# ASSIGNMENT 1

(Chen (Ella) Liang, Student ID: 46275313)

1.

When a flexible model compared with a less flexible model,

The advantage is that the flexible model could learn the information of the dataset more effectively, capture the data features, and fit the data well.

The disadvantage is that when the model tries to fit all data including errors well, overfitting will be produced on training dataset, which would make testing MSE increase. At the same time, as the complexity of the model increases, its interpretability will decrease.

It is suitable to use a less flexible model when the dataset is too small, or the main target of building model is interpreting the relationship between responding variable and independent variables.

2.

If the decision boundary is highly non-linear, we could expect a lower testing error when a relatively small value of k is chosen. Since when k value is small, the model will try to fit the training data perfectly, which make the model more flexible and easily to be influenced by errors. Therefore, a highly non-linear boundary will be given.

3.

Known condition,

$$\mathrm{V}\left[\hat{f}(x_0)\right] = E\left[\hat{f}^2(x_0)\right] - E[\hat{f}(x_0)]^2$$

$$\mathrm{Bias}\left[\hat{f}(x_0)\right] = \mathrm{E}[\hat{f}(x_0) - f(x_0)]$$

$\mathrm{E}(y_0) = E[f(x_0) + \varepsilon] = E[f(x_0)] = f(x_0)$ ($f(x_0)$is the true value)

Proving Procedure,

$$\mathrm{E}\left[y_0 - \hat{f}(x_0)\right]^2 = E\left[y_0^2 + \hat{f}^2(x_0) - 2y_0\hat{f}(x_0)\right]$$

$$= \mathrm{E}[y_0^2] + E\left[\hat{f}^2(x_0)\right] + E[2y_0\hat{f}(x_0)]$$

$$= \mathrm{V}[y_0] + E[y_0]^2 + V\left[\hat{f}(x_0)\right] + E[\hat{f}(x_0)]^2 - 2f(x_0)E[\hat{f}(x_0)]$$

$$= \mathrm{V}[y_0] + V\left[\hat{f}(x_0)\right] + \left(f(x_0)^2 - 2f(x_0)E\left[\hat{f}(x_0)\right] + E\left[\hat{f}(x_0)\right]^2\right)$$

$$= \mathrm{V}[y_0] + V\left[\hat{f}(x_0)\right] + \left(f(x_0) - E\left[\hat{f}(x_0)\right]\right)^2$$

$$= V\left[\hat{f}(x_0)\right] + [Bias\left(\hat{f}(x_0)\right)]^2 + V(\varepsilon)$$

4.(a)

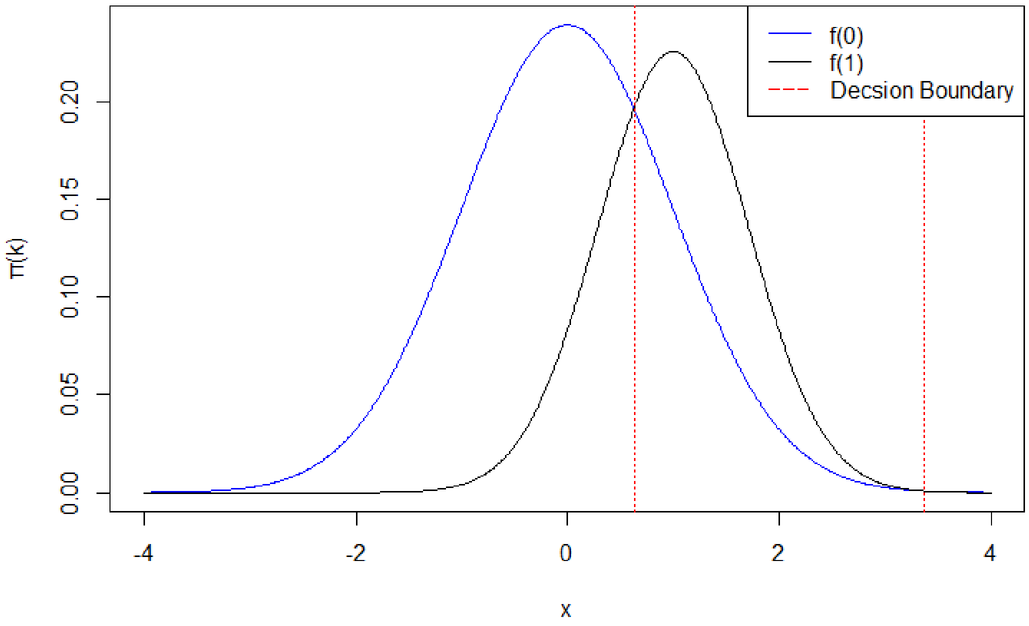

**Figure 1 Density Distribution and Decision Boundary of Class Y = 0 and Y = 1**

(b)

Since on the decision boundary,

$$\pi_0 f_0(x) = \pi_1 f_1(x)$$

which means,

$$\frac{0.6}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right) = \frac{0.4}{\sqrt{\pi}}\exp(-(x-1)^2)$$

$$\ln[\frac{3}{2\sqrt{2}}\exp\left(-\frac{1}{2}x^2\right)] = \ln[\exp(-(x-1)^2)]$$

$$\ln\frac{3}{2\sqrt{2}} - \frac{1}{2}x^2 = -(x-1)^2$$

$$\frac{1}{2}x^2 - 2x^2 + 1 + \ln\frac{3}{2\sqrt{2}} = 0$$

$$x_1 = 3.372,\ x_2 = 0.628\ \text{(3dp)}$$

Therefore, the decision boundary is x is equal to 3.372 and 0.628.

