Homework 3

Shijie Shi

Problem 1 (p168, exercise 4)

(1) 10%. X is uniformly distributed on [0,1].

(2) 1%. 0.1*0.1=0.01.

(3) $\mathbf{100 * 0.1^{100} = 0.1^{98}}\%$

(4) When the number of the predictor increases, the fraction of observations we can get from the fixed range of the features will decreases quickly, which means the available training sample will be very small.

(5) Assume the length of the hypercube, which contains 10% of the observations, is L:
when p=1, L=0.1
when p=2, L=$\sqrt{0.1}$ = 0.316
when p=100, L=$\sqrt[100]{0.1}$ = 0.977
When p becomes large, L approaches 1, which means we need to use almost all of the observations to make the prediction, this is in contrast to KNN and other local approaches are trying to perform prediction using the nearby observations of the test observation.

Problem 2 (p170, exercise 6)

(1)

$$\boldsymbol{log\left(\frac{P(A)}{1 - P(A)}\right) = -6 + 0.05(40) + 3.5}$$

$$\boldsymbol{log\left(\frac{P(A)}{1 - P(A)}\right) = -0.5}$$

$$\boldsymbol{P(A) = \frac{1}{\sqrt{e} + 1} \approx 0.378}$$

(2)

$$\boldsymbol{log\left(\frac{P(A)}{1 - P(A)}\right) = -6 + 0.05X_1 + 3.5}$$

$$\frac{P(A)}{1 - P(A)} = e^{0.05X_1 - 2.5}$$

$$P(A) = \frac{e^{0.05X_1 - 2.5}}{1 + e^{0.05X_1 - 2.5}} \geq 0.5$$

$$e^{0.05X_1 - 2.5} \geq e^0$$

$$0.05X_1 - 2.5 \geq 0$$

$$X_1 \geq 50$$

A student with a 3.5 undergrad GPA needs to study at least 50 hours to have a 50% chance of getting an A in this statistic class.

Problem 3 (p170, exercise 8)

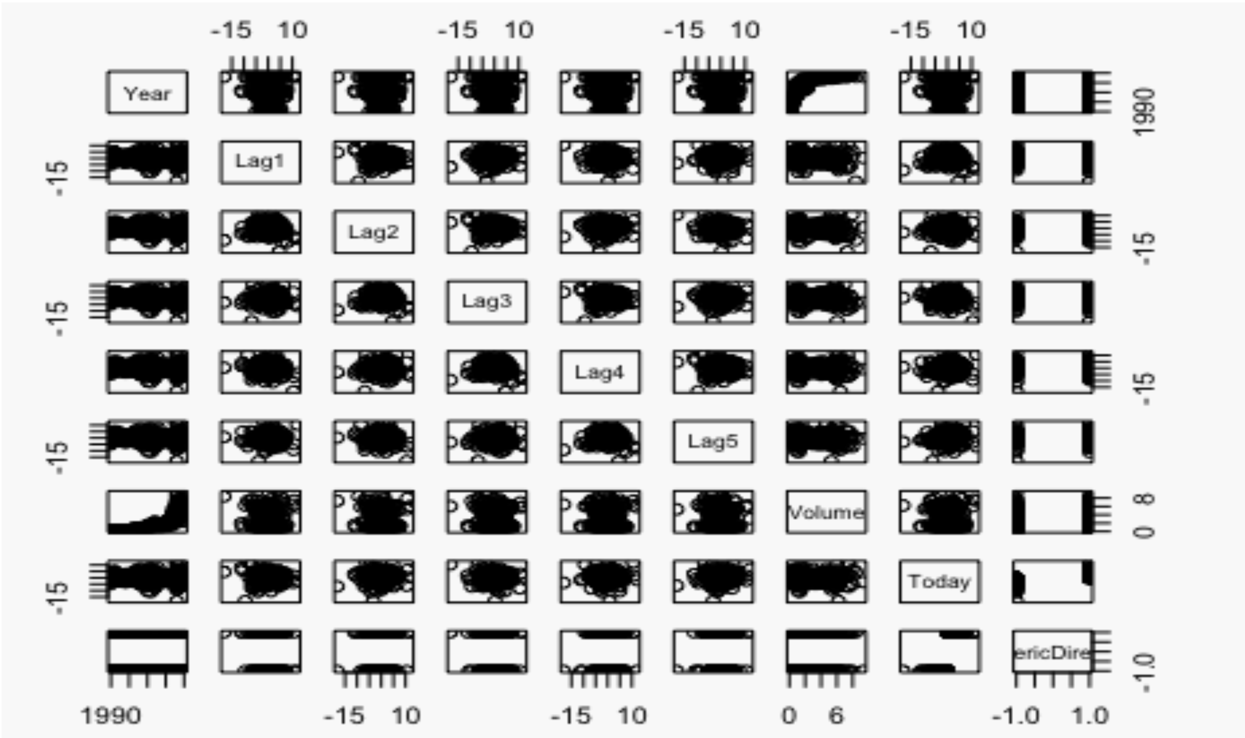
I would choose the logistic regression.
KNN with K=1 has zero training error so the testing error for this method is 36% (18% * 2), which is worse than the logistic regression (testing error=30%).

