Machine Learning Project:

Unsupervised Learning, Classification and Regression.

Student ID: 210501801

Date: 03-04-2024

Dataset: Pima Indian Diabetes.csv

Link to dataset: Pima Indians Diabetes Database (kaggle.com)

Number of pages: 10

Contents

1.Data description and variables

The Pima Indian dataset is a binary classification dataset that contains data collected from a study conducted on a group of Pima Indian women from the ages 21 to 81 near Phoenix, Arizona, USA, during the 1980s. This research was conducted as the non-insulin-dependent diabetes mellitus was commonly found in Pima Indians. The Pima Indians provide valuable insights valuable insights into the effects of genetic, environment, and lifestyle factors in the development of type 2 diabetes, with implications for understanding and managing the disease in diverse populations as well.

The dataset contains 768 observations with 8 input variables and 1 output variable. The list below represents the various attributes of the Pima Indian dataset.

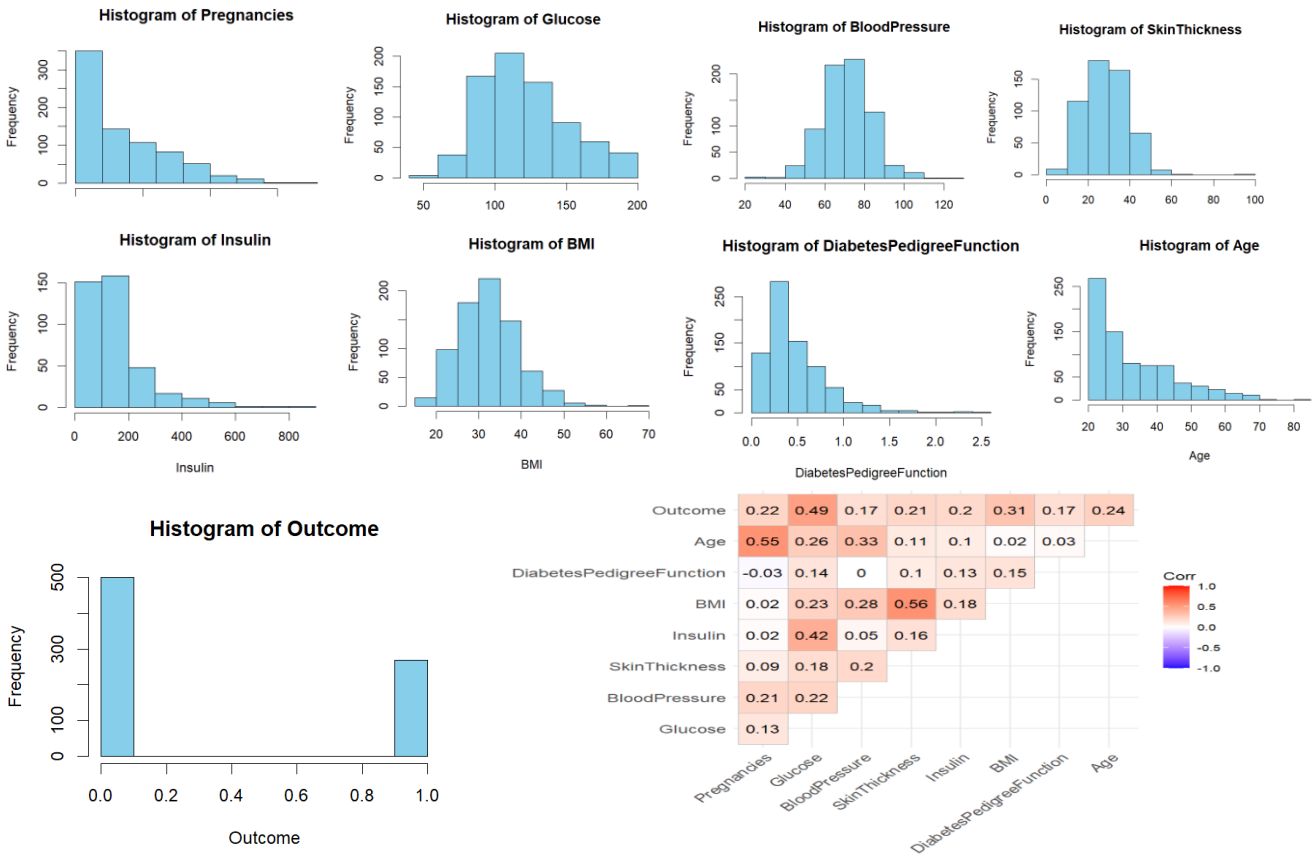
1. **Pregnancies**: Number of times patients were pregnant.
2. **Glucose**: Plasma glucose concentration.
3. **Blood Pressure**: Diastolic blood pressure (mm Hg).
4. **Skin Thickness**: Triceps skin fold thickness (mm).
5. **Insulin**: 2-Hour serum insulin (mu U/ml).
6. **BMI**: Body mass index (weight in kg/ (height in m) ^2).
7. **Diabetes Pedigree Function**: Diabetes pedigree function (likelihood of diabetes based on family history).
8. **Age**: Age of patient in years.
9. **Outcome**: Class variable (0 or 1) indicating whether the individual developed diabetes (1: Yes, 0: No).

