ITEC-620-001 FINAL PROJECT

THREE METHODS OF PREDICTING DIABETES

YU CHIEH CHENG
HE JIN CHU
CHUEH HSIEN LUO

DATASET INTRODUCTION

- Diabetes Dataset
- National Institute of Diabetes and Digestive and Kidney Diseases
- Observer all females and at least 21 years old of Pima Indian heritage
- Rows: 768 ; Columns: 9
- Total of 6,912 records

ISSUE BEING ANALYZED

- Predict based on diagnostic measurements whether a patient has diabetes
- Dependent variable: Outcome(1:malignant 0:benign)
- Independent variables: Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age

SNAPSHOT OF THE DATA SET

A tibble: 10 × 9

Pregnancies <dbl>	Glucose <dbl>	BloodPressure <dbl>	SkinThickness <dbl>	Insulin <dbl>	BMI <dbl>	DiabetesPedigreeFunction <dbl>	Age <dbl>	Outcome <dbl>
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31.0	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0.0	0.232	54	1

1–10 of 10 rows

Original dataset

A tibble: 6 × 8

Pregnancies <dbl>	Glucose <dbl>	BloodPressure <dbl>	SkinThickness <dbl>	Insulin <dbl>	BMI <dbl>	DiabetesPedigreeFunction <dbl>	Age <dbl>
9	89	62	0	0	22.5	0.142	33
10	101	76	48	180	32.9	0.171	63
2	122	70	27	0	36.8	0.340	27
5	121	72	23	112	26.2	0.245	30
1	126	60	0	0	30.1	0.349	47
1	93	70	31	0	30.4	0.315	23

6 rows

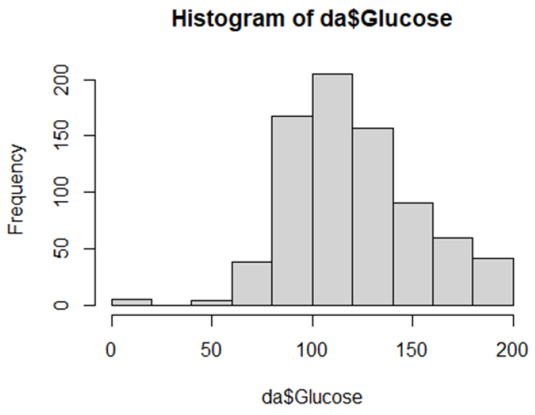
New data selected from original dataset

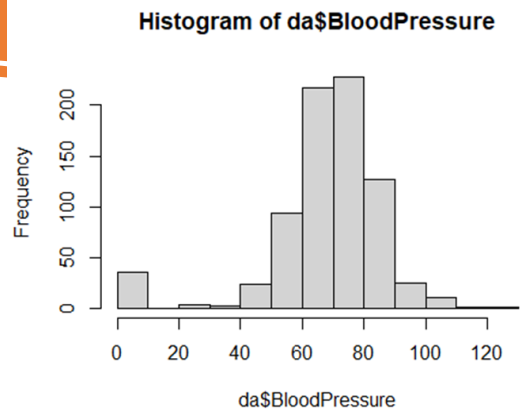
MISSING VALUES DETECTION

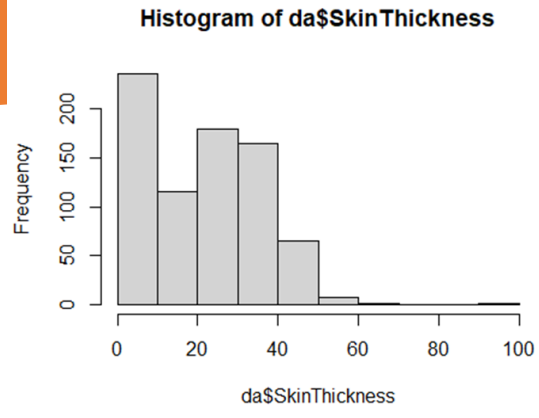
Using is.na function and colSums function

