STAT 318/462 : Data Mining

Assignment 1

> **Student Name : Xiao Meng**
>
> **Student ID : 88211337**

1.

Verses a less flexible model,

- One of the **advantage** of more flexible model:

  It can generate a much wider range of possible shapes to estimate f, which leads a more accuracy of prediction.

- One of the **disadvantage** of more flexible model:

  It always contains more predictors, which makes model more complex so that it is harder to interpret the relationship between predictors and responses.

- When we have a goal of inference, a less flexible approach which makes the relationship easier to understand will be preferred.

2.

In this classification problem, for a lower testing error rate, we will choose a relatively **small** value of k. Because when we perform a k-nearest neighbor classification, as K grows, the model will become less flexible and produce a decision boundary that is close to linear. In this problem for which decision boundary is highly non-linear, a relatively small value of k will fit data well and lead a lower testing error rate.

3.

Proof Procedure:

- With definitions:

$$
\begin{aligned}
&Bias(\hat{f}\,(x_0)) = E(\hat{f}\,(x_0) - f(x_0)) \\
&Var(\hat{f}\,(x_0)) = E(\hat{f}\,(x_0)^2) - [E(\hat{f}\,(x_0))]^2 \\
&y_0 = f(x_0) + \epsilon \\
&E(\epsilon) = 0 \\
&Var(\epsilon) = E(\epsilon^2) - [E(\epsilon)]^2 = E(\epsilon^2)
\end{aligned}
$$

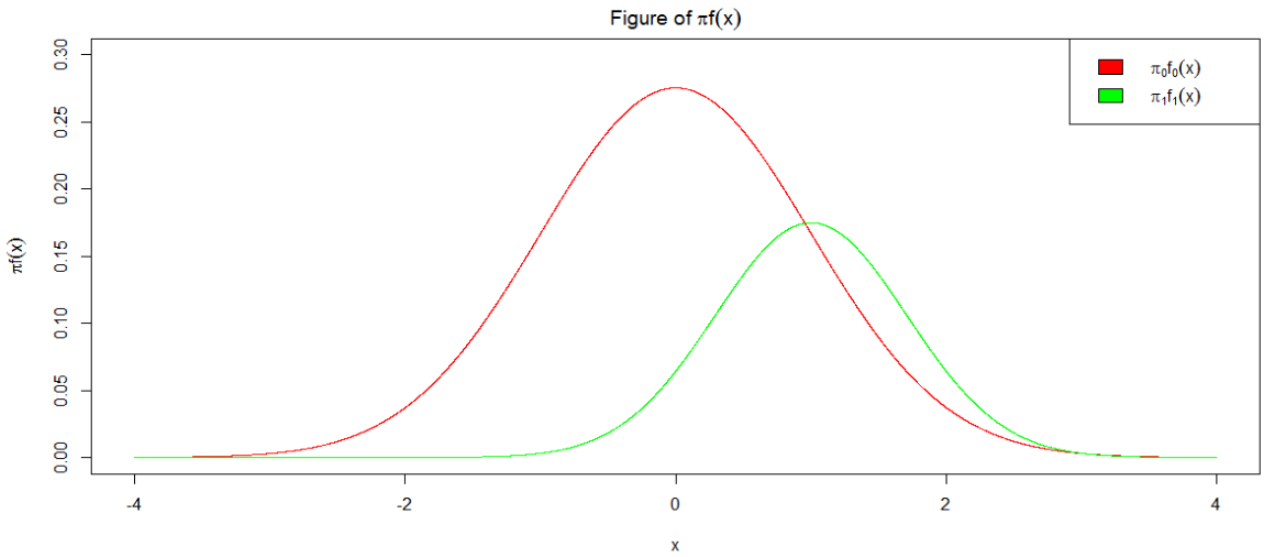
- We can get:

$$
\begin{aligned}
E(y_0 - \hat{f}(x_0))^2 &= E(f(x_0) + \epsilon - \hat{f}(x_0))^2 \\
&= E[(f(x_0) - \hat{f}^2(x_0) + \epsilon^2 - 2\epsilon(f(x_0) - \hat{f}(x_0))] \\
&= E[(f^2(x_0) + \hat{f}^2(x_0) - 2f(x_0)\hat{f}(x_0)] + E(\epsilon^2) - 2E(\epsilon)E(f(x_0) - \hat{f}(x_0)) \\
&= E(f^2(x_0)) + E(\hat{f}^2(fx_0)) - 2E(f(x_0))E(\hat{f}(x_0)) + E(\epsilon^2) \\
&= E(\hat{f}^2(x_0)) - [E(\hat{f}(x_0)]^2 + [E(\hat{f}(x_0)]^2 + E(f^2(x_0)) - 2E(f(x_0))E(\hat{f}(x_0)) + Var(\epsilon) \\
&= Var(\hat{f}(x_0)) + Var(\epsilon) + [E(\hat{f}(x_0)) - E(f(x_0))]^2 \\
&= Var(\hat{f}(x_0)) + Var(\epsilon) + [E(\hat{f}(x_0) - f(x_0))]^2 \\
&= Var(\hat{f}(x_0)) + Var(\epsilon) + [Bias(\hat{f}(x_0))]^2
\end{aligned}
$$

---

4. a)

Figure 1. $\pi_0 * f_0(x)$ and $\pi_1 * f_1(x)$:



b)

As it is a binary classification problem and $\pi_0 = 0.69$, therefore $\pi_1 = 1 - \pi_0 = 0.31$.

$$
\begin{aligned}
&\pi_0 f_0(x) = \pi_1 f_1(x) \\
&0.69\frac{1}{\sqrt{2\pi}}exp(-\frac{1}{2}x^2) = 0.31\frac{1}{\sqrt{\pi}}exp(-(x-1)^2) \\
&ln(\frac{0.69}{0.31\sqrt{2}}) - \frac{1}{2}x^2 + x^2 - 2x + 1 = 0 \\
&\frac{1}{2}x^2 - 2x + 1 + ln(\frac{0.69}{0.31\sqrt{2}}) \\
&x_1 = 3.0454(4dp) \\
&x_2 = 0.9546(4dp)
\end{aligned}
$$

The decision boundary are $x = 3.0454(4dp)$ and $x = 0.9546(4dp)$.

c)

With the decision boundary, we can know the class to observation of $X = 3$ is Class 1.

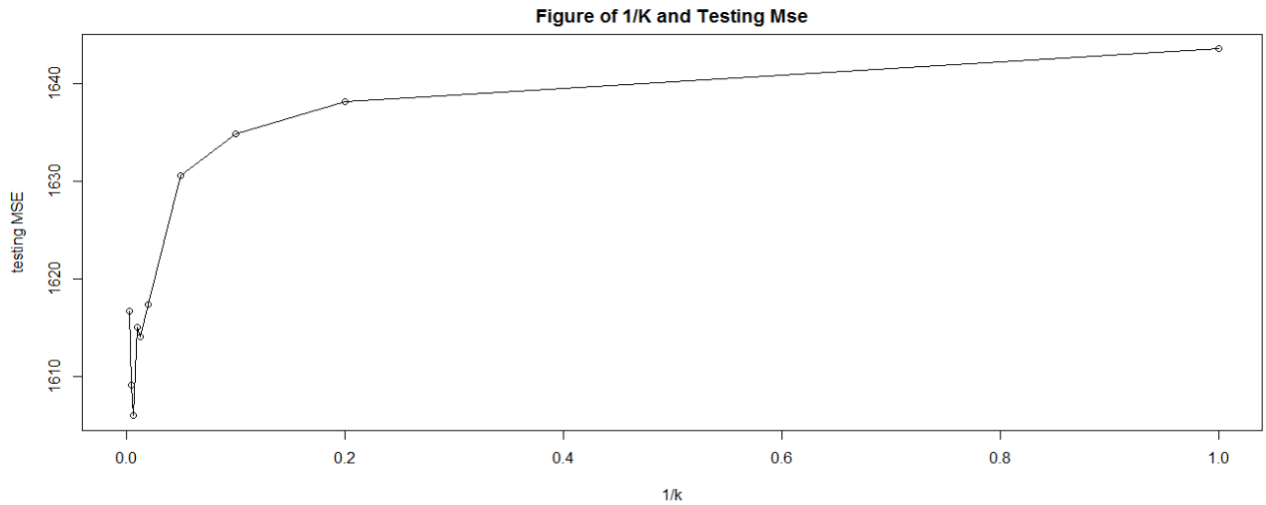
d)