2. Exploratory Data Analysis

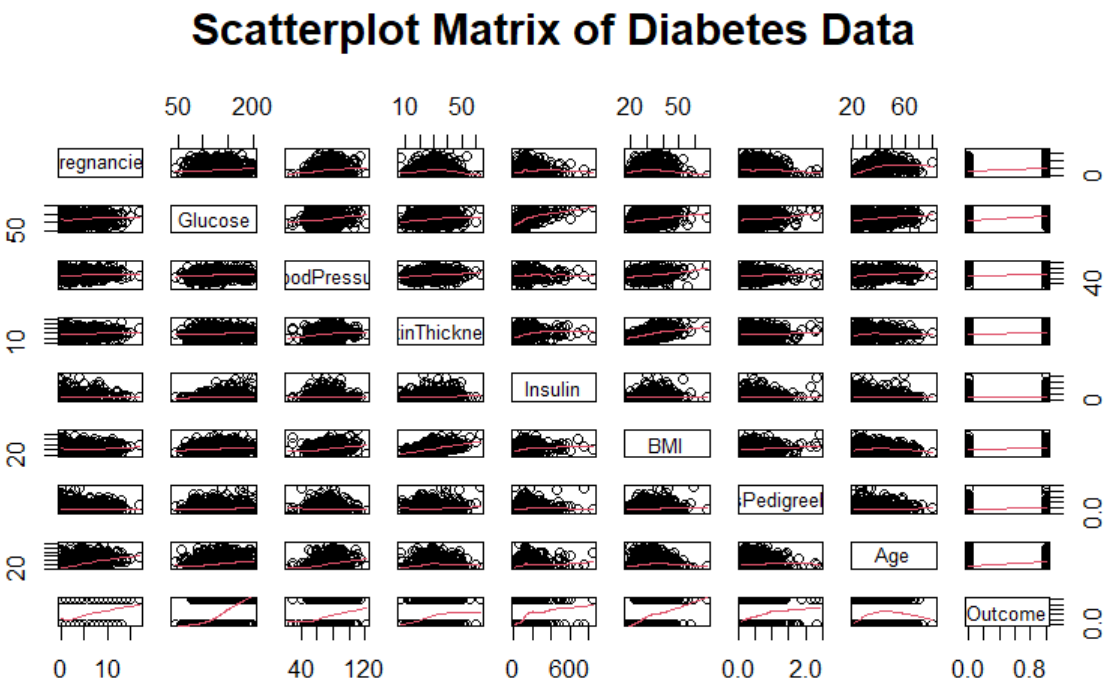
```
> summary(diabetes_data)
  Pregnancies        Glucose      BloodPressure    SkinThickness      Insulin          BMI        
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0   Min.   : 0.00  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.:27.30  
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5   Median :32.00  
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8   Mean   :31.99  
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2   3rd Qu.:36.60  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0   Max.   :67.10  
 DiabetesPedigreeFunction      Age             Outcome     
 Min.   :0.0780           Min.   :21.00   Min.   :0.000  
 1st Qu.:0.2437           1st Qu.:24.00   1st Qu.:0.000  
 Median :0.3725           Median :29.00   Median :0.000  
 Mean   :0.4719           Mean   :33.24   Mean   :0.349  
 3rd Qu.:0.6262           3rd Qu.:41.00   3rd Qu.:1.000  
 Max.   :2.4200           Max.   :81.00   Max.   :1.000  

> colSums(is.na(diabetes_data_copy))
            Pregnancies                 Glucose           BloodPressure           SkinThickness 
                      0                       5                      35                     227 
                Insulin                     BMI DiabetesPedigreeFunction                     Age 
                    374                      11                       0                       0 
                Outcome 
                      0 
> # With a total of 768 values, NA values of each x variable is shown below.
```

We can see from the summary of the data that the minimum values for glucose, blood pressure, skin thickness, BMI and Insulin are 0. These values do not make sense. Suggesting that there are missing values. Upon checking for the number of missing values, it was found that almost 50% of insulin values were missing while 30% of skin thickness values were missing. As the data set only consist of 768 observations, we cannot afford to drop these NA values as they would affect our results drastically. To solve this issue, we then approximated the NA values by using the respective distributions of each output variable.



