STATS503 HW1

Wenjing Li

1/29/2020

Question 1

(a) The sample size n is extremely large, and the number of predictors p is small.

Since the sample size is large, we actually have enough data to get a more acurrate model. In this case, the number of predictors is relatively small compared to the number of the whole data set, so the performance of a flexible statistical learning method is better than an inflexible method. This is because the large data set makes the variance term relatively small, increasing the complexity of the model can help reduce the bias term.

(b) The number of predictors p is extremely large, and the number of observations n is small.

Since the number of predictors is relatively large compared to the number of the whole data set, it could be better to use a less complexible model and reduce the variance term. In this case, the performance of a flexible statistical learning method is worse than an inflexible method.

(c) The relationship between the predictors and response is highly non-linear.

Since the relationship is non-linear, using a simple linear model or even a qudratic model is not enough to describe the relationship between the response and the predictors. Simple models could have a extremely large bias term. In order to reduce the bias term, adding the model complexity is a good way. In this case, the performance of a flexible statistical learning method is better than an inflexible method.

(d) The variance of the error terms, i.e. $\sigma^2 = Var(e)$, is extremely high.

Since the variance is extremely high, we need to reduce the varience term by reducing the complexity of the model. Though the bias term could increase, compared to the decrease of the variance term, the increase of the bias term is relatively small. In this case, the performance of a flexible statistical learning method is worse than an inflexible method.

Question 2

In this problem, we use a diabetes data set and try to create models using KNN algorithm.

Data Manipulation

```
## upload training data and check data
dat = read.csv("C:/Users/wenji/Downloads/STATS 503/HW1/diabetes_train.csv")
dat$Outcome = factor(dat$Outcome > 1/2)
levels(dat$Outcome) = c("not having diabetes", "having diabetes")
summary(dat)
```

```
##   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
##  not having diabetes:223
##  having diabetes    :205
##
##
##
##
```

```
## upload test data and check data
dat_test = read.csv("C:/Users/wenji/Downloads/STATS 503/HW1/diabetes_test.csv")
dat_test$Outcome = factor(dat_test$Outcome > 1/2)
levels(dat_test$Outcome) = c("not having diabetes", "having diabetes")
summary(dat_test)
```

```
##   Pregnancies       Glucose      BloodPressure    SkinThickness
##  Min.   : 0.00   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.00   1st Qu.:106.5   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.00   Median :129.0   Median : 70.00   Median :25.00
##  Mean   : 3.75   Mean   :130.4   Mean   : 66.52   Mean   :21.15
##  3rd Qu.: 6.00   3rd Qu.:154.5   3rd Qu.: 78.50   3rd Qu.:33.00
##  Max.   :13.00   Max.   :198.0   Max.   :110.00   Max.   :46.00
##     Insulin            BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.00   Min.   : 0.00   Min.   :0.0840           Min.   :21.00
##  1st Qu.:  0.00   1st Qu.:28.40   1st Qu.:0.2517           1st Qu.:24.00
##  Median : 45.50   Median :32.60   Median :0.3925           Median :29.50
```

```
##  Mean   : 88.75   Mean   :33.04   Mean   :0.4748           Mean   :33.02
##  3rd Qu.:151.25   3rd Qu.:37.05   3rd Qu.:0.6338           3rd Qu.:39.25
##  Max.   :480.00   Max.   :67.10   Max.   :1.8930           Max.   :69.00
##                Outcome
##  not having diabetes:45
##  having diabetes    :63
##
##
##
##
```

Training data set:

There is no missing data. However, some data in the data set is impossible. Like no one could live with no glucose or insulin in his or her body. Also, BMI and diastolic blood pressure are impossible to be 0. Based on above, I omit all the false data, in case those data could make the model become less accurate.

Although it also looks very rare that people can get pregnant in 17 times, we don't have enough clue to delete these data.