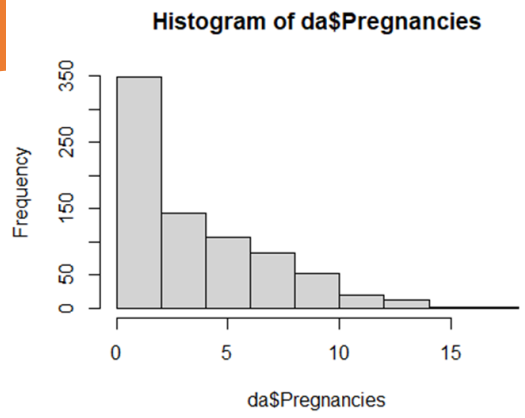
```
> colSums(is.na(da))
            Pregnancies                Glucose          BloodPressure          SkinThickness                Insulin
                      0                      0                      0                      0                      0
                    BMI DiabetesPedigreeFunction                    Age                Outcome
                      0                      0                      0                      0
```

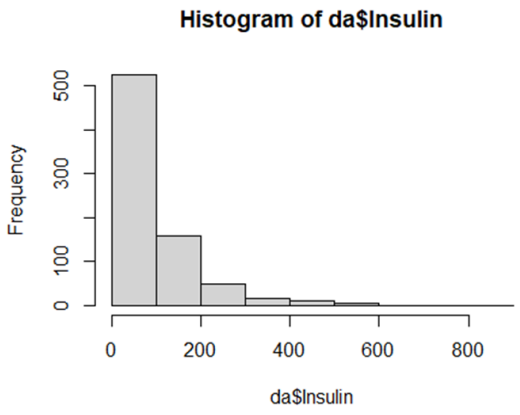
HISTOGRAM

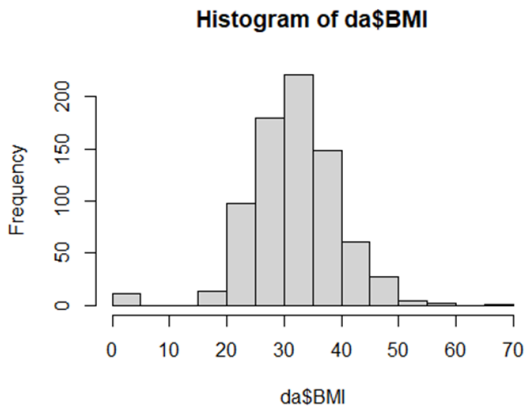

Histogram of da$Glucose
Frequency
0
50
100
150
200
0
50
100
150
200
da$Glucose

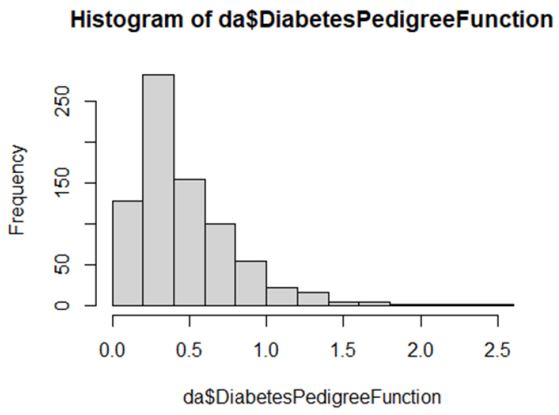

Histogram of da$BloodPressure
Frequency
0
50
100
150
200
0
20
40
60
80
100
120
da$BloodPressure

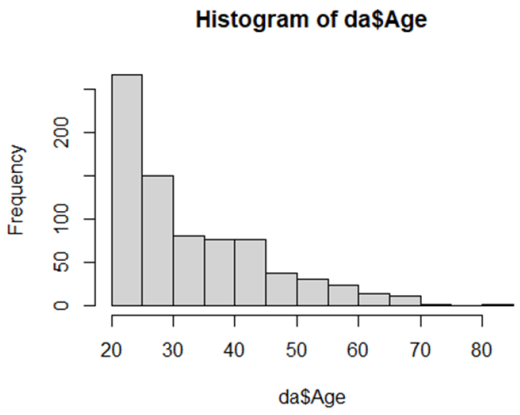

Histogram of da$SkinThickness
Frequency
0
50
100
150
200
0
20
40
60
80
100
da$SkinThickness


