV
rmse.test.kcv <- sqrt(mean(error.test.kcv^2))
rmse.test.kcv
```

```
## [1] 4.444037
```

#sub3 Use lasso penalty to fit a regularized regression model with cross validation approach. Do the same variables disappear as in stepwise approach? Report the RMSE of this model on test data.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

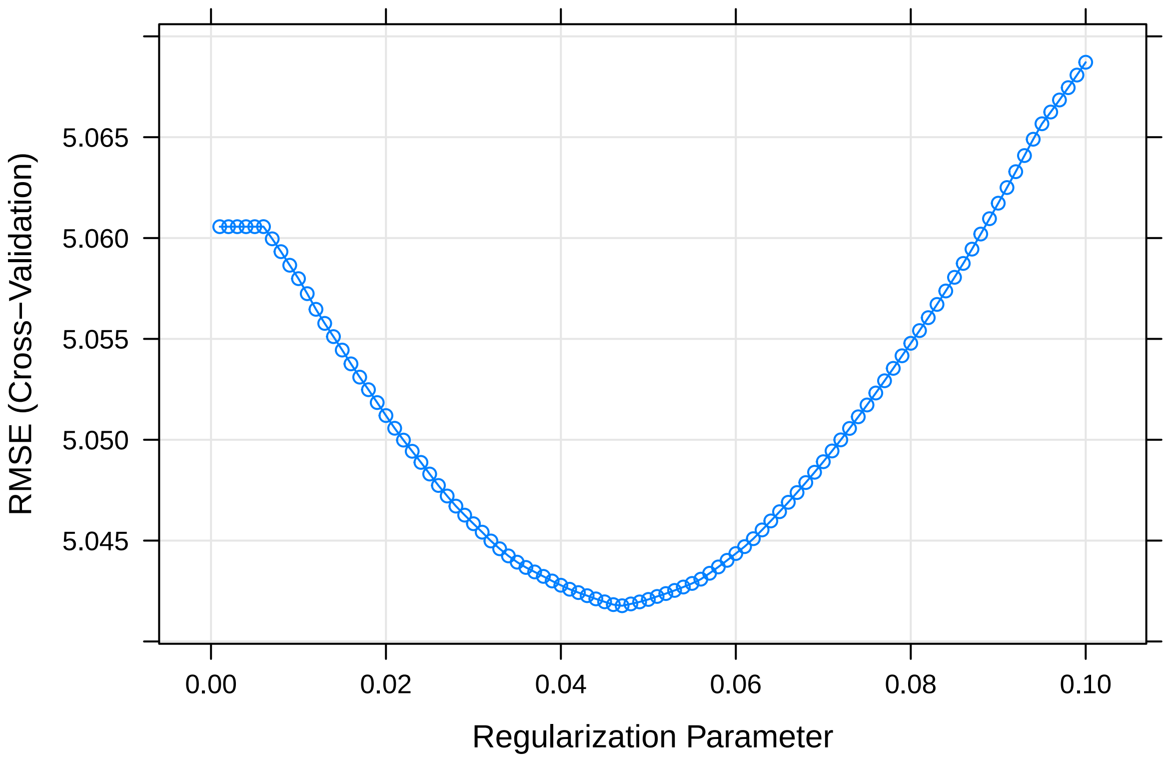
```
library(caret)
set.seed(0)
train_control <- trainControl(method="cv", number=10)
glmnet.lasso <- train(MEDV~ ., data = training.data, method = "glmnet", trControl = train_control, tune(
glmnet.lasso$bestTune # best lambda
```

```
##    alpha lambda
## 47     1  0.047
```

```
lasso.model <- coef(glmnet.lasso$finalModel, glmnet.lasso$bestTune$lambda)
lasso.model # coefficients
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  36.572596109
## CRIM         -0.076905645
## ZN            0.033608827
## INDUS         .
## CHAS          4.321138582
## NOX         -15.945073001
## RM            3.856701972
## AGE           .
## DIS          -1.383806791
## RAD           0.231501515
## TAX          -0.008014878
## PTRATIO      -0.850395740
## LSTAT        -0.596545500
```

```
plot(glmnet.lasso)
```




```
#report the RMSE of this model on test data
#remove the predictor variable
yhat.lasso = predict(glmnet.lasso, s = glmnet.lasso$bestTune, newdata=data.frame(test.data))
error.test.lasso <- yhat.lasso - test.data$MEDV
rmse.test.lasso <- sqrt(mean(error.test.lasso^2))
rmse.test.lasso
```

