Homework 6

4375 Machine Learning with Dr. Mazidi

Anthony Martinez | amm180005

10/3/21

Problem 1: Comparison with Linear Regression

Step 1. Load Auto data and make train/test split

Using the Auto data in package ISLR, set seed to 1234 and divide into 75% train, 25% test

```
# your code here
library(ISLR)
df <- Auto
attach(df)

set.seed(1234)
i <- sample(1:nrow(df), round(nrow(df)*0.75), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

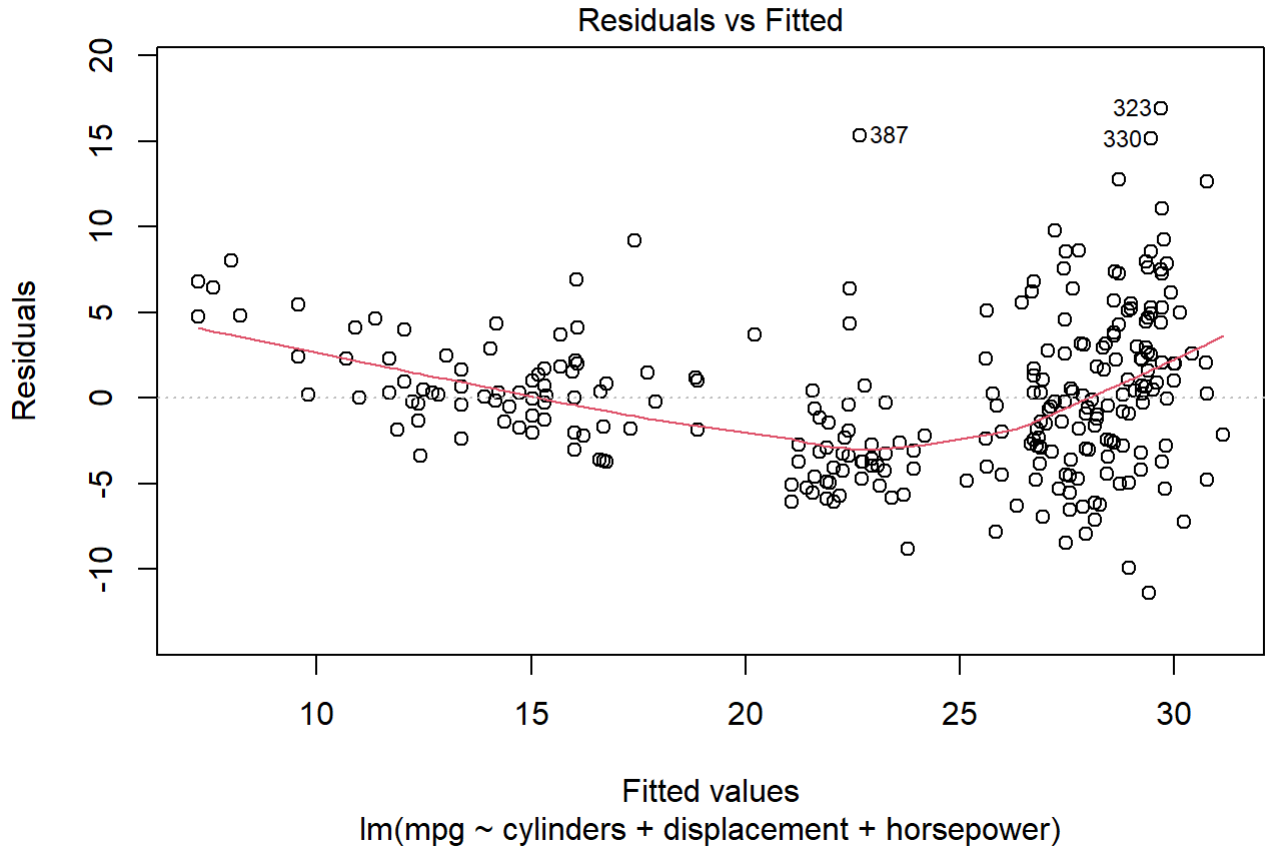
Step 2. Build linear regression model

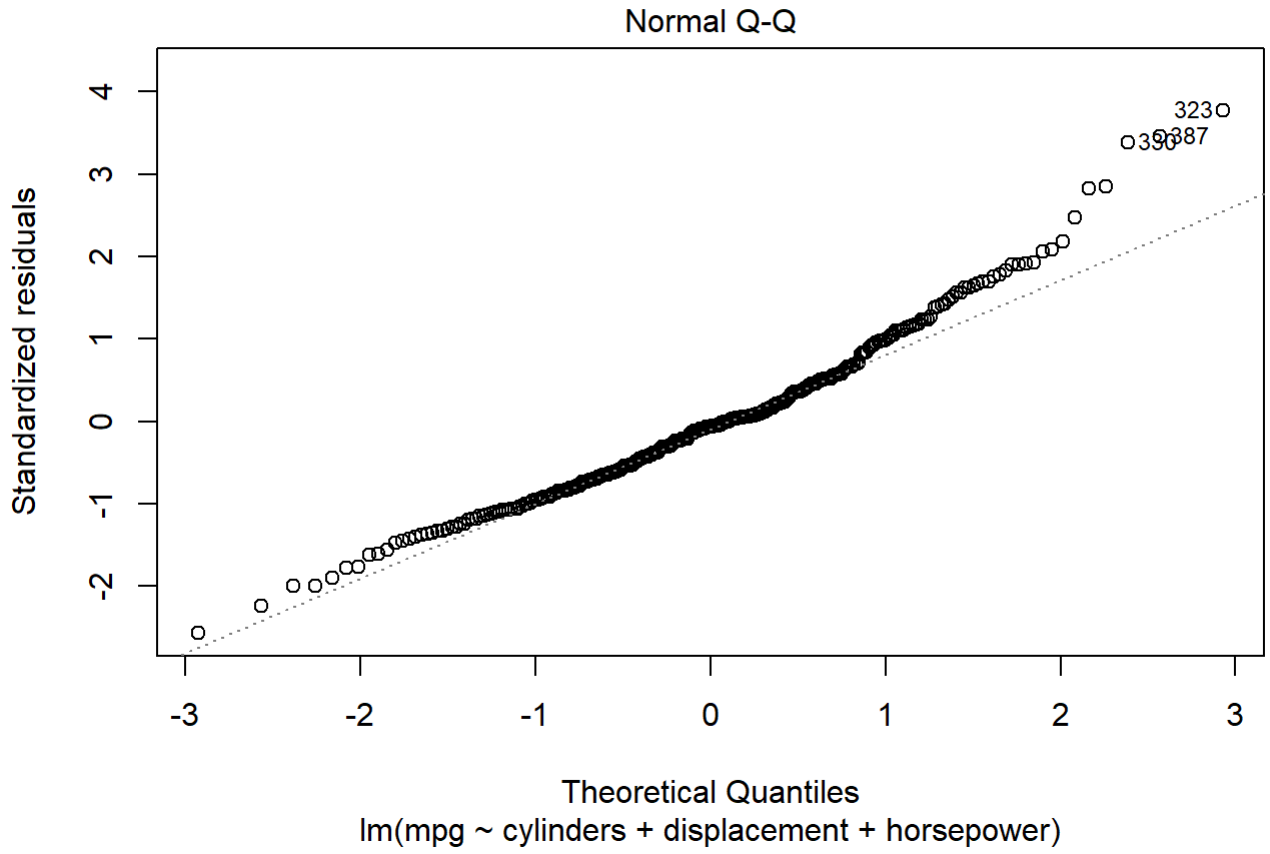
Build a linear regression model on the train data, with mpg as the target, and cylinders, displacement, and horsepower as the predictors. Output a summary of the model and plot the model to look at the residuals plots.