Histogram of da$Pregnancies
Frequency
0
50
150
250
350
0
5
10
15
da$Pregnancies

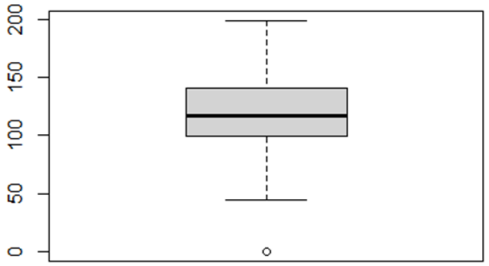

Histogram of da$Insulin
Frequency
0
100
300
500
0
200
400
600
800
da$Insulin


Histogram of da$BMI
Frequency
0
50
100
150
200
0
10
20
30
40
50
60
70
da$BMI

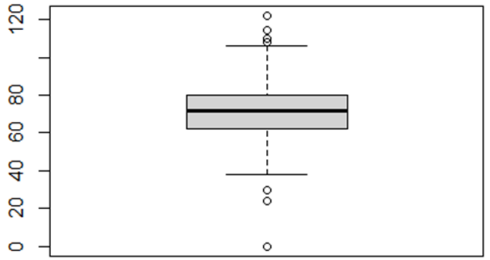

Histogram of da$DiabetesPedigreeFunction
Frequency
0
50
150
250
0.0
0.5
1.0
1.5
2.0
2.5
da$DiabetesPedigreeFunction