```
## [1] 4.423969
```

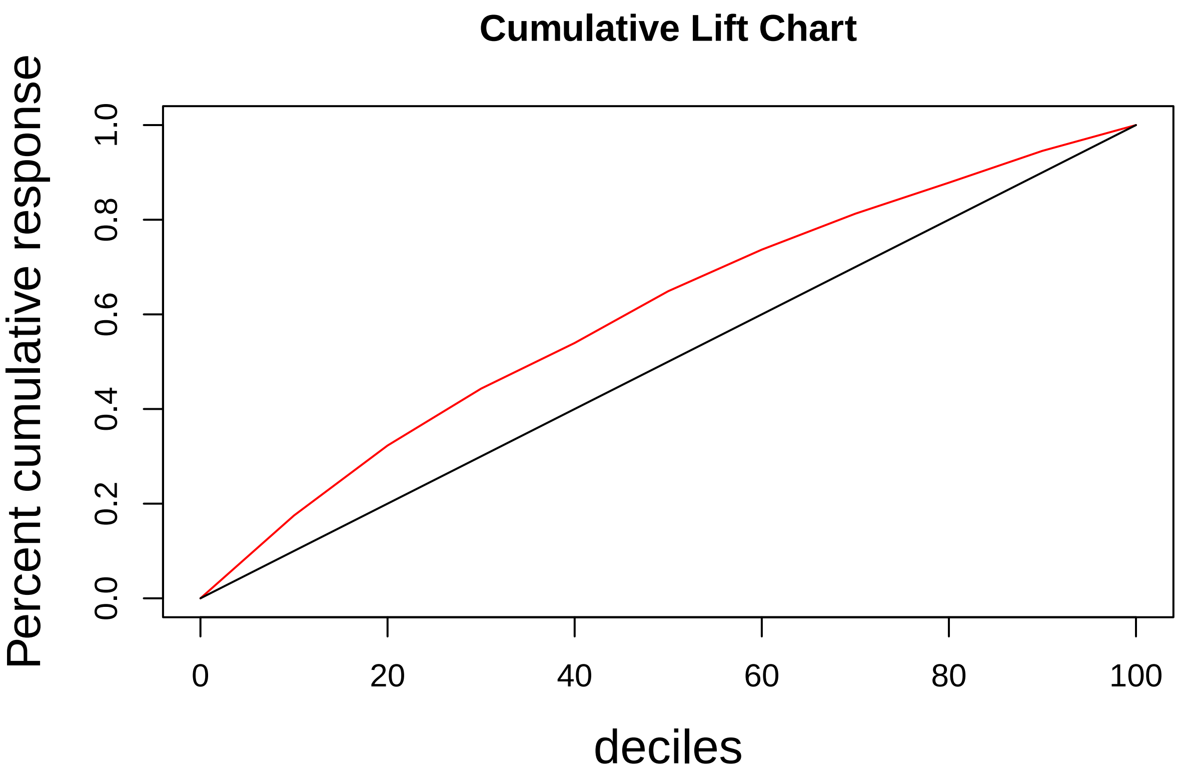
#sub4 Compare the models obtained in the above three steps. Create lift charts on test data for all models and comment on that.

```
#mlr model
# use this for getting Lift chart on test data
library(gains)
pred.prob.test_1 <- predict(fit2, newdata = test.data,type = "response")
gain_1 <- gains(test.data$MEDV, pred.prob.test_1)
gain_1
```

```
## Depth                            Cume    Cume Pct                     Mean
##   of          Cume      Mean     Mean    of Total    Lift    Cume    Model
## File     N       N      Resp     Resp        Resp   Index    Lift    Score
## -------------------------------------------------------------------------
##    10    5       5     39.90    39.90       17.5%     175     175    36.96
##    20    5      10     33.72    36.81       32.3%     148     161    33.39
##    30    5      15     27.42    33.68       44.3%     120     148    28.64
##    40    5      20     22.00    30.76       54.0%      96     135    26.30
##    50    5      25     24.96    29.60       64.9%     109     130    22.89
##    60    5      30     20.02    28.00       73.7%      88     123    20.95
##    70    5      35     17.32    26.48       81.3%      76     116    19.98
##    80    5      40     14.94    25.04       87.8%      66     110    17.59
##    90    5      45     15.34    23.96       94.6%      67     105    14.72
```

