Stats 503 Homework 1

Xingwen Wei, Veronica Zhao, Ruoyu Duan

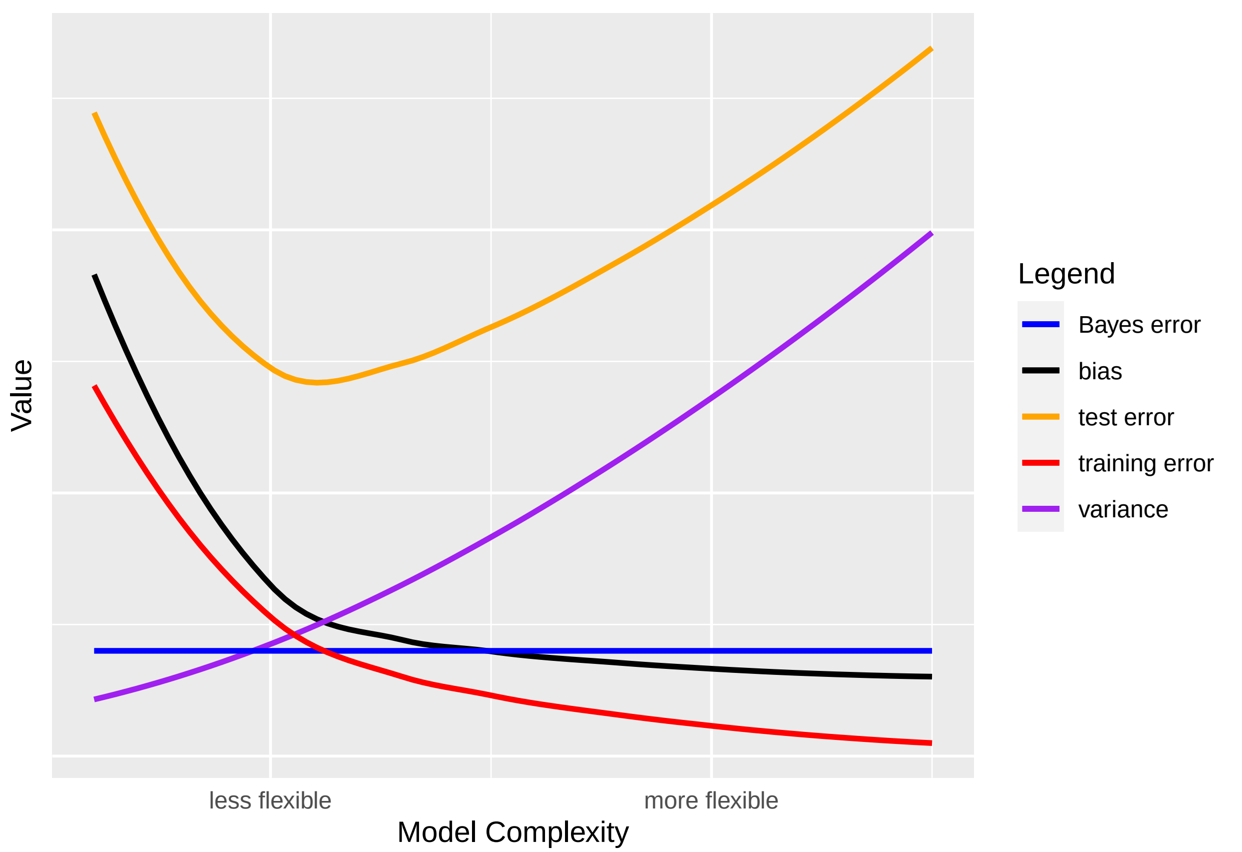
February 3, 2021

Question 1

```
library(ggplot2)

x <- c(1:20)
bias <- 1/x +0.1
variance <- x^1.5/100 + 0.1
Bayes_error <- 0.2
test_error <- bias + variance + Bayes_error
training_error <- x^(-0.5) - 0.2

ggplot(data=data.frame(x, bias, variance, Bayes_error, training_error, test_error)) +xlab("Model Comple
```



Question 2

a. When we have a large sized sample and a few predictors, we would expect the performance of a flexible method to be better. The model variance will be small for all model because of the large sample size. The bias will be smaller with a more flexible model. Since we cannot do anything about the Bayes error, we should choose a more flexible model in this situation to achieve a better test error.

b. When we have a small sized sample and many predictors, we would expect the performance of an inflexible method to be better. The model variance will be large for all model because of the small sample size. The bias will be smaller with a more flexible model. However, since the sample size is small, there is high chance that the sample is not representative and the model learned from this sample will be far from the true model even with high flexibility. Thus, we should choose an inflexible model in this situation to achieve a better test error.

c. When the predictor and response have a highly non-linear relationship, we would expect the performance of a flexible method to be better. We assume we have a large sample size because it will take a large sample to reach the conclusion that the relationship between predictor and response is not linear. The model variance will be small for all model because of the large sample size. The bias will be smaller with a more flexible model. Moreover, the non-linear relationship will be better represented in a more flexible model. Thus, we should choose a flexible model in this situation to achieve a better test error.

d. When the variance of the error term is large, we have to get a model not too flexible and not too inflexible. If we do not have a very large sample size, it is safer to use a inflexible model so that we do not pick the noise from the error terms as signal. On the other hand, if we have a very large sample size and the error term is normally distributed around zero, we can afford to apply a flexible model to reduce the bias while keeping the model variance low.