Histogram of da$Age
Frequency
0
50
100
200
20
30
40
50
60
70
80
da$Age

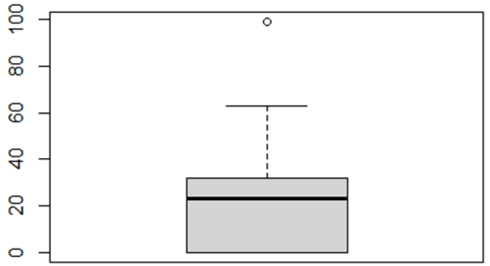
BOXPLOT



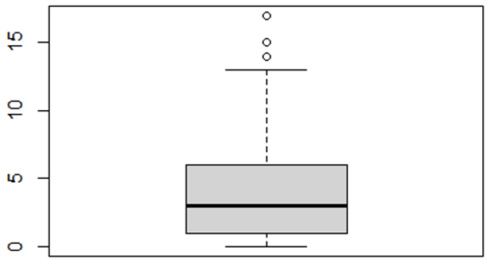
Glucose Boxplot



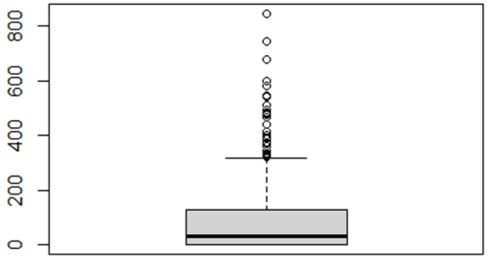
BloodPressure Boxplot



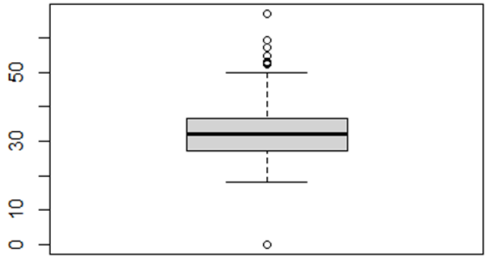
SkinThickness Boxplot



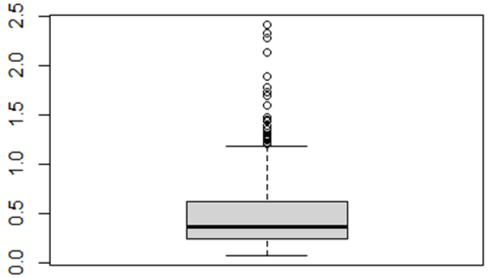
Pregnancies Boxplot



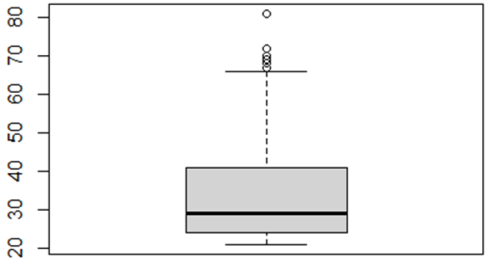
Insulin Boxplot



BMI Boxplot



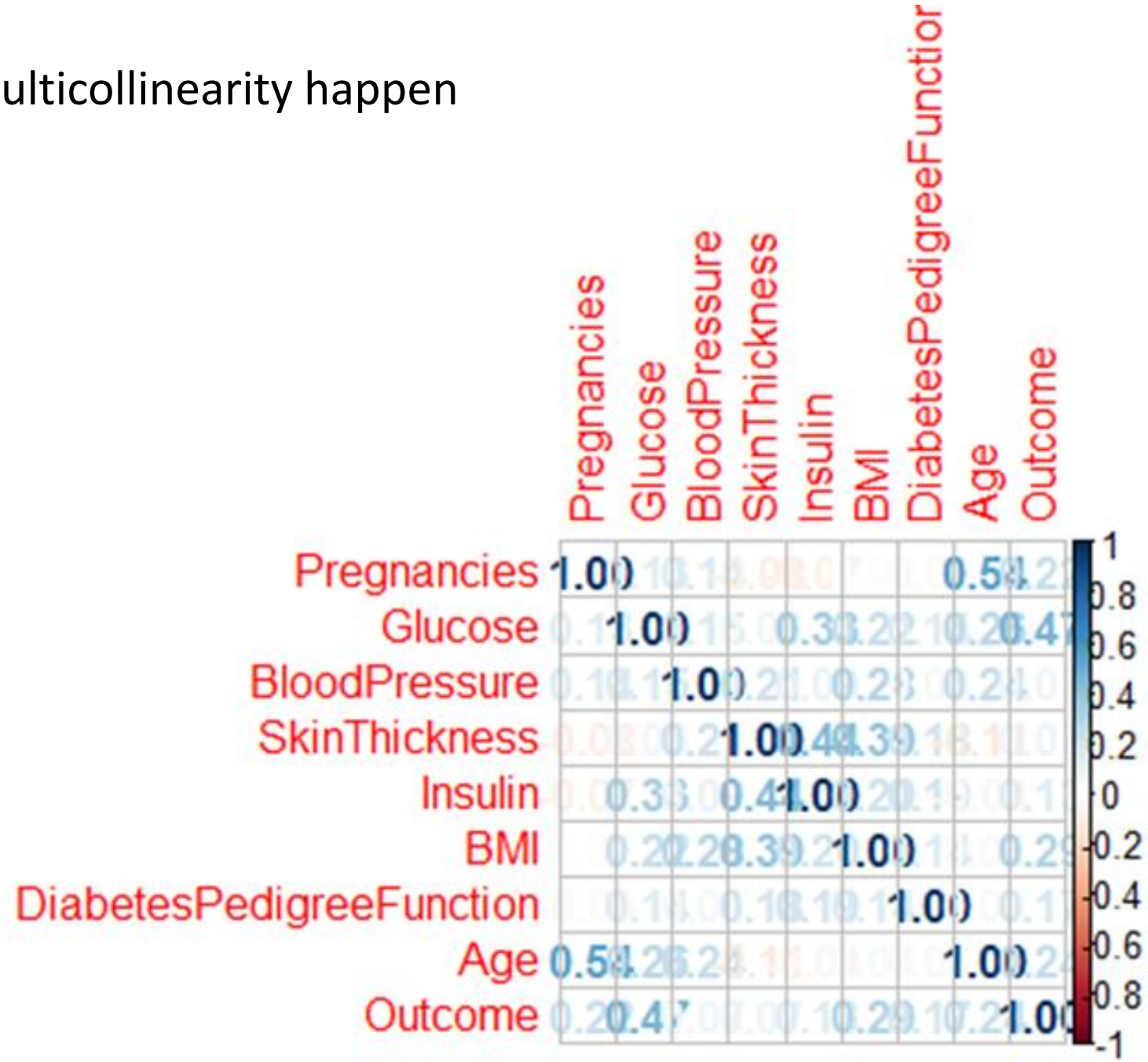
DiabetesPedigreeFunction Boxplot



Age Boxplot

CORRELATION BETWEEN INDEPENDENT VARIABLES

No multicollinearity happen



NORMALIZATION

```{r}
da.norm <- scale(da)
set.seed(12345)
```

Using scale function to normalize numeric data into z score and set.seed to make sure every output from randomness will be the same.

ANALYSIS METHOD 1 – KNN(1/2)

```{r}
da <- as.data.frame(da)
set.seed(12345)
training <- sample(1:nrow(da), 0.6*nrow(da))
ycol <- match('Outcome',colnames(da))
da.training <- da[training,-ycol]
da.training.results <- da[training,ycol] > 0.5
da.test <- da[-training,-ycol]
da.test.results <- da[-training,ycol] > 0.5

da.training.norm <- scale(da.training)
da.test.norm <- scale(da.test,center=attr(da.training.norm,"scaled:center")
                      [1:ncol(da.test)],scale=attr(da.training.norm,"scaled:scale")[1:ncol(da.test)])

da.knn <- knn.reg(da.training, da.test, da.training.results, k=5)
da.knn <- knn(da.training.norm, da.test.norm, da.training.results, k=100)
sum(da.knn == da.test.results) / length(da.test.results)  #accuracy rate
table(da.knn, da.test.results)   #confusion matrix
```

Diabetes .test.results		
da.knn	FALSE	TRUE
FALSE	189	77
TRUE	14	28

Accuracy rate: 0.7045455