Problem 4 (p171, exercise 10)

(1) I recoded the *direction* into a numerical variable where -1 represents “Down” and +1 represent “Up”. From the correlation table below, we see all the numbers are small except for the high correlation between *today* and *direction* (0.72002), which is understandable because *direction* is decided by the value of *today*. From the paired plot, we find a high correlation between *year* and *volume*.

```
rm(list=ls())
library(ISLR)
Direction = Weekly$Direction
Weekly$Direction = NULL
Weekly$NumericDirection = as.numeric( Direction )
# Maps Down=>1 and Up=>2
Weekly$NumericDirection[ Weekly$NumericDirection==1 ] = -1
# Maps Down=>-1 and Up=>2
Weekly$NumericDirection[ Weekly$NumericDirection==2 ] = +1
# Maps Down=>-1 and Up=>+1

summary(Weekly)


# Table 1 The summary table

##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.195   Min.   :-18.195   Min.   :-18.195
```

```
##  1st Qu.:1995   1st Qu.: -1.154   1st Qu.: -1.154   1st Qu.: -1.158
##  Median :2000   Median :  0.241   Median :  0.241   Median :  0.241
##  Mean   :2000   Mean   :  0.151   Mean   :  0.151   Mean   :  0.147
##  3rd Qu.:2005   3rd Qu.:  1.405   3rd Qu.:  1.409   3rd Qu.:  1.409
##  Max.   :2010   Max.   : 12.026   Max.   : 12.026   Max.   : 12.026
##       Lag4               Lag5              Volume           Today
##  Min.   :-18.195   Min.   :-18.195   Min.   :0.087   Min.   :-18.195
##  1st Qu.: -1.158   1st Qu.: -1.166   1st Qu.:0.332   1st Qu.: -1.154
##  Median :  0.238   Median :  0.234   Median :1.003   Median :  0.241
##  Mean   :  0.146   Mean   :  0.140   Mean   :1.575   Mean   :  0.150
##  3rd Qu.:  1.409   3rd Qu.:  1.405   3rd Qu.:2.054   3rd Qu.:  1.405
##  Max.   : 12.026   Max.   : 12.026   Max.   :9.328   Max.   : 12.026
##  NumericDirection
##  Min.   :-1.000
##  1st Qu.:-1.000
##  Median : 1.000
##  Mean   : 0.111
##  3rd Qu.: 1.000
##  Max.   : 1.000

Weekly.cor = cor(Weekly)
Weekly.cor[9,]


# Table 2 The correlation table

##             Year             Lag1             Lag2             Lag3
##         -0.02220         -0.05000          0.07270         -0.02291
##             Lag4             Lag5           Volume            Today
##         -0.02055         -0.01817         -0.01800          0.72002
## NumericDirection
##          1.00000

pairs(Weekly)


# Table 3 The pairs plot
```