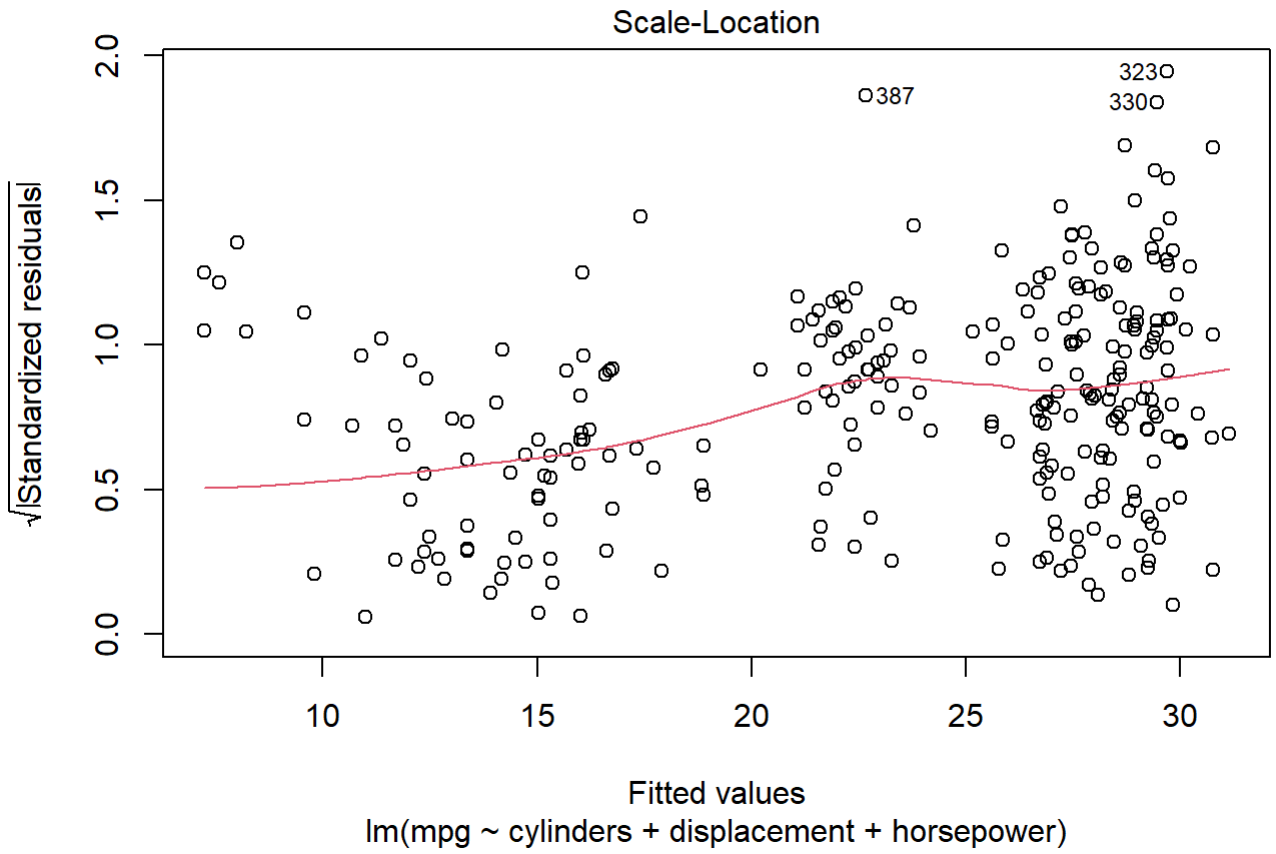
```
# your code here
lm1 <- lm(mpg~cylinders+displacement+horsepower, data=train)
summary(lm1)
```

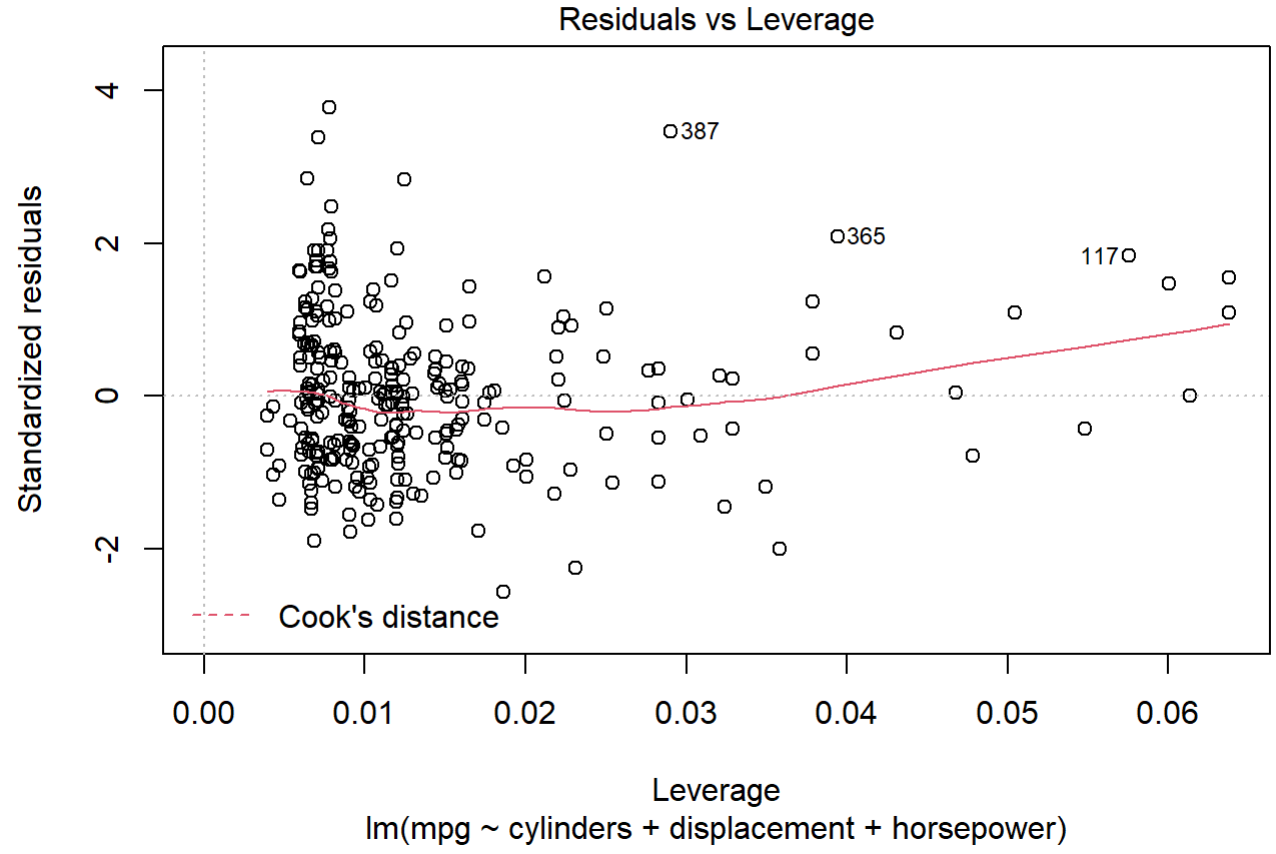
```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4207  -3.1426  -0.2873   2.2997  16.8950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.118273   1.484992  27.016  < 2e-16 ***
## cylinders    -1.078971   0.490277  -2.201   0.0285 *
## displacement -0.020006   0.009753  -2.051   0.0411 *
## horsepower   -0.067336   0.015062  -4.471 1.12e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.488 on 290 degrees of freedom
## Multiple R-squared:  0.6718, Adjusted R-squared:  0.6685
## F-statistic: 197.9 on 3 and 290 DF,  p-value: < 2.2e-16
```