Question 3

a.

```
dist <- function(t, x){
  sqrt((t[1]-x[1])^2+(t[2]-x[2])^2+(t[3]-x[3])^2)
}

pred <- function(K){
  t <- table(obs[1:K,4])
  cat(sprintf("Prediction: %s", names(t[which.max(t)])))
}

test <- c(0,0,0)
c1 <- list(0,3,0, 'red')
c2 <- list(2,0,0, 'red')
c3 <- list(0,1,3, 'red')
c4 <- list(0,1,2, 'green')
c5 <- list(-1,0,1, 'green')
c6 <- list(1,1,2, 'red')
obs <- as.data.frame(rbind(c1, c2, c3, c4, c5, c6))
obs[4] <- as.factor(unlist(obs[4]))

dst <- c()

for(x in (1:6)){
  d <- dist(test, unname(unlist(obs[x,1:3])))
  dst[x] <- d
  cat(sprintf("Euclidean distance to observation %d is: %f \n", x, d))}
```

```
## Euclidean distance to observation 1 is: 3.000000
## Euclidean distance to observation 2 is: 2.000000
## Euclidean distance to observation 3 is: 3.162278
## Euclidean distance to observation 4 is: 2.236068
## Euclidean distance to observation 5 is: 1.414214
## Euclidean distance to observation 6 is: 2.449490
```

```r
obs <- cbind(obs, dst)
obs <- obs[order(obs[,5]),]

K <- 3

pred(K)
```

```
## Prediction: green
```

b. When we have large training set and nonlinear relationship between predictor and response, we expect the best value for K to be small. As mentioned above on Question 2c, we want a more flexible model to represent the nonlinear relationship while maintaining a low model variance because of the large training sample size.

Question 4

Data Preprocessing.

```r
train <- read.csv("C:/Users/xingw/Desktop/503/stats503/hw1/diabetes_train.csv", header=T)
test <- read.csv("C:/Users/xingw/Desktop/503/stats503/hw1/diabetes_test.csv", header=T)
```

To get familiar with the data, we want to see a data summary first. Noticing the outcome is a binary label, we set it as a factor. Also, after detecting apparent anomalies in attribute "Glucose", "BloodPressure", "SkinThickness", "Insulin", and "BMI", we replace the impossible 0's with NA. We have noted that about half of the observations for attribute "Insulin" and "SkinThickness" are missing, so we decided to not include this predictor in our model. Then we remove all the observations that is not complete.

```r
train$Outcome <- factor(train$Outcome)



summary(train)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.:103.0   1st Qu.: 64.00   1st Qu.: 0.00
##  Median : 3.000   Median :123.0   Median : 72.00   Median :22.50
##  Mean   : 4.054   Mean   :124.8   Mean   : 69.67   Mean   :20.07
##  3rd Qu.: 7.000   3rd Qu.:145.0   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :114.00   Max.   :99.00
##     Insulin            BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.00   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.00   1st Qu.:27.88   1st Qu.:0.2537           1st Qu.:25.00
##  Median :  0.00   Median :32.50   Median :0.4025           Median :31.00
##  Mean   : 84.07   Mean   :32.55   Mean   :0.5023           Mean   :34.33
##  3rd Qu.:130.00   3rd Qu.:36.80   3rd Qu.:0.6750           3rd Qu.:41.25
##  Max.   :846.00   Max.   :59.40   Max.   :2.4200           Max.   :81.00
##  Outcome
##  0:223
##  1:205
##
##
##
##
```

```r
train$Glucose[train$Glucose == 0] <- NA
train$BloodPressure[train$BloodPressure == 0] <- NA
train$SkinThickness[train$SkinThickness == 0] <- NA
train$Insulin[train$Insulin == 0] <- NA
```

```r
train$BMI[train$BMI == 0] <- NA


summary(train)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   : 44.0   Min.   : 30.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.:104.0   1st Qu.: 64.00   1st Qu.:22.00
##  Median : 3.000   Median :123.0   Median : 74.00   Median :30.00
##  Mean   : 4.054   Mean   :125.6   Mean   : 72.91   Mean   :29.73
##  3rd Qu.: 7.000   3rd Qu.:145.0   3rd Qu.: 80.00   3rd Qu.:36.00
##  Max.   :17.000   Max.   :199.0   Max.   :114.00   Max.   :99.00
##                   NA's   :3       NA's   :19       NA's   :139
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   : 14.0   Min.   :18.20   Min.   :0.0780           Min.   :21.00
##  1st Qu.: 92.0   1st Qu.:28.02   1st Qu.:0.2537           1st Qu.:25.00
##  Median :133.5   Median :32.55   Median :0.4025           Median :31.00
##  Mean   :171.3   Mean   :33.01   Mean   :0.5023           Mean   :34.33
##  3rd Qu.:199.0   3rd Qu.:36.88   3rd Qu.:0.6750           3rd Qu.:41.25
##  Max.   :846.0   Max.   :59.40   Max.   :2.4200           Max.   :81.00
##  NA's   :218     NA's   :6
##  Outcome
##  0:223
##  1:205
##
##
##
##
##
```

```r
train_c <- train[complete.cases(train[,-c(4, 5)]),-c(4, 5)]
```