```
##   100      5      50      12.42      22.80      100.0%       54     100       8.77
```

```
# Plot Lift chart: Percent cumulative response
x_1 <- c(0, gain_1$depth)
pred.y_1 <- c(0, gain_1$cume.pct.of.total)
avg.y_1 <- c(0, gain_1$depth/100)
plot(x_1, pred.y_1, main = "Cumulative Lift Chart", xlab = "deciles",
     ylab = "Percent cumulative response", type = "l", col = "red", cex.lab = 1.5)
lines(x_1, avg.y_1, type = "l")
```




```
RMSE_test
```

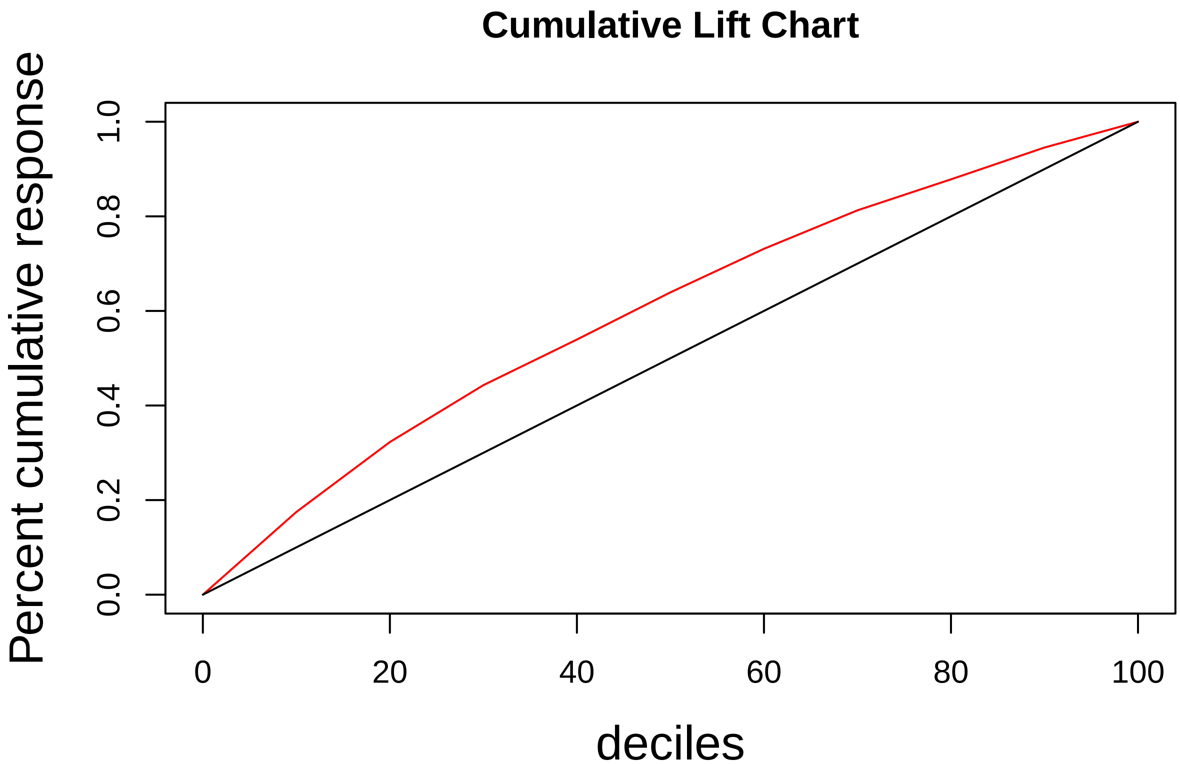
```
## [1] 4.422687
```

```
#stepwise regression with cross validation model
# use this for getting Lift chart on test data
library(gains)
pred.prob.test_2 <- predict(step_kcv$finalModel, newdata = test.data,type = "response")
gain_2 <- gains(test.data$MEDV, pred.prob.test_2)
gain_2
```