Test data set:

For the test data set, we use the same process to manipulate data.

```
dat_clean = dat %>%
  filter(Glucose != 0 & BloodPressure != 0 & Insulin != 0 & BMI != 0)
summary(dat_clean)
```

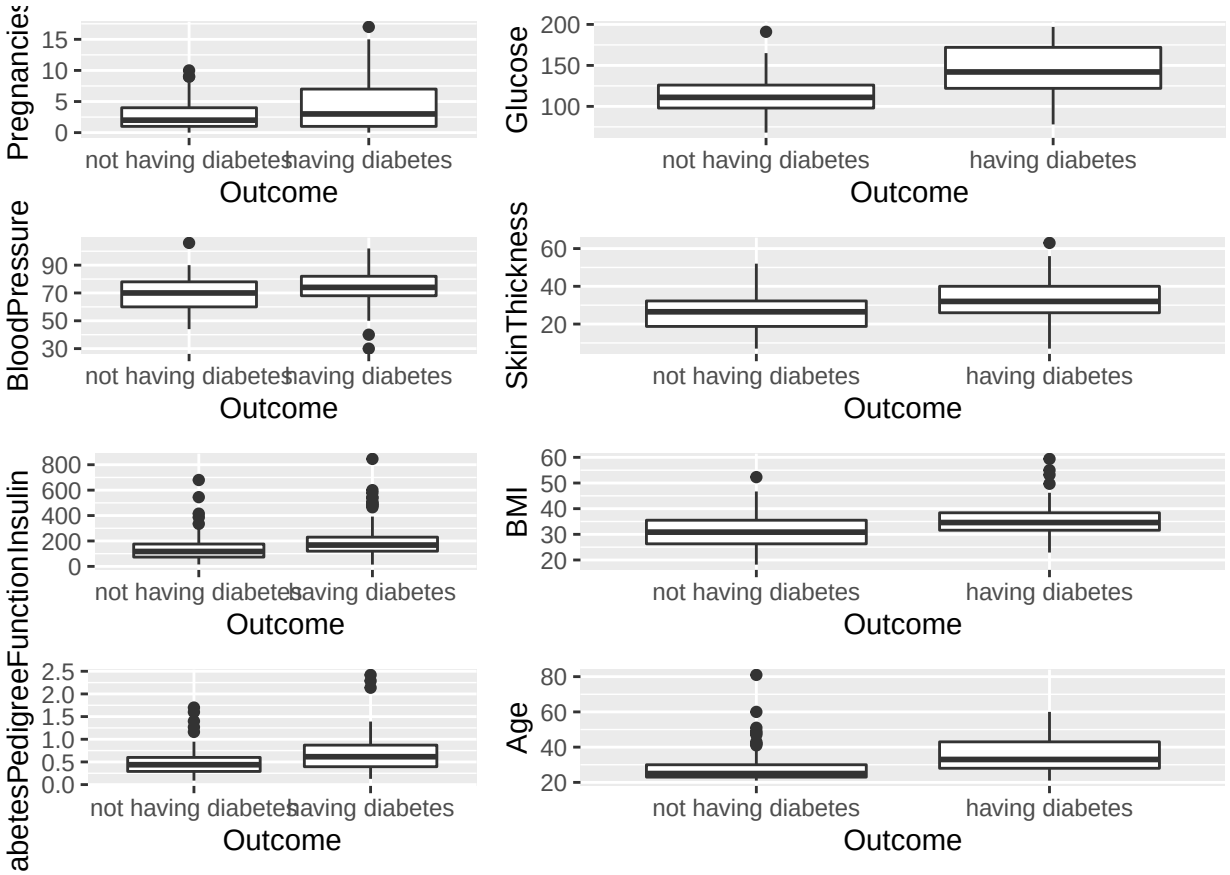
```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   : 68.0   Min.   : 30.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.:104.0   1st Qu.: 64.00   1st Qu.:22.00
##  Median : 2.000   Median :124.0   Median : 72.00   Median :29.00
##  Mean   : 3.483   Mean   :126.4   Mean   : 71.36   Mean   :29.31
##  3rd Qu.: 6.000   3rd Qu.:145.0   3rd Qu.: 80.00   3rd Qu.:36.00
##  Max.   :17.000   Max.   :197.0   Max.   :106.00   Max.   :63.00
##     Insulin          BMI        DiabetesPedigreeFunction      Age
##  Min.   : 14   Min.   :18.20   Min.   :0.0880           Min.   :21.00
##  1st Qu.: 92   1st Qu.:29.00   1st Qu.:0.3130           1st Qu.:24.00
##  Median :135   Median :33.30   Median :0.4970           Median :28.00
##  Mean   :172   Mean   :33.37   Mean   :0.5811           Mean   :31.82
##  3rd Qu.:200   3rd Qu.:37.50   3rd Qu.:0.7300           3rd Qu.:37.00
##  Max.   :846   Max.   :59.40   Max.   :2.4200           Max.   :81.00
##                Outcome
##  not having diabetes:112
##  having diabetes    : 97
##
##
##
##
```

```
dat_test_clean = dat_test %>%
  filter(Glucose != 0 & BloodPressure != 0 & Insulin != 0 & BMI != 0)
summary(dat_test_clean)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   : 75.0   Min.   : 48.00   Min.   :10.00
##  1st Qu.: 1.000   1st Qu.:108.8   1st Qu.: 62.00   1st Qu.:23.00
##  Median : 2.500   Median :137.0   Median : 70.00   Median :32.00
##  Mean   : 3.536   Mean   :135.1   Mean   : 72.23   Mean   :30.11
##  3rd Qu.: 5.000   3rd Qu.:164.8   3rd Qu.: 82.00   3rd Qu.:37.25
##  Max.   :13.000   Max.   :198.0   Max.   :110.00   Max.   :46.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   : 29.0   Min.   :19.60   Min.   :0.0850           Min.   :21.00
##  1st Qu.: 91.0   1st Qu.:28.77   1st Qu.:0.2805           1st Qu.:24.00
##  Median :147.5   Median :33.30   Median :0.3910           Median :29.00
##  Mean   :169.6   Mean   :33.87   Mean   :0.4551           Mean   :33.30
##  3rd Qu.:226.5   3rd Qu.:36.85   3rd Qu.:0.6268           3rd Qu.:42.25
##  Max.   :480.0   Max.   :67.10   Max.   :1.1890           Max.   :61.00
##                 Outcome
##  not having diabetes:23
##  having diabetes    :33
##
##
##
##
```