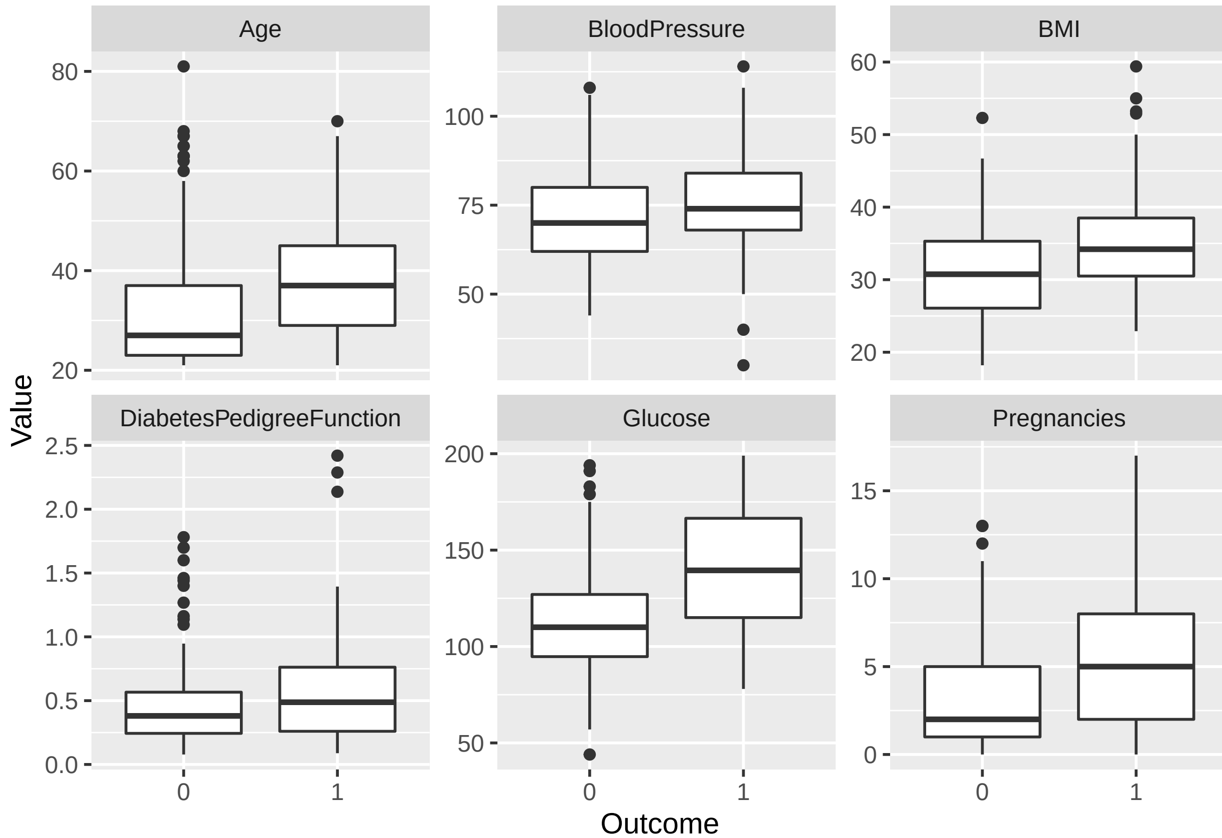
Exploratory Data Analysis. From the correlation matrix of variables, we do not find any obvious signs for collinearities.

```r
library(GGally)
```