```
plot(lm1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(mpg ~ cylinders + displacement + horsepower)
387
323
330
20
15
10
5
0
-5
-10
10
15
20
25
30


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(mpg ~ cylinders + displacement + horsepower)
323
387
330
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


Scale-Location
√|Standardized residuals|
387
323
330
0.0
0.5
1.0
1.5
2.0
10
15
20
25
30
Fitted values
lm(mpg ~ cylinders + displacement + horsepower)


Residuals vs Leverage
Standardized residuals
387
365
117
Cook's distance
-2
0
2
4
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Leverage
lm(mpg ~ cylinders + displacement + horsepower)

```
detach(df)
```

Step 3. Evaluate on the test data

Evaluate the model on the test data. Output correlation and mse.

```
# your code here
pred <- predict(lm1, newdata=test)
cor_lm1 <- cor(pred,test$mpg)
mse_lm1 <- mean((pred-test$mpg)^2)

print(paste("cor:", cor_lm1))
```

```
## [1] "cor: 0.805826128416024"
```

```
print(paste("mse:", mse_lm1))
```

```
## [1] "mse: 21.7583405424536"
```

Step 4. Try knn

Use knnreg() in library caret to fit the training data. Use the default k=1. Output your correlation and mse.

```
# your code here
library(caret)
```

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

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

```
fit <- knnreg(train[,2:4],train[,1], k=1)

predict_knn <- predict(fit, test[,2:4])
cor_knn <- cor(predict_knn, test$mpg)
mse_knn <- mean((predict_knn-test$mpg)^2)

print(paste("cor:", cor_knn))
```

```
## [1] "cor: 0.843707789760545"
```

```
print(paste("mse:", mse_knn))
```

```
## [1] "mse: 18.961006037415"
```

Step 5. Analysis

a. Compare correlation metric that each algorithm achieved. Your commentary here: The correlation for was slightly higher for the knn model than the linear regression model. Which means knn worked a slightly better for this data

b. Compare the mse metric that each algorithm achieved. Your commentary here:

The MSE was lower for the knn model in comparison to the linear model. This means knn worked slightly better for this data set. Recall, we want to minimize errors. Having two models with similar correlations we would want to take the model with a lower MSE.

c. Why do you think that the mse metric was so different compared to the correlation metric? Your commentary here:

Clustering algorithms work best when the data is scaled. If we were to scale the data and re run the knn algorithm the MSE would most likely be lower for the knn model.

d. Why do you think that kNN outperformed linear regresssion on this data? In your 2-3 sentence explanation, discuss bias of the algorithms. Your commentary here:

The data here was very scattered. It was hard to see any relationship when we plotted the residuals. Because of this it makes sense that linear regression would preform poorly on this data.Recall that linear regression is a high bias algorithm. Meaning, it will try to find a line of best fit even if one does not exist. Since we chose a small K value for the knn model there will be low bias and high variance.

Problem 2: Comparison with Logistic Regression

Step 1. Load Breast Cancer data, create regular and small factors, and divide into train/test

Using the BreastCancer data in package mlbench, create factor columns Cell.small and Cell.regular as we did in the last homework. Set seed to 1234 and divide into 75% train, 25% test.

Advice: use different names for test/train so that when you run parts of your script over and over the names don’t collide.

```
# your code here
library(mlbench)
data("BreastCancer")

BreastCancer$Cell.small <-0
BreastCancer$Cell.small[BreastCancer$Cell.size == 1] <-1
BreastCancer$Cell.small <- factor(BreastCancer$Cell.small)

BreastCancer$Cell.regular <-0
BreastCancer$Cell.regular[BreastCancer$Cell.shape == 1] <-1
BreastCancer$Cell.regular <- factor(BreastCancer$Cell.regular)

set.seed(1234)
i <- sample(1:nrow(BreastCancer), 0.75*nrow(BreastCancer), replace=FALSE)
train1 <- BreastCancer[i,]
test1 <- BreastCancer[-i,]
```