ANALYSIS METHOD 1 – KNN(2/2)

```
> da.knn.new <- knn.reg(da.training.norm, danew.norm, da.training.results, k=100)
> da.knn.new$pred
[1] 0.23 0.51 0.20 0.23 0.34 0.09
```

Predict numbers: 0.23, 0.51, 0.20, 0.23, 0.34, 0.09

ANALYSIS METHOD 2 -- LOGISTIC REGRESSION(1/2)

```{r}
da.lr <- glm(Outcome ~ ., family=binomial(link='logit'),data=da[training,])
summary(da.lr)
```

```{r}
da.test.probabilities <- predict(da.lr,da.test,type = "response")
da.lr.classifications <- round(da.test.probabilities,0)
sum(da.lr.classifications == da.test.results) / length(da.test.results) #accuracy rate
table(da.lr.classifications,da.test.results)   #confusion matrix
```

Diabetes .test. results		
Classifications	FALSE	TRUE
0	177	41
1	26	64

Accuracy rate: 0.7824675

ANALYSIS METHOD 2 -- LOGISTIC REGRESSION(2/2)

```
> round(predict(da.lr,danew,type="response"),2)
   1    2    3    4    5    6
0.13 0.54 0.31 0.20 0.26 0.08
```

Predict numbers: 0.13, 0.54, 0.31, 0.20, 0.26, 0.08

ANALYSIS METHOD 3 -- CLASSIFICATION TREE(1/3)

```{r}
set.seed(12345)
training <- sample(1:nrow(da), 0.6*nrow(da))
ycol <- match('Outcome',colnames(da))
da.training <- da[training,-ycol]
da.training.results <- da[training,ycol] > 0.5
da.test <- da[-training,-ycol]
da.test.results <- da[-training,ycol] > 0.5
da.tree <- tree(Outcome ~ ., data=da[training,])
plot(da.tree)
text(da.tree)
da.tree <- tree(Outcome ~ ., data=da[training,], mindev=0.001)
plot(da.tree)
text(da.tree, cex=0.6)
da.tree.proportions <- predict(da.tree,da[-training,])
da.tree.classifications <- round(da.tree.proportions,0)
sum(da.tree.classifications == da.test.results) / nrow(da[-training,]) #accuracy rate
table(da.tree.classifications, da.test.results)   #confusion matrix
```