In order to understand the relationship between each predictor variable and the response, we use boxplot for each predictor against the outcome. According to the boxplot, predictor variables "Pregnancies" and "Glucose" are more useful in predicting the outcome than other predictors.

```r
library(tidyr)

explore <- gather(data=train_c, key="Predictor", value="Value", -"Outcome")
ggplot(data=explore, aes(x=Outcome, y=Value)) + geom_boxplot() + facet_wrap(.~Predictor, scales='free_y
```



KNN.

```
library(mice)
library(class)

# Standardize train data
train_label <- train_c$Outcome
train_x <- train_c[-7]
train_mu <- colMeans(train_x)
train_std <- sqrt(diag(var(train_x)))
train_x <- scale(train_x, center=train_mu, scale=train_std)

# Preprocess test data
test$Glucose[test$Glucose == 0] <- NA
test$BloodPressure[test$BloodPressure == 0] <- NA
test$SkinThickness[test$SkinThickness == 0] <- NA
test$Insulin[test$Insulin == 0] <- NA
test$BMI[test$BMI == 0] <- NA
# Impute missing test data
test_temp <- mice(data=test[,-c(4, 5)], m=1, method='pmm')
```

```
##
##  iter imp variable
##   1   1  Glucose  BloodPressure  BMI
##   2   1  Glucose  BloodPressure  BMI
##   3   1  Glucose  BloodPressure  BMI
##   4   1  Glucose  BloodPressure  BMI
```

```
##   5   1  Glucose  BloodPressure  BMI
```

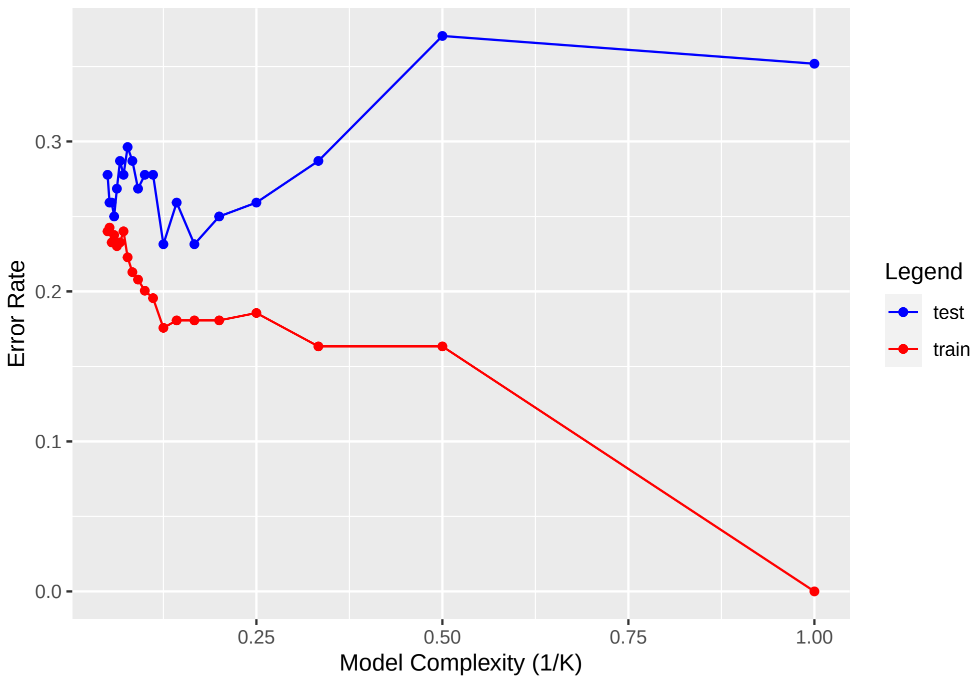
```r
test_c <- complete(test_temp)
# Standardize test data
test$Outcome <- factor(test$Outcome)
test_label <- test$Outcome
test_x <- test_c[-7]
test_x <- scale(test_x, center=train_mu, scale=train_std)

# KNN prediction
k_range <- 1:20
train_error <- c()
test_error <- c()

for(this_k in k_range){
  pred_train <- knn(train=train_x, test=train_x, cl=train_label, k=this_k)
  pred_test <- knn(train=train_x, test=test_x, cl=train_label, k=this_k)
  train_error[this_k] <- mean(pred_train != train_label)
  test_error[this_k] <- mean(pred_test != test_label)
  # Evaluation
  # table(pred, test_label)
}

result <- data.frame(train_error, test_error, k_range)
```

```r
library(ggplot2)
ggplot(result, aes(x=1/k_range)) + geom_line(aes(y=train_error, color="train")) + geom_point(aes(y=trai
```



In order to get most information, we use PMM to impute missing data in the test file after removing the two predictor variables not used in the training set.

```
## According to the KNN error plot, we would prefer K= 6
##  as it gives the minimum testing error 0.231481481481481.
```