(c)

Using Bayes Boundary (shown as red dashed line in Figure 1), we know that when x is greater 3.372, x should belong to Y = 0.

(d)

$$\Pr(\mathrm{Y} = 1 | \mathrm{X} = 4) = \frac{f(x|Y = 1)\pi_1}{f(x|Y = 1)\pi_1 + f(x|Y = 0)\pi_0}$$

$$= \frac{\frac{0.4}{\sqrt{\pi}}\exp(-(x-1)^2)}{\frac{0.4}{\sqrt{\pi}}\exp(-(x-1)^2) + \frac{0.6}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right)}$$

$$= \frac{2.785e - 05}{8.030e - 05 + 2.785e - 05} = 0.2575$$

So, the probability that an observation with X = 4 in class 1 is 0.2575 (4dp).

5.

(a) After learning from training data, the scatter plots of testing MSE for each 1/k is shown as follows,

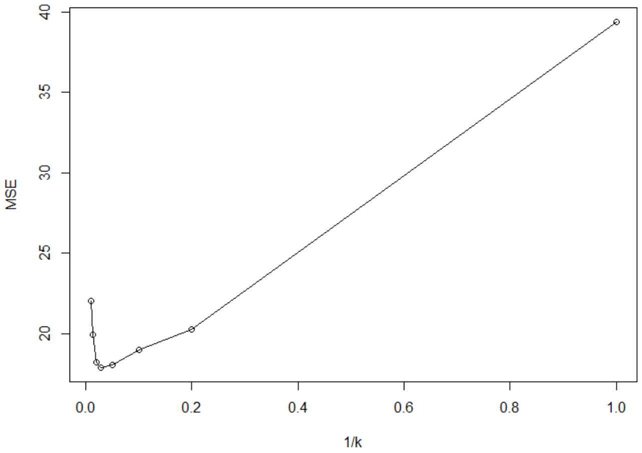

**Figure 2 scatter plot of testing MSE for each value of 1/k**

Figure 2 shows that the model perform the lowest testing MSE when k is equal to 35.

(b)

The testing MSE of loess model is 18.2985 (4dp). (calculated using R)

(c)

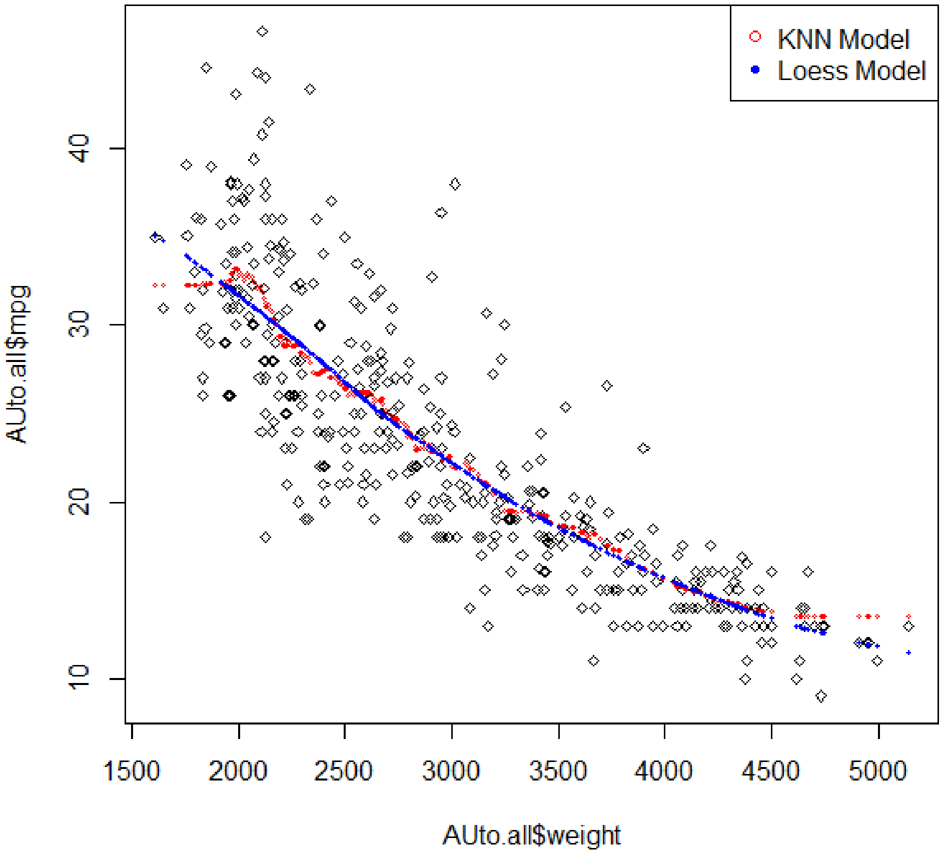

**Figure 4 KNN Model and Loess Model Prediction for Full Dataset**

Both of KNN model and loess model perform good in predicting the full database. However, the KNN model is less continuous and less smooth compared to Loess model. In addition, when weight is lower than 2000 and greater than 4500, points distribute dispersedly, the KNN model will be influenced more deeply because the nearest neighbours of close x-values are quite similar.