$$
\begin{aligned}
Pr(Y=1|X=3) &= \frac{Pr(Y=1|X=3)}{Pr(Y=0|X=3)+Pr(Y=1|X=3)} \\
&= \frac{\pi_1 f_1(3)}{\pi_0 f_0(3)+\pi_1 f_1(3)} \\
&= 0.512
\end{aligned}
$$

5.

a)

Figure 2. KNN testing MSE with 1/K:



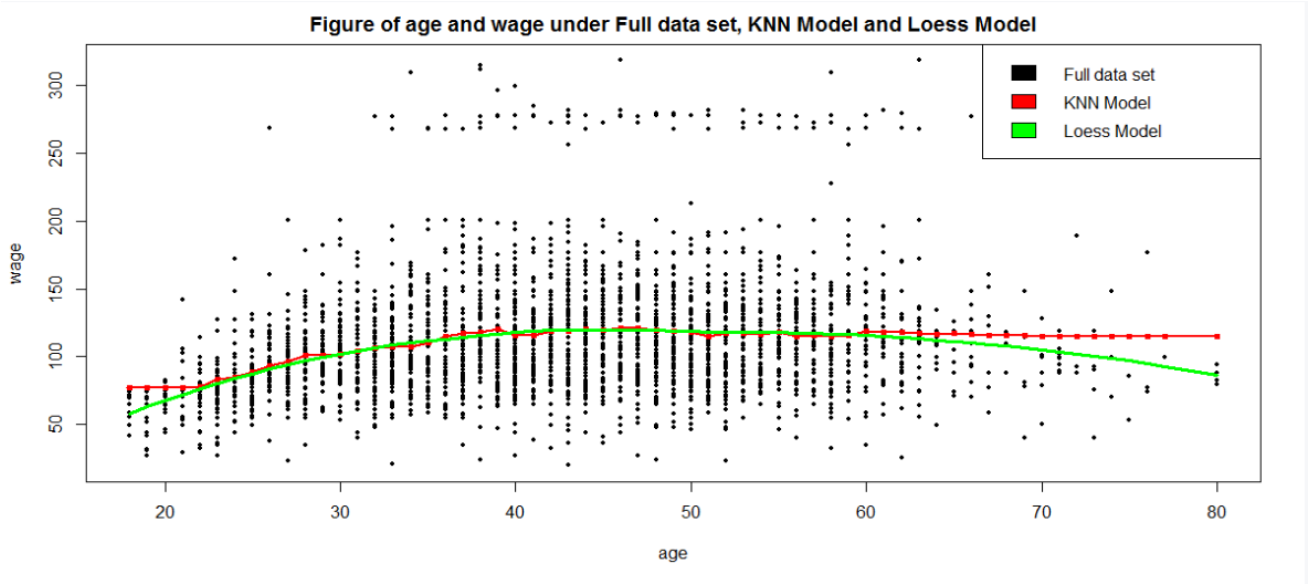
As the figure shown, with **K = 150** the testing MSE is the smallest. When the 1/k increases, the testing MSE goes as a U shape, which means the best value of K comes out in between. Because when K is very small, the model will over fit the training data, which will have a high variance. As K increases to larger than the best one, the model becomes more inflexible and has a high bias.

b)

The testing MSE of Loose Model is 1601.967.

c)

Figure 3. The Full data set, KNN Model and Loess Model of age and wage:



From Figure 3 we can get KNN Model with 150 neighborhoods and Loess Model of full data set. as shown, these two model almost get a similar trend of relationship between age and wage. when observations is sufficient, they both have a good performance. Compare to KNN Model, Loess Model can show the trend with a small number of observations, as the border of figure shows. Which KNN Model not perform well in these parts. That is because KNN Model is deeply influenced by the number of neighborhoods. Consider to have a good performance for the whole observation, we choose a value of neighborhood with the lowest testing MSE, which might sacrifice accuracy in some areas with infrequent observations. For example, in the age area between 60 and 80, if we choose 150 neighborhoods for each observation, it will be effected by large number of neighborhoods that are not in this area and bring a similar response with these 'out range' neighborhoods.

d)

With (a) we can get, when defining a neighborhood for KNN regression, if we choose the number of neighborhood(K value) is small, the model will lead a high variance and low bias. While the K value increases, the variance will become lower and bias increase. Consider bias-variance trade-off, we choose K value which is neither too small nor too big but with a lowest testing MSE to get a balance between variance and bias.

[Appendix]

- **Code of 4(a)**

```
x = seq(-4, 4, length=10000)
pi0 = 0.69
pi1 = 0.31
fx0 = pi0*dnorm(x, 0, 1)
fx1 = pi1*dnorm(x, 1, sqrt(0.5))
plot(x, fx0,
     type="l",
     ylab=expression(pi*f(x)),
     main=expression("Figure of "*pi*f(x)),
     xlim=c(-4, 4),
     ylim=c(0, .3),
     col="red")
```

```
lines(x, fx1, col="green")
legend("topright",
       c(expression(pi[0]*f[0](x)), expression(pi[1]*f[1](x))),
       fill=c("red", "green"))
```

- Code of 5(a)

```
#read original training data and testing data from csv files:
x.train = read.csv("C:/Users/Administrator/Desktop/WageTrain.csv", header=T)
x.pred = read.csv("C:/Users/Administrator/Desktop/WageTest.csv", header=T)

#initialise:
k = c(1, 5, 10, 20, 50, 75, 100, 150, 200, 300)
y.pred_k = list()
k_MSE = list()

#loop for each value of k, return testing MSE:
for (i in k){
  y.pred = kNN(i, x.train$age, x.train$wage, x.pred$age)
  y.pred_k = c(y.pred_k, list(y.pred))
  MSE = mean((y.pred - x.pred$wage)^2)
  k_MSE = c(k_MSE, list(MSE))
}

#plot a figure of testing MSE with each value of k:
plot(1/k, k_MSE,
     xlim=c(0, 1),
     ylab="testing MSE",
     main="Figure of 1/K and Testing Mse",
     type="o")
```

- Code of 5(b)

```
#fit the Loess model to training data to calculate the testing MSE:
loess_train = loess(x.train$wage~x.train$age, control=loess.control(surface="direct"))
loess_pred = predict(loess_train, newdata=x.pred$age)
loess_MSE = mean((loess_pred - x.pred$wage)^2)
loess_MSE
```

- Code of 5(c)

```
#define the full data set of x and y:
fulldata = rbind(x.train, x.pred)
plot(fulldata$age, fulldata$wage,
     xlab="age",
     ylab="wage",
     main="Figure of age and wage under Full data set, KNN Model and Loess Model",
     pch=20,
```

```
      cex=0.8)

#Best KNN Model:
knn_y = kNN(k=150, x.train$age, x.train$wage, fulldata$age)
knn_fd = data.frame(fulldata$age, knn_y)
knn_fd = knn_fd[order(knn_fd[,1]),]
points(knn_fd,
       col="red",
       type="o",
       pch=20,
       lwd=3)

#Loess Model:
loess_y = predict(loess_train, newdata=fulldata$age)
loess_fd = data.frame(fulldata$age, loess_y)
loess_fd = loess_fd[order(loess_fd[,1]),]
lines(loess_fd,
      col="green",
      lty=1,
      lwd=3)
legend("topright",
       c("Full data set", "KNN Model", "Loess Model"),
       fill=c("black", "red", "green"))
```