```
## Depth                            Cume    Cume Pct                     Mean
##  of          Cume      Mean      Mean    of Total     Lift    Cume    Model
## File    N      N       Resp      Resp        Resp    Index    Lift    Score
## ---------------------------------------------------------------------------
##    10    5      5     39.90     39.90       17.5%      175     175     36.97
##    20    5     10     33.72     36.81       32.3%      148     161     33.37
##    30    5     15     27.42     33.68       44.3%      120     148     28.59
##    40    5     20     22.00     30.76       54.0%       96     135     26.26
##    50    5     25     22.76     29.16       63.9%      100     128     22.80
##    60    5     30     21.00     27.80       73.1%       92     122     21.01
##    70    5     35     18.54     26.48       81.3%       81     116     19.93
```

```
##     80      5      40      14.94      25.04       87.8%      66      110      17.58
##     90      5      45      15.34      23.96       94.6%      67      105      14.76
##    100      5      50      12.42      22.80      100.0%      54      100       8.74
```

```r
# Plot Lift chart: Percent cumulative response
x_2 <- c(0, gain_2$depth)
pred.y_2 <- c(0, gain_2$cume.pct.of.total)
avg.y_2 <- c(0, gain_2$depth/100)
plot(x_2, pred.y_2, main = "Cumulative Lift Chart", xlab = "deciles",
     ylab = "Percent cumulative response", type = "l", col = "red", cex.lab = 1.5)
lines(x_2, avg.y_2, type = "l")
```




```r
rmse.test.kcv
```

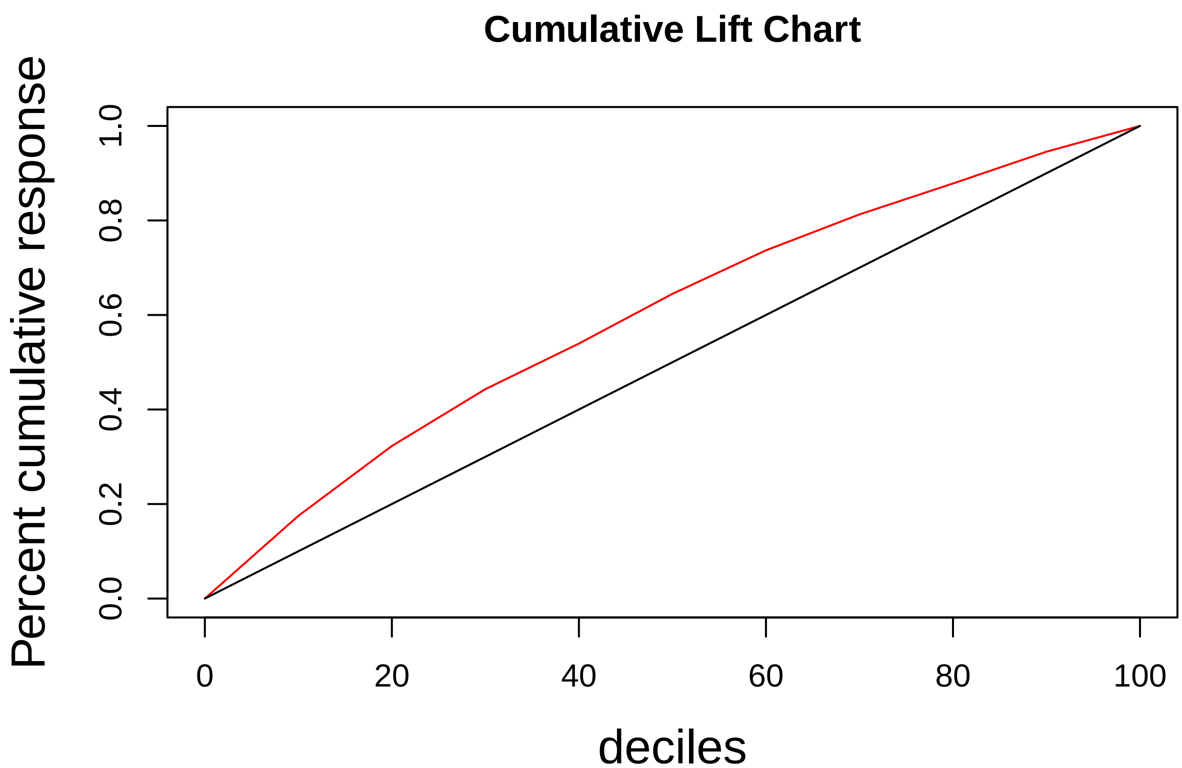
```
## [1] 4.444037
```