Step 2. Build logistic regression model

Build a logistic regression model with Class as the target and Cell.small and Cell.regular as the predictors. Output a summary of the model.

```
# your code here
glm1 <- glm(Class~Cell.small+Cell.regular, data=train1, family=binomial)

summary(glm1)
```

```
##
## Call:
## glm(formula = Class ~ Cell.small + Cell.regular, family = binomial,
##     data = train1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8367  -0.0474  -0.0474   0.6399   3.6861
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.4820     0.1731   8.564  < 2e-16 ***
## Cell.small1    -4.6801     0.7429  -6.300 2.98e-10 ***
## Cell.regular1  -3.5944     0.7655  -4.695 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 678.03  on 523  degrees of freedom
## Residual deviance: 240.57  on 521  degrees of freedom
## AIC: 246.57
##
## Number of Fisher Scoring iterations: 8
```

Step 3. Evaluate on the test data

Evaluate the model on the test data. Output accuracy and a table (or confusion matrix).

```
# your code here
probs1 <- predict(glm1, newdata = test1, type="response")
pred1 <- ifelse(probs1>0.5, 2,1)
acc2 <- mean(pred1==as.integer(test1$Class))
acc2
```

```
## [1] 0.8914286
```

```
table(pred1,test1$Class)
```

```
##
## pred1 benign malignant
##     1    100         2
##     2     17        56
```

Step 4. Try knn

Use the knn() function in package class to use the same target and predictors as step 2. Output accuracy and a table of results for knn.

```
# your code here
library(class)

# Split data so that the predictors are now just Cell.small and Cell.regular
set.seed(1234)
ind <- sample(2, nrow(BreastCancer), replace=TRUE, prob=c(0.75,0.25))
train2 <- BreastCancer[ind==1, 12:13]
test2 <- BreastCancer[ind==2, 12:13]

trainLabels = BreastCancer[ind==1,11]
testLabels = BreastCancer[ind==2,11]


new_pred <- knn(train=train2, test=test2, cl=trainLabels, k=1)
results <- new_pred==testLabels
acc3 <- length(which(results==TRUE)) / length(results)
acc3
```

```
## [1] 0.9289617
```

```
table(results,new_pred)
```

```
##        new_pred
## results benign malignant
##   FALSE      0        13
##   TRUE     115        55
```

Step 5. Try knn on original predictors

Run kNN using predictor columns 2-6, 8-10, using default k=1. Output accuracy and a table of results.

Compare the results from step 4 above to a model which uses all the predictors. Provide some analysis on why you see these results: Commentary: When knn on only the 2 predictors the correlation was slightly higher than when running knn on all predictors. When we look at the table, knn on the original predictors had 6 false predictions for benign while knn with 2 predictors had 0.

```
# your code here
original_pred <- knn(train=train1[c(2:6,8:10)], test=test1[c(2:6,8:10)], cl=train1$Class, k=1)
results2 <- original_pred==test1$Class
acc4 <- length(which(results2==TRUE)) / length(results2)
acc4
```

```
## [1] 0.92
```

```
table(results2, original_pred)
```

```
##         original_pred
## results2 benign malignant
##    FALSE      6         8
##    TRUE     109        52
```

Step 6. Try logistic regression on original predictors

Run logistic regression using predictor columns 2-6, 8-10. Output accuracy and a table of results.

Compare the results from the logistic regression and knn algorithms using all predictors except column 7 in the steps above. Provide some analysis on why you see these results: The accuracy for this model is slightly lower than the model from the previous step. Meaning the knn model worked better than the logistic regression model when using the original predictors (minus column 7). The cause for the slight preformance differnce is due to the fact that the logistic regression model missclassified 1 more benign cell compared to the knn model as shown in the tables.

```
# your code here
glm_all <- glm(train1$Class~., data=train1[c(2:6,8:10)], family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
probsAll <- predict(glm_all, newdata = test1[c(2:6,8:10)], type="response")
predAll <- ifelse(probsAll>0.5, 2,1)
accAll <- mean(predAll==as.integer(test1$Class))
accAll
```

```
## [1] 0.9085714
```

```
table(predAll, test1$Class)
```

```
##
## predAll benign malignant
##       1    109         8
##       2      8        50
```