(2) Lag 2 shows the statistical significance. Details are shown below.

```
Weekly$NumericDirection = NULL
Weekly$Direction = Direction

M1_FiveLag = glm( Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data
=Weekly, family=binomial )
print( summary(M1_FiveLag) )

# Table 4 The logistic regression with five lags and volume as predictors

##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.695  -1.256   0.991   1.085   1.458
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2669     0.0859    3.11   0.0019 **
## Lag1         -0.0413     0.0264   -1.56   0.1181
## Lag2          0.0584     0.0269    2.18   0.0296 *
## Lag3         -0.0161     0.0267   -0.60   0.5469
```

```
## Lag4          -0.0278     0.0265   -1.05   0.2937
## Lag5          -0.0145     0.0264   -0.55   0.5833
## Volume        -0.0227     0.0369   -0.62   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500
##
## Number of Fisher Scoring iterations: 4
```

(3) The overall corrected prediction is: $(\mathbf{54}+\mathbf{557}) \div \mathbf{1089} \;=\; \mathbf{0.561}$

For the Up part, the logistic model predicted it very well. The correct rate is 92.1%.
$557 \div (557+48) \;= 0.921$.

For the Down part, the logistic model predicted it badly. The correct rate is only 11.2%.
$54 \div (54+430) \;= 0.112$.

False positive rate: 88.84%
False negative rate: 7.93%
Overall error rate: 43.89%
Details below.

```
p_hat = predict( M1_FiveLag, newdata=Weekly, type="response" )
y_hat = rep( "Down", length(p_hat) )
y_hat[ p_hat > 0.5 ] = "Up"
CM = table( predicted=y_hat, truth=Weekly$Direction )
print( CM )


# Table 5 The confusion table for M1_FiveLag

##          truth
## predicted Down  Up
##      Down   54  48
##      Up    430 557
```

(4) Logistic Regression

Overall fraction of correct predictions: $(9+56) \div (9+5+34+56) \;=\; 0.625$

```
train = ( Weekly$Year < 2009 )
test = ( Weekly$Year >= 2009 )
M2_lag2_LM = glm( Direction ~ Lag2, data=Weekly, family=binomial, subset=trai
n )
```

```r
p_hat = predict(M2_lag2_LM, newdata=Weekly[test,], type="response" )
y_hat = rep( "Down", length(p_hat) )
y_hat[ p_hat > 0.5 ] = "Up"
CM = table( predicted=y_hat, truth=Weekly[test,]$Direction )
print( CM )

# Table 6 The confusion table for M2_lag2_LM

##           truth
## predicted Down Up
##      Down    9  5
##      Up     34 56
```

(5) LDA

Overall fraction of correct predictions: 0.625

```r
library(MASS)
M2_lag2_LDA= lda( Direction ~ Lag2, data=Weekly, family=binomial, subset=trai
n )
lda.predict = predict( M2_lag2_LDA, newdata=Weekly[test,] )
CM = table( predicted=lda.predict$class, truth=Weekly[test,]$Direction )
print( CM )

# Table 7 The confusion table for LDA fitting

##           truth
## predicted Down Up
##      Down    9  5
##      Up     34 56


mean(lda.predict$class == Weekly[test,]$Direction)

## [1] 0.625
```

(6) QDA

Overall fraction of correct predictions: 0.5865

```r
M2_lag2_QDA= qda( Direction ~ Lag2, data=Weekly, family=binomial, subset=trai
n )
qda.predict = predict( M2_lag2_QDA, newdata=Weekly[test,] )
CM = table( predicted=qda.predict$class, truth=Weekly[test,]$Direction )
print( CM )

# Table 8 The confusion table for QDA fitting

##           truth
## predicted Down Up
##      Down    0  0
##      Up     43 61
```

```
mean(qda.predict$class == Weekly[test,]$Direction)

## [1] 0.5865
```

