Name: CHAN King Yeung
SID:1155119394
STAT4001 Homework 1

Question 1

Given $TSS = SYY = \sum(y_i - \bar{y})^2$, $RSS = \sum(y_i - \hat{y}_i)^2$ and $r = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2\cdot\sum(y_i-\bar{y})^2}}$, such that

$$\begin{aligned}
R^2 &= 1 - \frac{RSS}{SYY} \\
&= \frac{SYY-RSS}{SYY} \\
&= \frac{SS_{reg}}{SYY} \text{ where } SS_{reg} = \frac{SXY^2}{SXX} = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sum(x_i-\bar{x})^2} \text{ is the variation explained by } \beta_1 \text{ in the simple regression model} \\
&= \frac{SXY^2}{SXX\cdot SYY} \\
&= \frac{[\sum(x_i-\bar{x})(y_i-\bar{y})]^2}{\sum(x_i-\bar{x})^2\cdot\sum(y_i-\bar{y})^2} \\
&= \left[\frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2\cdot\sum(y_i-\bar{y})^2}}\right]^2 \\
&= r^2
\end{aligned}$$

Question 2

Please refer to the following console of output (see Appendix for your reference)

```
> library(boot)
> x = c(0.4, 0.8, 1.2)
> y = c(1.2, 1.8, 3.2)
> data = data.frame(x, y)
> reg = glm(y ~ x)
> loocv = cv.glm(data, reg)
> loocv$delta[1]
[1] 0.48
```

The LOOCV error is 0.48 for the given data.

Question 3
Please refer to the following console of output (see Appendix for your reference)

```
> set.seed(4001)
> x = c(4.3, 2.1, 5.3)
> y = c(2.4, 1.1, 2.8)
> data = data.frame(x, y)
> B = nrow(data)
> bt = list(data, data, data)
> alpha = rep(0, B)
>
> alphaValue = function(data) {
+   X = bt[[i]]$x
+   Y = bt[[i]]$y
+   alpha = (var(Y) - cov(X, Y)) / (var(X) + var(Y) - 2 * cov(X, Y))
+   return(alpha)
+ }
>
> for(i in 1:B) {
+   # AS QUESTION REQUIRED: remove possible of getting same index
+   repeat{
+     index = sample(B, replace = T)
+     if(length(unique(index)) != 1)
+       break
+   }
+   bt[[i]] = data[index, ]
+   alpha[i] = alphaValue(bt[[i]])
+   rm(index)
+ }
>
> alphaSE = sd(alpha)
> alphaSE
[1] 0
```

The standard error of $\hat{\alpha}$ is 0, given that random seed is 4001.

Question 4

a) $L(\boldsymbol{\beta}) = \prod \frac{e^{(\beta_0+\beta_1 x_i)y_i}}{1+e^{\beta_0+\beta_1 x_i}}$, given that $\beta_0 = c_0$, we have

$$L(\beta_1) = \prod \frac{e^{(c_0+\beta_1 x_i)y_i}}{1+e^{c_0+\beta_1 x_i}}$$

$$l(\beta_1) = \sum\left[(c_0+\beta_1 x_i)y_i - \ln\left(1+e^{c_0+\beta_1 x_i}\right)\right]$$

$$\begin{aligned} l'(\beta_1) &= \sum\left[x_i y_i - \frac{e^{c_0+\beta_1 x_i}\cdot x_i}{1+e^{c_0+\beta_1 x_i}}\right] \\ &= \sum x_i\left[y_i - \frac{e^{c_0+\beta_1 x_i}}{1+e^{c_0+\beta_1 x_i}}\right] \\ &= \sum x_i[y_i - \Pr(y_i=1|x_i)] \end{aligned}$$

$$\begin{aligned} l''(\beta_1) &= -\sum x_i \frac{e^{c_0+\beta_1 x_i}\cdot x_i(1+e^{c_0+\beta_1 x_i}) - e^{c_0+\beta_1 x_i}\cdot x_i\cdot e^{c_0+\beta_1 x_i}}{\left(1+e^{c_0+\beta_1 x_i}\right)^2} \\ &= -\sum x_i^2 \frac{e^{c_0+\beta_1 x_i}}{1+e^{c_0+\beta_1 x_i}}\cdot\frac{1}{1+e^{c_0+\beta_1 x_i}} \\ &= -\sum x_i^2 \Pr(y_i=1|x_i)\Pr(y_i=0|x_i) \end{aligned}$$