Diabetes .test.results		
da.tree.classifications	FALSE	TRUE
0	160	50
1	43	55

Accuracy rate: 0.6980519

ANALYSIS METHOD 3 -- CLASSIFICATION TREE(2/3)

```
> predict(da.tree,danew)
        1         2         3         4         5         6
0.0000000 0.1666667 0.0000000 0.0000000 1.0000000 0.0000000
```

Predict numbers: 0.00, 0.17, 0.00, 0.00, 1.00, 0.00

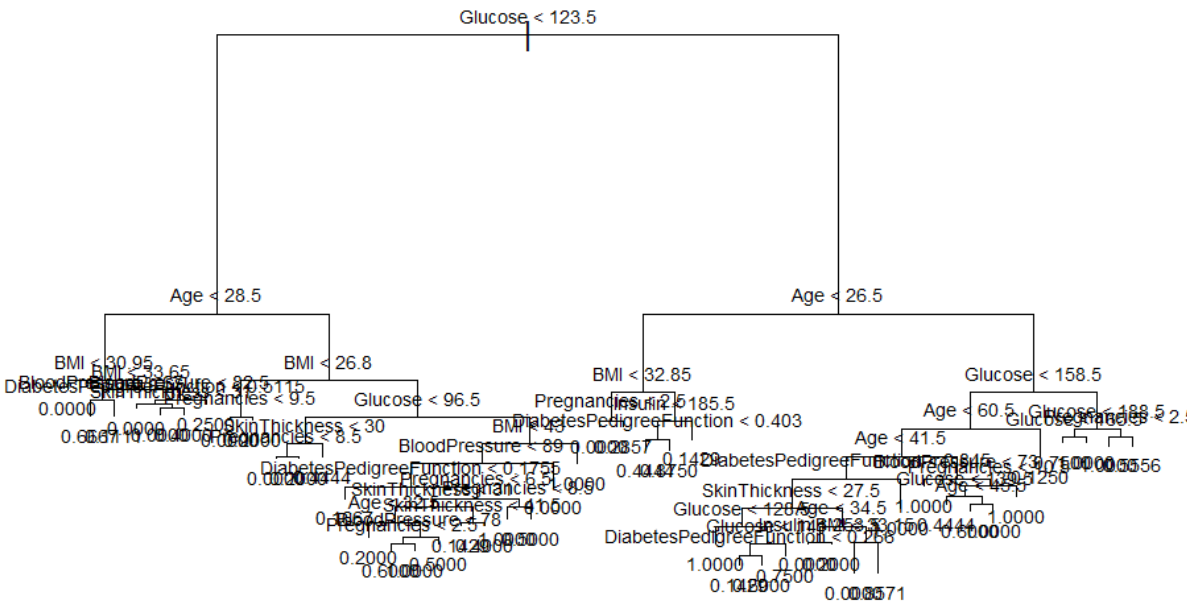
ANALYSIS METHOD 3 -- CLASSIFICATION TREE(3/3)

```{r}
best.mindev <- -1
error.rate <- -1
best.error.rate <- 99999999
for (i in seq(from=0.0005, to=0.05, by=0.0005)) {
  da.tree <- tree(Outcome ~ ., data=da[training,], mindev=i)
  da.tree.proportions <- predict(da.tree,da[-training,])
  da.tree.classifications <- round(da.tree.proportions,0)
  error.rate <- 1- (sum(da.tree.classifications == da.test.results) / nrow(da[-training,]))
  if (error.rate < best.error.rate) {
    best.mindev <- i
    best.error.rate <- error.rate
  }
}
print(paste("The optimal value of mindev is",best.mindev,"with an overall error rate of",best.error.rate))
```