(7) KNN with K=1

Overall fraction of correct predictions: 0.5

```
library(class)

X.train = data.frame( Lag2=Weekly[train, ]$"Lag2" )
Y.train = Weekly[ train, ]$"Direction"
X.test = data.frame( Lag2=Weekly[ test, ]$"Lag2" )

M2_lag2_KNN1 = knn( X.train, X.test, Y.train, k=1 )
CM = table( predicted=M2_lag2_KNN1, truth=Weekly[ test, ]$Direction )
print( CM )

# Table 9 The confusion table for KNN(N=1) fitting

##          truth
## predicted Down Up
##      Down   21 30
##      Up     22 31


mean (M2_lag2_KNN1 == Weekly[test,]$Direction)

## [1] 0.5
```

(8) Having the same highest correct rate (0.625), logistic regression and LDA (using only lag2 predictor) provide the best results on the data.

(9) I tried different combinations of predictors, including possible transformations and interactions, as well as different K for KNN classifier. The best prediction I can get is shown below.
Overall, the original LDA and logistic regression have better performance in terms of test error rate.

```
lag2_LDA= lda( Direction ~ Lag2+Lag2*Lag2, data=Weekly, family=binomial, subs
et=train )
lda2.predict = predict(lag2_LDA, newdata=Weekly[test,] )
CM = table( predicted=lda2.predict$class, truth=Weekly[test,]$Direction )
print( CM )

##          truth
## predicted Down Up
```

```
##       Down     9   5
##       Up      34  56
```

```r
# Table 12 The confusion table for LDA fitting with lag2, and Lag2*lag2
mean(lda2.predict$class == Weekly[test,]$Direction)
```

```
## [1] 0.625
```

Problem 5

(5) The modified function for weight:

```
# Find a colum with the weight
    weightCol = row.find('td',attrs={'class':'weight'})
# If there is no such column, let weight be nan.
    if weightCol==None:
      entry['weight'] = np.nan
#"some athletes may not have a weight record, even if there is a column"
    else:
      weightRaw = weightCol.contents[0].strip()
      if weightRaw =='':
        entry['weight'] = np.nan
      else:
        entry['weight']=weightRaw
```

(6) Multinomial logistic Regression, MLR

```r
library("nnet")
rosters=read.csv('~/Documents/IPython/hw3_files/rosters.csv')
rosters$sport <- relevel(rosters$sport, ref = "Baseball")
MLR <- multinom(sport ~ height + weight, data = rosters)
```

```
## # weights:  24 (15 variable)
## initial  value 395.978843
## iter  10 value 274.330405
## iter  20 value 217.212577
## iter  30 value 209.616614
## iter  40 value 209.159658
## iter  50 value 209.060903
## iter  60 value 209.025730
## final  value 209.012776
## converged
```