$$\begin{aligned} \beta_1^{(t+1)} &= \beta_1^{(t)} + \Delta l'\left[\beta_1^{(t)}\right] \\ &= \beta_1^{(t)} - \frac{l'(\beta_1)}{l''(\beta_1)} \end{aligned}$$

b) Please refer to the following console of output (see Appendix for your reference)

```
> likelihood = function(x, y, beta0, beta1) {
+   prod(exp((beta0 + beta1 * x) * y) / (1 + exp(beta0 + beta1 * x)))
+ }
>
> logLikeli = function(x, y, beta0, beta1) {
+   sum((beta0 + beta1 * x) * y - log(1 + exp(beta0 + beta1 * x)))
+ }
>
> logLikeli_prime = function(x, y, beta0, beta1) {
+   sum(x * (y - 1 / (1 + exp(-(beta0 + beta1 * x)))))
+ }
>
> logLikeli_pprime = function(x, y, beta0, beta1) {
+   -sum(x ^ 2 * 1 / (1 + exp(beta0 + beta1 * x)) * 1 / (1 + exp(-(beta0 + beta1 * x))))
+ }
>
> logReg = function(x, y, beta0, beta1 = runif(1), f) {
+   repeat {
+     beta1_new = beta1 - logLikeli_prime(x, y, beta0, beta1) / logLikeli_pprime(x, y,
beta0, beta1)
+     if(abs(likelihood(x, y, beta0, beta1_new) - likelihood(x, y, beta0, beta1)) <= f) {
+       beta1 = beta1_new
+       break
+     }
+     beta1 = beta1_new
+   }
+   return(c(beta1, logLikeli(x, y, beta0, beta1)))
+ }
```

c) Please refer to the following console of output (see Appendix for your reference)

```
> set.seed(4001)
> logReg(data1$x, data1$y, beta0 = -0.66, f = 1e-14)
[1]   0.6937058 -27.7916111
```

The trace of $\hat{\beta}_1$ is 0.6937 and the trace of the log-likelihood is -27.7916, given that random seed is 4001.

d) Please refer to the following console of output (see Appendix for your reference)

```
> set.seed(4001)
> logReg(data2$x2, data2$y2, beta0 = 0, f = 1e-4)
[1]  3.078936e+02 -3.758561e-05
```

The trace of $\hat{\beta}_1$ is 307.8936 and the trace of the log-likelihood is -3.7586×10^{-5}, given that random seed is 4001.

e)

i. $L(\boldsymbol{\beta}) = \prod \frac{e^{[\beta_0+\beta_1(x_i+c)]y_i}}{1+e^{\beta_0+\beta_1(x_i+c)}}$, given that $\beta_0 = 0$, we have

$$L(\beta_1) = \prod \frac{e^{\beta_1(x_i+c)y_i}}{1+e^{\beta_1(x_i+c)}}$$

$$l(\beta_1) = \sum\{\beta_1(x_i+c)y_i - \ln[1+e^{\beta_1(x_i+c)}]\}$$

$$\begin{aligned} l'(\beta_1) &= \sum\left[(x_i+c)y_i - \frac{e^{\beta_1(x_i+c)}\cdot(x_i+c)}{1+e^{\beta_1(x_i+c)}}\right] \\ &= \sum(x_i+c)\left[y_i - \frac{e^{\beta_1(x_i+c)}}{1+e^{\beta_1(x_i+c)}}\right] \\ &= \sum(x_i+c)[y_i - \Pr(y_i=1|x_i)] \end{aligned}$$

$$\begin{aligned} l''(\beta_1) &= -\sum(x_i+c)\frac{e^{\beta_1(x_i+c)}\cdot(x_i+c)\left[1+e^{\beta_1(x_i+c)}\right] - e^{\beta_1(x_i+c)}\cdot(x_i+c)\cdot e^{\beta_1(x_i+c)}}{\left[1+e^{\beta_1(x_i+c)}\right]^2} \\ &= -\sum(x_i+c)^2\frac{e^{\beta_1(x_i+c)}}{1+e^{\beta_1(x_i+c)}}\cdot\frac{1}{1+e^{\beta_1(x_i+c)}} \\ &= -\sum(x_i+c)^2\Pr(y_i=1|x_i)\Pr(y_i=0|x_i) \end{aligned}$$

$$\begin{aligned} \beta_1^{(t+1)} &= \beta_1^{(t)} + \Delta l'\left[\beta_1^{(t)}\right] \\ &= \beta_1^{(t)} - \frac{l'(\beta_1)}{l''(\beta_1)} \end{aligned}$$

ii. Please refer to Part b. Although the objective function changed, by substitute $\beta = 0$ and $x_i = x_i + 0.5$, we are still able to reuse the same function here.