(d)

We adjust the bias-variance trade-off through changing k value to achieve a balance condition. When k value is small, the flexibility of the model is large and the model is quite similar with the training data, which means the bias is small as well as the variance is large. However, as k-value increases, the flexibility of the model will decrease. The modelling curve tends to be smooth, with large bias and small variance.

# Appendix: R Code

## R Code for Question 4

```
#QUESTION4#
##(a) ##
X <- seq(-4,4,0.01)
Y_X0 <- (0.6/sqrt(2*pi))*exp(-0.5*X^2)
Y_X1<- (0.4/sqrt(pi))*exp(-(X-1)^2)
plot(X,Y_X0,xlab='x',ylab='π(k)',type='l',col='blue')
points(X,Y_X1,type = 'l')

##(b) decision boundary##
sqrt(2-2*log(3/(2*sqrt(2))))+2  #3.371939
-(sqrt(2-2*log(3/(2*sqrt(2)))))+2  #0.6280609
abline(v=0.6280609,lty=3,col='red')
abline(v=3.371939,lty=3,col='red')
text.legend=c('f(0)','f(1)','Decision Boundary')
legend('topright',lty = c(1,1,5),legend=text.legend,col=c('blue','black','red'))

##(c)##
y0_x4<-(0.6/sqrt(2*pi))*exp(-0.5*4^2)
y1_x4<-(0.4/sqrt(pi))*exp(-(4-1)^2)
#X=4 belong to Y=0#

##(d)##
Pr_Y1_X4=y1_x4/(y1_x4+y0_x4)
Pr_Y1_X4
```

## R Code for Question 5

```
Auto.train<-read.csv('D:/2018 first semester/datamining/AutoTrain.csv',header = TRUE)
Auto.test<-read.csv('D:/2018 first semester/datamining/AutoTest.csv',header = TRUE)
AUto.all<-rbind(Auto.train,Auto.test)

##(a)##
## STAT318/462 kNN regression function
kNN <- function(k,x.train,y.train,x.pred) {
n.pred <- length(x.pred);                          y.pred <- numeric(n.pred)
## Main Loop
for (i in 1:n.pred){
  d <- abs(x.train - x.pred[i])
  dstar = d[order(d)[k]]
  y.pred[i] <- mean(y.train[d <= dstar])
}
## Return the vector of predictions
invisible(y.pred)
}
```

```
y_pred1<-kNN(1,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
y_pred5<-kNN(5,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
y_pred10<-kNN(10,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
y_pred20<-kNN(20,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
y_pred35<-kNN(35,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
y_pred50<-kNN(50,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
y_pred75<-kNN(75,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
y_pred100<-kNN(100,Auto.train$weight,Auto.train$mpg,Auto.test$weight)
MSE1<-mean((y_pred1-Auto.test$mpg)^2)
MSE5<-mean((y_pred5-Auto.test$mpg)^2)
MSE10<-mean((y_pred10-Auto.test$mpg)^2)
MSE20<-mean((y_pred20-Auto.test$mpg)^2)
MSE35<-mean((y_pred35-Auto.test$mpg)^2)
MSE50<-mean((y_pred50-Auto.test$mpg)^2)
MSE75<-mean((y_pred75-Auto.test$mpg)^2)
MSE100<-mean((y_pred100-Auto.test$mpg)^2)
MSE<-c(MSE1,MSE5,MSE10,MSE20,MSE35,MSE50,MSE75,MSE100)
k<-c(1,5,10,20,35,50,75,100)
plot(k,MSE,type='o')
plot(1/k,MSE,type='o')

##(b)##
model_train<-loess(mpg~weight,data = Auto.train,control=loess.control(surface="direct"))
loess_test<-predict(model_train,data.frame(weight = Auto.test$weight))
loess_MSE_dataframe<-data.frame((loess_test-Auto.test$mpg)^2)
loess_MSE<-mean(loess_MSE_dataframe[,1])

##(c)##
x11(width = 12,height = 12)
plot(AUto.all$weight,AUto.all$mpg,pch=5,cex=0.6)
y_pred35_all<-kNN(35,Auto.train$weight,Auto.train$mpg,AUto.all$weight)
points(AUto.all$weight,y_pred35_all,col='red',cex=0.4)
loess_all<-predict(model_train,data.frame(weight=AUto.all$weight))
points(loess_all~AUto.all$weight,col='blue',pch=20,cex=0.6)
text.legend=c('KNN Model','Loess Model')
legend('topright',pch = c(1,20),legend=text.legend,col=c('red','blue'))
```