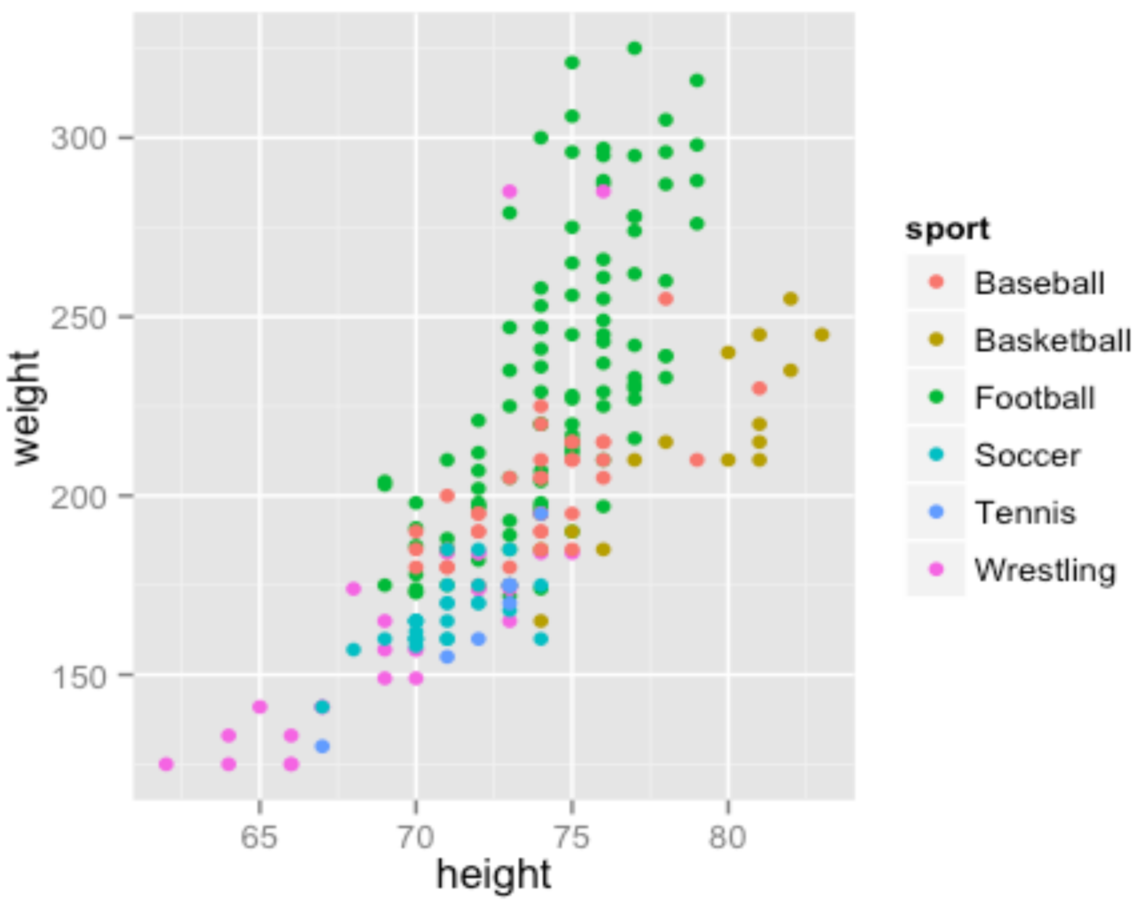
```r
summary(MLR)
```

```
## Call:
## multinom(formula = sport ~ height + weight, data = rosters)
```

```
##
## Coefficients:
##              (Intercept)   height   weight
## Basketball        -75.32  1.28962 -0.11415
## Football           17.57 -0.36691  0.04989
## Soccer             14.26  0.02111 -0.08866
## Tennis            -19.23  0.63065 -0.15578
## Wrestling          46.99 -0.60916 -0.01959
##
## Std. Errors:
##              (Intercept)  height  weight
## Basketball      0.247645 0.05711 0.02139
## Football        6.262267 0.10491 0.01125
## Soccer          0.519162 0.05303 0.02139
## Tennis          0.003663 0.08413 0.03529
## Wrestling       2.122585 0.05494 0.01837
##
## Residual Deviance: 418
## AIC: 448
```

(7) Scatter plot of weight vs. height, color each point according to the sport

```
library("ggplot2")
ggplot(data=rosters, aes(x=height,y=weight, color=sport))+geom_point()
```