iii. Please refer to the following console of output (see Appendix for your reference)

```
> set.seed(4001)
> logReg(data2$x2 + 0.5, data2$y2, beta0 = 0, f = 1e-8)
[1]  3.951730e+01 -5.104804e-09
```

The trace of $\hat{\beta}_1$ is 39.5173 and the trace of the log-likelihood is -5.1048×10^{-9}, given that random seed is 4001.

f) The decision boundary is defined at $\Pr(y_i = 1|x_i) = \Pr(y_i = 0|x_i) \Longrightarrow \frac{e^{\beta_0+\beta_1x_i}}{1+e^{c_0+\beta_1x_i}} = \frac{1}{1+e^{c_0+\beta_1x_i}}$, such that $x = -\frac{\beta_0}{\beta_1}$. Thus, we have $x = 0$ for part (d) and $x = -0.5$ for part (e)(iii) which are their corresponding decision boundaries.

Question 5

a) Please refer to the following console of output (see Appendix for your reference)

```
> knn = function(x, y, new, K) {
+   distance = rep(0, length(y))
+   for(i in 1:length(y)) {
+     distance[i] = sqrt(sum((new - x[i, ]) ^ 2))
+   }
+   rank = rank(distance)
+   data = data.frame(rank, y)
+   data = data[order(rank),][1:K,]
+   if(mean(data[, 2]) > 0.5)
+     return(1)
+   else
+     return(0)
+ }
```

b) Please refer to the following console of output (see Appendix for your reference)

```
> knn(data4$x, data4$y, data4$x_new, K = 8)
[1] 0
```

The predicted label for x_{new} is 0, given that K is 8.

Question 6

We select a model with the minimum training error over the whole dataset.

```
> library(ISLR)
> library(boot)
> library(MASS)
> library(class)
>
> train = (Weekly$Year < 2009)
> Direction = Weekly$Direction[!train]
> test = Weekly[!train, ]
>
> # all possible models
> modelBoolean = expand.grid(c(TRUE,FALSE), c(TRUE,FALSE), c(TRUE,FALSE), c(TRUE,FALSE),
c(TRUE,FALSE), c(TRUE,FALSE))
> names(modelBoolean) = c(paste("Log", 1:5, sep = ""), "Volume")
> terms = names(Weekly)[2:7]
> model = apply(modelBoolean, 1, function(x) as.formula(paste(c("Direction ~ 1", terms[x]),
collapse = " + ")) )
>
> # select model
> cv = rep(1, 2 ^ length(terms))
> for(i in 1:(2 ^ length(terms))) {
+   set.seed(4001)
+   logReg = glm(model[[i]], data = Weekly, family = binomial)
+   cv[i] = cv.glm(Weekly, logReg, K = 10)$delta[1]
+   if(cv[i] == min(cv))
+     flag = i
+ }
> model_terms = strsplit(as.character(model[[flag]])[3], " ")
> model_terms = model_terms[[1]][model_terms[[1]] != "1" & model_terms[[1]] != "+"]
> model_terms
[1] "Lag2"
```

By cross-validation, the model contains variable "Lag2" has the minimum training error using logistic regression model. For the consistency of the models evaluation, we will only use this term to apply with the following models.

a) Please refer to the following console of output (see Appendix for your reference)

```
> logReg = glm(model[[flag]], data = Weekly, family = binomial, subset = train)
> logReg_prob = predict(logReg, test, type = "response")
> logReg_pred = rep("Down", length(logReg_prob))
> logReg_pred[logReg_prob > 0.5] = "Up"
>
> logReg

Call:  glm(formula = model[[flag]], family = binomial, data = Weekly,
    subset = train)

Coefficients:
(Intercept)         Lag2
     0.2033       0.0581

Degrees of Freedom: 984 Total (i.e. Null);  983 Residual
Null Deviance:          1355
Residual Deviance: 1351   AIC: 1355
> mean(logReg_pred == Direction)
[1] 0.625
```

The estimates are $\hat{\beta}_0 = 0.2033$ and $\hat{\beta}_1 = 0.0581$. We test the fitted model with testing data which observed after 2009. The error of predicting the label is 0.375.