```r
#lasso penalty with cross validation model
# use this for getting Lift chart on test data
library(gains)
pred.prob.test_3 <- predict(glmnet.lasso, s = glmnet.lasso$bestTune , newdata = test.data)
gain <- gains(test.data$MEDV, pred.prob.test_3)
gain
```

```
## Depth                            Cume     Cume Pct                       Mean
##   of           Cume      Mean    Mean     of Total     Lift    Cume     Model
## File     N       N       Resp    Resp         Resp    Index    Lift     Score
## -------------------------------------------------------------------------------
##     10      5       5      39.90      39.90       17.5%     175      175      36.96
##     20      5      10      33.72      36.81       32.3%     148      161      33.02
##     30      5      15      27.42      33.68       44.3%     120      148      28.65
##     40      5      20      22.00      30.76       54.0%      96      135      26.14
##     50      5      25      23.98      29.40       64.5%     105      129      22.98
```

```
##   60      5      30      21.00      28.00       73.7%      92     123      21.21
##   70      5      35      17.32      26.48       81.3%      76     116      20.19
##   80      5      40      14.94      25.04       87.8%      66     110      17.81
##   90      5      45      15.34      23.96       94.6%      67     105      15.04
##  100      5      50      12.42      22.80      100.0%      54     100       9.03
```

```
# Plot Lift chart: Percent cumulative response
x <- c(0, gain$depth)
pred.y <- c(0, gain$cume.pct.of.total)
avg.y <- c(0, gain$depth/100)
plot(x, pred.y, main = "Cumulative Lift Chart", xlab = "deciles",
     ylab = "Percent cumulative response", type = "l", col = "red", cex.lab = 1.5)
lines(x, avg.y, type = "l")
```




```
rmse.test.lasso
```

```
## [1] 4.423969
```

the best rmse of the three models is the orgianl model with no varibale selection done on it

part B

```
## Include the functions required for data partitioning
source("/Users/cindydunn/Desktop/Grad_School/math540/Variable_Selection/myfunctions copy.r")
#get data
b_housing_log <- read.csv("/Users/cindydunn/Desktop/Grad_School/math540/hw1/data/BostonHousing.csv")
#remove variable 'MEDV' from the data
b_housing_log <- b_housing_log[,-13]
attach(b_housing_log)
```

```
## The following objects are masked from b_housing:
##
##     AGE, CAT..MEDV, CHAS, CRIM, DIS, INDUS, LSTAT, NOX, PTRATIO, RAD,
```

```
##     RM, TAX, ZN
```

#q1 Partition the data into training, validation, and test data sets. Create a logistic # regression model on training data using all regressors and report the # performance of that model on test data. What is the effect on the odds of # houses having high median value when the per capita crime rate of a town is # increased by 0.1?

```
RNGkind (sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
set.seed(000) ## set seed so that you get same partition each time
p3 <- partition.3(b_housing_log, 0.6, 0.3) ## creating 60:30:10 partition
training.data <- p3$data.train # training data
validation.data <- p3$data.val # validation data
test.data <- p3$data.test # test data

# Create a logistic regression model on training data using all regressors and report the performance o
fit3 <- glm(CAT..MEDV ~ ., data = training.data, family = "binomial")
summary(fit3)
```

```
##
## Call:
## glm(formula = CAT..MEDV ~ ., family = "binomial", data = training.data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.65666  -0.09117  -0.02326  -0.00171   2.22246
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.881800   9.254296   0.636 0.525053
## CRIM         0.051483   0.065637   0.784 0.432831
## ZN           0.050882   0.021777   2.336 0.019468 *
## INDUS       -0.155984   0.133267  -1.170 0.241816
## CHAS         0.659457   1.142670   0.577 0.563859
## NOX         -0.339132   7.528861  -0.045 0.964072
## RM           1.543275   0.697762   2.212 0.026984 *
## AGE          0.018754   0.019182   0.978 0.328241
## DIS         -0.741303   0.294033  -2.521 0.011697 *
## RAD          0.316989   0.142246   2.228 0.025850 *
## TAX         -0.011442   0.006415  -1.784 0.074485 .
## PTRATIO     -0.533165   0.252337  -2.113 0.034608 *
## LSTAT       -0.579229   0.170767  -3.392 0.000694 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 275.012  on 303  degrees of freedom
## Residual deviance:  74.016  on 291  degrees of freedom
## AIC: 100.02
##
## Number of Fisher Scoring iterations: 9
```