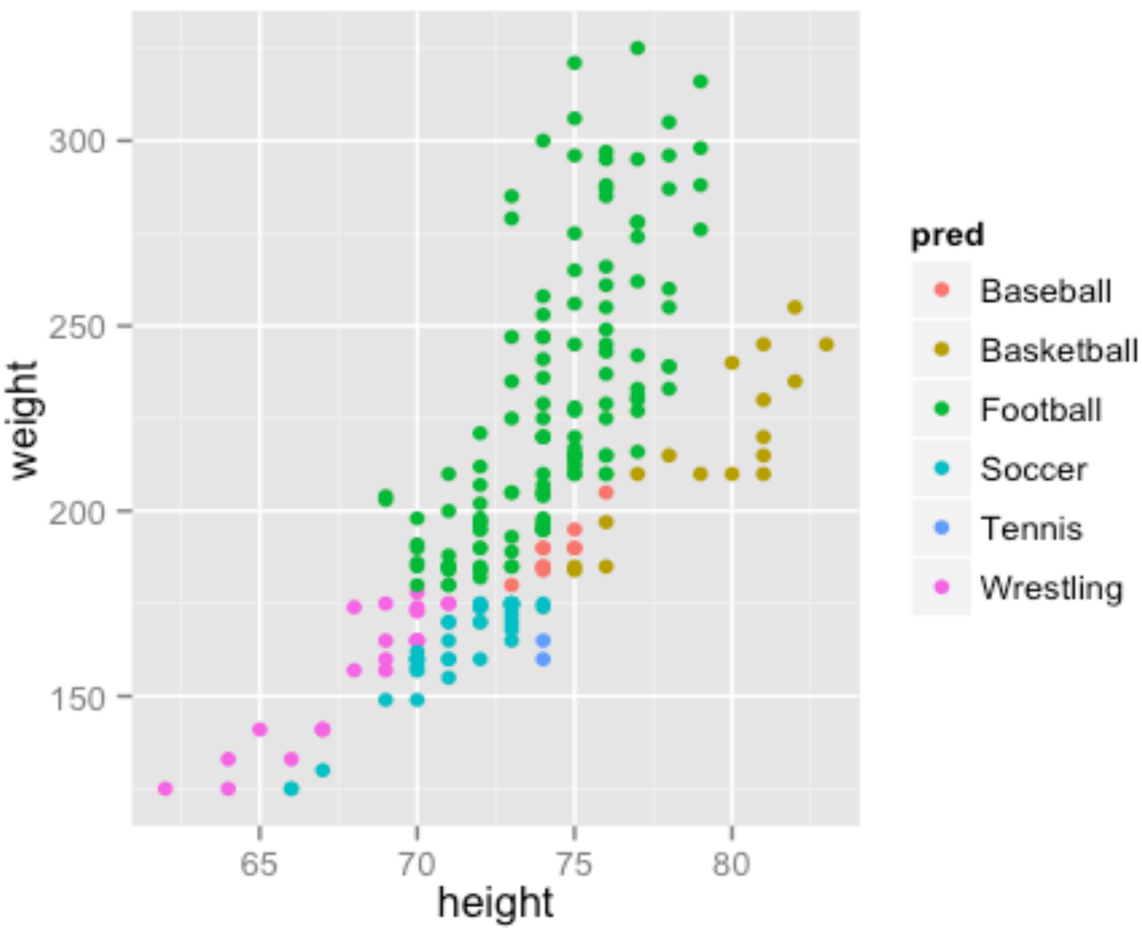


(8) Produce predictions for the training data using the function predict

```
MLR.pred=predict(MLR,rosters)
summary(MLR.pred)

##     Baseball  Basketball      Football        Soccer        Tennis    Wrestling
##           10          17           129            39             2           24
```

(9) Scatter plot of weight vs. height, color each point according to the prediction

```
rosters$pred = MLR.pred
ggplot(data=rosters, aes(x=height,y=weight,color=pred)) + geom_point()
```



(10) “0-1" Loss
The error rate for the training data is: (1 - 0.6199)*100% = 38.01%.

```
MLR.pred=predict(MLR,rosters)
mean(MLR.pred==rosters$sport)

## [1] 0.6199
```