b) Please refer to the following console of output (see Appendix for your reference)

```
> lda = lda(model[[flag]], data = Weekly, subset = train)
> lda_prod = predict(lda, test)
>
> lda
Call:
lda(model[[flag]], data = Weekly, subset = train)

Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843

Group means:
            Lag2
Down -0.03568254
Up    0.26036581

Coefficients of linear discriminants:
           LD1
Lag2 0.4414162
> mean(lda_prod$class == Direction)
[1] 0.625
```

The estimates are $\hat{\mu}_{up} = 0.2604$, $\hat{\mu}_{down} = -0.0357$, $\hat{\pi}_{up} = 0.4477$ and $\hat{\pi}_{down} = 0.5523$. The predicted error is 0.375.

c) Please refer to the following console of output (see Appendix for your reference)

```
> qda = qda(model[[flag]], data = Weekly, subset = train)
> qda_prod = predict(qda, test)
>
> qda
Call:
qda(model[[flag]], data = Weekly, subset = train)

Prior probabilities of groups:
     Down        Up 
0.4477157 0.5522843 

Group means:
            Lag2
Down -0.03568254
Up    0.26036581
> mean(qda_prod$class == Direction)
[1] 0.5865385
```

The estimates are $\hat{\mu}_{up} = 0.2604$, $\hat{\mu}_{down} = -0.0357$, $\hat{\pi}_{up} = 0.4477$ and $\hat{\pi}_{down} = 0.5523$. The predicted error is 0.4134.

d) Please refer to the following console of output (see Appendix for your reference)

```
> accuracy = rep(0, 20)
> for(i in 1:10) {
+   set.seed(4001)
+   knn = knn(as.matrix(Weekly[, model_terms][train]), as.matrix(Weekly[,
model_terms][!train]), Weekly$Direction[train], k = i)
+   accuracy[i] = mean(knn==Direction)
+   if(accuracy[i] == max(accuracy))
+     flag = i
+ }
> accuracy[flag]
[1] 0.6153846
>
> accuracy = rep(0, 20)
> X = scale(Weekly[, model_terms])
> for(i in 1:10) {
+   set.seed(4001)
+   knn = knn(as.matrix(X[train]), as.matrix(X[!train]), Weekly$Direction[train], k = i)
+   accuracy[i] = mean(knn==Direction)
+   if(accuracy[i] == max(accuracy))
+     flag = i
+ }
> accuracy[flag]
[1] 0.6153846
```

We simulate from $K = 1$ to $K = 20$ for finding the minimum test error. Also, we apply standardisation to KNN for reducing the variation due to X. Both non-standardised and standardised KNN share the same predicted error which is 0.3846.

By the predicted error, we can rank the performance of models as follow

$$\text{Logistic regression} = \text{LDA} > \text{KNN} > \text{QDA}$$

The above result is biased to the model with 1 variable "Lag2" only. Other model with different can be investigated in the future.