Generate plots using the cleaned data set

```
## generate plot
box1 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = Pregnancies))
box2 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = Glucose))
box3 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = BloodPressure))
box4 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = SkinThickness))
box5 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = Insulin))
box6 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = BMI))
box7 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = DiabetesPedigreeFunction))
box8 = ggplot(dat_clean) +
  geom_boxplot(aes(x = Outcome, y = Age))
grid.arrange(box1, box2, box3, box4, box5, box6, box7, box8,
             widths = c(4, 6))
```



From the box plots above, we can conclude that having diabetes or not highly related to the glucose concentration.

KNN model

After dealing with the data set, we can start working on the KNN model.

At first, select relative data from data set dat_clean and dat_test_clean

```
## select data
train_label = dat_clean %>% .$Outcome
test_label = dat_test_clean %>% .$Outcome
train_x = dat_clean %>%
  select('Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI',
         'DiabetesPedigreeFunction', 'Age')
test_x = dat_test_clean %>%
  select('Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI',
         'DiabetesPedigreeFunction', 'Age')

## scale training and test data in the same way
mean_train = colMeans(train_x)
std_train = sqrt(diag(var(train_x)))
train_x = scale(train_x, center = mean_train, scale = std_train)
test_x = scale(test_x, center = mean_train, scale = std_train)
```

Compute KNN model when k = 1, 2, 3, . . . , 20 and get training error and test error respectively.

```r
k_range = c(1:20)
train_error = c()
test_error = c()
for(i in 1:length(k_range)){
  pred_train <- knn(train_x, train_x, train_label, k = k_range[i])
  train_error[i] = mean(pred_train != train_label)
  pred_test = knn(train_x, test_x, train_label, k = k_range[i])
  test_error[i] = mean(pred_test != test_label)
}

## generate plot
errors = data.frame(train_error, test_error, k_range)
ggplot(errors, aes(x = k_range)) +
  geom_line(aes(y = train_error), col = "red") +
  geom_point(aes(y = train_error), col = "red") +
  geom_line(aes(y = test_error), col = "blue") +
  geom_point(aes(y = test_error), col = "blue") +
  ylab("Error Rate") + xlab("K") +
  ggtitle("Training and test error rate for KNN") +
  theme_minimal()
```