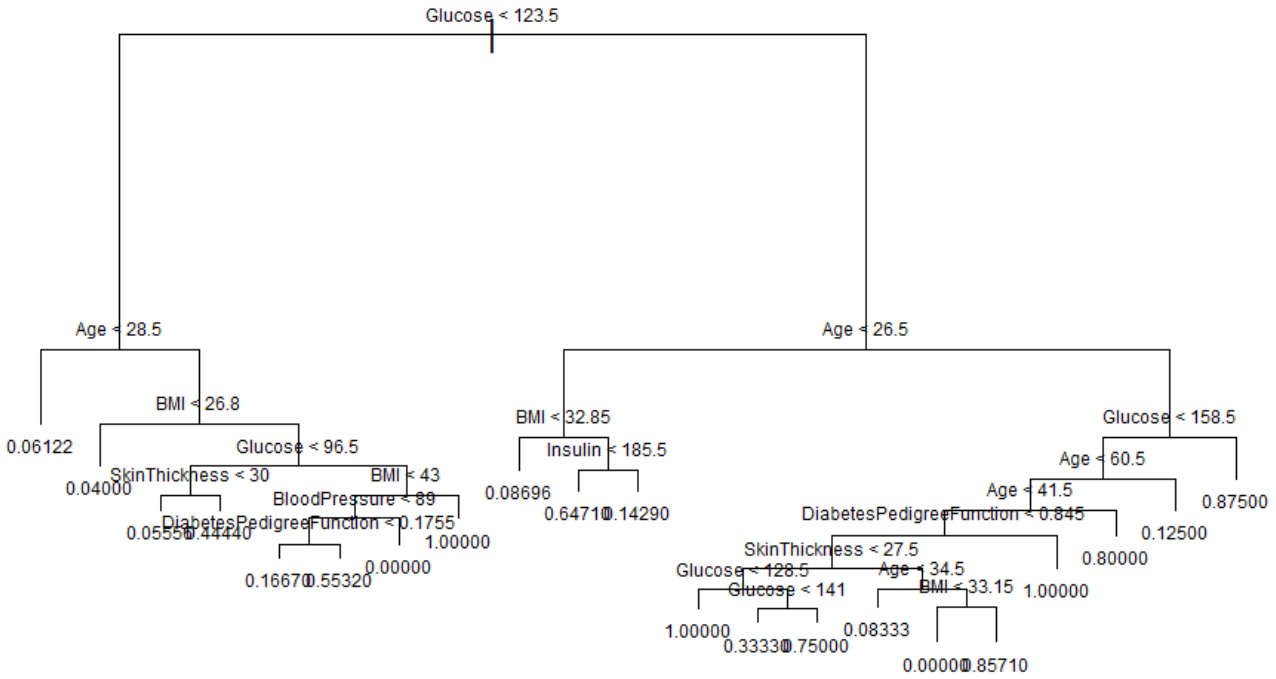
The optimal value of mindev is 0.0075 with an overall error rate of 0.282467532467532

The tree correctly classified 71.75% of the observation in test data

CLASSIFICATION TREE COMPARE



Original tree plot



The best mindev tree plot

PREDICTION RESULT

	KNN	Logistic Regression	Classification Tree
Predict Number	0.23, 0.51, 0.20, 0.23, 0.34, 0.09	0.13, 0.54, 0.31, 0.20, 0.26, 0.08	0.00, 0.17, 0.00, 0.00, 1.00, 0.00
Accuracy rate	0.7045455	0.7824675	0.6980519

This is a table that organizes the accuracy and prediction values of the above three methods

IMPLICATION FOR DECISION MAKER

- According to the result of accuracy rate, we can know the highest accuracy rate is logistic regression.

- We think if decision maker wants to know more about the dataset, they could try logistic regression to predict numbers.

CHALLENGES WITH ANALYZING AND MODELING

A limited data set

- The data only has 798 samples

Do not accurately represent the population

- Only select females over 21 years old of Pima Indian heritage but cannot represent for all population

Outliers that may influence outcome

- Wrong conclusions

THANK YOU