Appendix

```
## Question 2
library(boot)
x = c(0.4, 0.8, 1.2)
y = c(1.2, 1.8, 3.2)
data = data.frame(x, y)
reg = glm(y ~ x)
loocv = cv.glm(data, reg)
loocv$delta[1]


## Question 3
set.seed(4001)
x = c(4.3, 2.1, 5.3)
y = c(2.4, 1.1, 2.8)
data = data.frame(x, y)
B = nrow(data)
bt = list(data, data, data)
alpha = rep(0, B)

alphaValue = function(data) {
  X = bt[[i]]$x
  Y = bt[[i]]$y
  alpha = (var(Y) - cov(X, Y)) / (var(X) + var(Y) - 2 * cov(X, Y))
  return(alpha)
}

for(i in 1:B) {
  # AS QUESTION REQUIRED: remove possible of getting same index
  repeat{
    index = sample(B, replace = T)
    if(length(unique(index)) != 1)
      break
  }
  bt[[i]] = data[index, ]
  alpha[i] = alphaValue(bt[[i]])
  rm(index)
}

alphaSE = sd(alpha)
alphaSE


## Question 4
# Part b
likelihood = function(x, y, beta0, beta1) {
  prod(exp((beta0 + beta1 * x) * y) / (1 + exp(beta0 + beta1 * x)))
}

logLikeli = function(x, y, beta0, beta1) {
  sum((beta0 + beta1 * x) * y - log(1 + exp(beta0 + beta1 * x)))
}

logLikeli_prime = function(x, y, beta0, beta1) {
  sum(x * (y - 1 / (1 + exp(-(beta0 + beta1 * x)))))
}

logLikeli_pprime = function(x, y, beta0, beta1) {
  -sum(x ^ 2 * 1 / (1 + exp(beta0 + beta1 * x)) * 1 / (1 + exp(-(beta0 + beta1 * x))))
}

logReg = function(x, y, beta0, beta1 = runif(1), f) {
  repeat {
    beta1_new = beta1 - logLikeli_prime(x, y, beta0, beta1) / logLikeli_pprime(x, y, beta0,
beta1)
    if(abs(likelihood(x, y, beta0, beta1_new) - likelihood(x, y, beta0, beta1)) <= f) {
      beta1 = beta1_new
      break
    }
    beta1 = beta1_new
  }
  return(c(beta1, logLikeli(x, y, beta0, beta1)))
}

# Part c
set.seed(4001)
logReg(data1$x, data1$y, beta0 = -0.66, f = 1e-14)

# Part d
set.seed(4001)
logReg(data2$x2, data2$y2, beta0 = 0, f = 1e-4)

# Part e
set.seed(4001)
logReg(data2$x2 + 0.5, data2$y2, beta0 = 0, f = 1e-8)


## Quesiton 5
knn = function(x, y, new, K) {
  distance = rep(0, length(y))
  for(i in 1:length(y)) {
    distance[i] = sqrt(sum((new - x[i, ]) ^ 2))
  }
  rank = rank(distance)
  data = data.frame(rank, y)
  data = data[order(rank),][1:K,]
  if(mean(data[, 2]) > 0.5)
    return(1)
  else
    return(0)
}

knn(data4$x, data4$y, data4$x_new, K = 8)


## Question 6
library(ISLR)
library(boot)
library(MASS)
library(class)

train = (Weekly$Year < 2009)
Direction = Weekly$Direction[!train]
test = Weekly[!train, ]

# all possible models
modelBoolean = expand.grid(c(TRUE,FALSE), c(TRUE,FALSE), c(TRUE,FALSE), c(TRUE,FALSE),
c(TRUE,FALSE), c(TRUE,FALSE))
names(modelBoolean) = c(paste("Log", 1:5, sep = ""), "Volume")
terms = names(Weekly)[2:7]
model = apply(modelBoolean, 1, function(x) as.formula(paste(c("Direction ~ 1", terms[x]),
collapse = " + ")) )

# select model
cv = rep(1, 2 ^ length(terms))
for(i in 1:(2 ^ length(terms))) {
  set.seed(4001)
  logReg = glm(model[[i]], data = Weekly, family = binomial)
  cv[i] = cv.glm(Weekly, logReg, K = 10)$delta[1]
  if(cv[i] == min(cv))
    flag = i
}
model_terms = strsplit(as.character(model[[flag]])[3], " ")
model_terms = model_terms[[1]][model_terms[[1]] != "1" & model_terms[[1]] != "+"]
model_terms

# Part a
logReg = glm(model[[flag]], data = Weekly, family = binomial, subset = train)
logReg_prob = predict(logReg, test, type = "response")
logReg_pred = rep("Down", length(logReg_prob))
logReg_pred[logReg_prob > 0.5] = "Up"

logReg
mean(logReg_pred == Direction)

# Part b
lda = lda(model[[flag]], data = Weekly, subset = train)
lda_prod = predict(lda, test)

lda
mean(lda_prod$class == Direction)

# Part c
qda = qda(model[[flag]], data = Weekly, subset = train)
qda_prod = predict(qda, test)

qda
mean(qda_prod$class == Direction)

# Part d
accuracy = rep(0, 20)
for(i in 1:10) {
  set.seed(4001)
  knn = knn(as.matrix(Weekly[, model_terms][train]), as.matrix(Weekly[, model_terms][!train]),
Weekly$Direction[train], k = i)
  accuracy[i] = mean(knn==Direction)
  if(accuracy[i] == max(accuracy))
    flag = i
}
accuracy[flag]

accuracy = rep(0, 20)
X = scale(Weekly[, model_terms])
for(i in 1:10) {
  set.seed(4001)
  knn = knn(as.matrix(X[train]), as.matrix(X[!train]), Weekly$Direction[train], k = i)
  accuracy[i] = mean(knn==Direction)
  if(accuracy[i] == max(accuracy))
    flag = i
}
accuracy[flag]
```