```r
# What is the effect on the odds of houses having high median value when the per capita crime rate of a
fit3$coefficients
```

```
## (Intercept)        CRIM          ZN       INDUS        CHAS         NOX
##  5.88180042  0.05148281  0.05088152 -0.15598370  0.65945734 -0.33913162
##          RM         AGE         DIS         RAD         TAX     PTRATIO
##  1.54327478  0.01875399 -0.74130288  0.31698899 -0.01144228 -0.53316535
##       LSTAT
## -0.57922854
```

```r
exp(0.12306064*0.1)
```

```
## [1] 1.012382
```

the odds ratio of 1.012382 means the increase in the crime rate by 0.1 does not effect of odds of houses having high median value or that there or ever so slightly higher odds.

#q2 Considering "1" as the important class, conduct a search for the best cut-off value with the objective of striking a balance between sensitivity and specificity. Report the performance of the optimal model found in this search.

```r
library(caret)
pred.prob.val <- predict(fit3, newdata = validation.data,type = "response")
# pred.y.val <- ifelse(pred.prob.val > 0.5, 1, 0)
cutoff <- seq(0.1, 0.9, by = 0.05)
cutoff_and_kappa_df <- data.frame(cutoff = cutoff, kappa =0, sensitivity = 0, specificity = 0)
# make a new dataframe to store cutoff values and kappa values as well as sensitivity and specificity
for (i in 1:length(cutoff)) {
  pred.y.val <- ifelse(pred.prob.val > cutoff[i], 1, 0)
  # store the kappa value for each cutoff value
  cutoff_and_kappa_df[i,2] <- confusionMatrix(as.factor(pred.y.val), as.factor(validation.data$CAT..MEDV
  # store the sensitivity value for each cutoff value
    cutoff_and_kappa_df[i,3] <- confusionMatrix(as.factor(pred.y.val), as.factor(validation.data$CAT..MI
  # store the specificity value for each cutoff value
  cutoff_and_kappa_df[i,4] <- confusionMatrix(as.factor(pred.y.val), as.factor(validation.data$CAT..MEDV
}
# find the largest kappa value in the dataframe and the corresponding cutoff value aka the optimal pref
cutoff_and_kappa_df[cutoff_and_kappa_df$kappa == max(cutoff_and_kappa_df$kappa),]
```

```
##    cutoff     kappa sensitivity specificity
## 10   0.55 0.8350899        0.88   0.9685039
```

```r
cutoff_and_kappa_df
```

```
##    cutoff     kappa sensitivity specificity
## 1    0.10 0.7425743        1.00   0.8976378
## 2    0.15 0.8303098        1.00   0.9370079
## 3    0.20 0.8063420        0.96   0.9370079
## 4    0.25 0.8253375        0.96   0.9448819
## 5    0.30 0.8253375        0.96   0.9448819
## 6    0.35 0.7944712        0.88   0.9527559
## 7    0.40 0.8144644        0.88   0.9606299
## 8    0.45 0.8144644        0.88   0.9606299
## 9    0.50 0.8144644        0.88   0.9606299
## 10   0.55 0.8350899        0.88   0.9685039
## 11   0.60 0.8021477        0.80   0.9763780
## 12   0.65 0.7953551        0.76   0.9842520
```

```
## 13   0.70 0.7657534        0.72   0.9842520
## 14   0.75 0.7657534        0.72   0.9842520
## 15   0.80 0.7657534        0.72   0.9842520
## 16   0.85 0.6220352        0.52   0.9921260
## 17   0.90 0.5461783        0.44   0.9921260
```

```
#plot sensitivity and specificity and kappa values for each cutoff value
# on the y axis plot from 0 to 1 and on the x axis have the cutoff values
plot(cutoff_and_kappa_df$cutoff, cutoff_and_kappa_df$sensitivity, type = "l", col = "red", ylim = c(0,1
lines(cutoff_and_kappa_df$cutoff, cutoff_and_kappa_df$sensitivity, type = "l", col = "blue")
lines(cutoff_and_kappa_df$cutoff, cutoff_and_kappa_df$specificity, type = "l", col = "green")
lines(cutoff_and_kappa_df$cutoff, cutoff_and_kappa_df$kappa, type = "l", col = "red")
legend("topright", legend = c("kappa", "sensitivity", "specificity"), col = c("red", "blue", "green"), 
```