The histograms above show the distribution of the Pima Indian dataset after it has been cleaned. Based on the histogram of outcomes, we can see that one-third of Pima Indians have diabetes while two-third don't. This is a relatively high proportion. This is good as it can help prevent model bias towards the larger class as it ensures that the model learns from a representative sample of both classes, leading to more accurate predictions.



Using the scatterplot matrix, we visualize the correlations between the various inputs and output variables. This helps us to understand our data better.

3. Unsupervised Learning

3.1 K-Means Clustering

3.1.1 Methodology

We cluster the scaled variables glucose and blood pressure into 2 group. We do this to reveal an underlying pattern among glucose and blood pressure. The Welch 2 sample t-test will be used to evaluate the different in glucose mean between the 2 clusters.

3.1.2 Substantive issue

RQ: To examine of blood pressure is a significant differentiator for Glucose.

```
> summary(cluster1$diabetes_data_new.BloodPressure)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   24.0    60.0    68.0    66.8    73.6    96.0
> summary(cluster2$diabetes_data_new.BloodPressure)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  50.00   72.00   78.00   78.98   86.00  122.00
> ## Cluster 2 has higher BloodPressure than cluster 1.
```


```
        Welch Two Sample t-test

data:  cluster1$diabetes_data_new.Glucose and cluster2$diabetes_data_new.Glucose
t = -15.205, df = 653.25, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -33.61063 -25.92220
sample estimates:
mean of x mean of y
 107.8890  137.6554

>
> # Using Welch 2 sample t-test, the mean glucose level of cluster 2 is higher than cluster 1.
> # Hence suggesting that women with higher glucose levels tend to have higher blood pressure as well.
```

As seen from the cluster summary, cluster 2 had a higher blood pressure than cluster 1. We used a Welch 2 sample t-test to calculate the glucose mean of cluster 1 and cluster 2. From the results we see that cluster 2 had a higher glucose level at 137.6554 than cluster 1 at 107.8890. Using K-mean clustering , it is evident that patients with a higher plasma glucose level tend to have higher blood pressure as well. This pattern can be seen from the K-means plot above.

3.1.3 Relevant exisiting literature

The above results prove to be true as elevated levels of glucose in your bloodstream can result in a condition known as atherosclerosis. This occurs when fatty substances accumulate within your blood vessels, causing them to narrow. As the blood vessels become narrower, the pressure within them increases. (The british diabetic association , 2024). As a result, making sense of the k-means clustering results obtained above.

3.2 Principal Component Analysis

3.2.1 Methodology

PCA is a dimension reducing statistical technique that tries to retain as much information as possible. First calculate the mean and variance of each variable. Then we performed PCA on standardized data. This reduces the dimensionality of the diabetes data by converting the original variables into a new set of uncorrelated variables. 2 graphs are then created to show the proportion of variance and cumulative variance explained by PCA. Eventually we will examine the total variance explained by PC1 and PC2.

3.2.2 Substantive issue

RQ: To analyse the loadings and identify significant variables using PC1 and PC2.

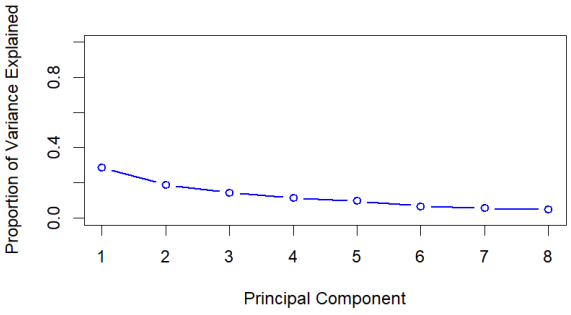
```
                              PC1         PC2         PC3          PC4         PC5          PC6
Pregnancies             -0.3025481  0.55306488  0.04671759 -0.1597409935 -0.40797344 -0.046464038
Glucose                 -0.4179820 -0.06546044 -0.45670165  0.2306613212  0.19268454  0.702948620
BloodPressure           -0.3794600  0.16812279  0.30027539  0.1136653024  0.75414223 -0.314467303
SkinThickness           -0.4002063 -0.32037897  0.39640037  0.0004163111 -0.39689210  0.033725600
Insulin                 -0.3088663 -0.22358931 -0.57311587  0.3147713604 -0.18055777 -0.627402264
BMI                     -0.4079971 -0.40277507  0.36445396  0.0110519819 -0.06610897  0.069267793
DiabetesPedigreeFunction -0.1566487 -0.26391155 -0.27672935 -0.8881458537  0.16674088 -0.072027322
Age                     -0.3784202  0.52636202 -0.06383483 -0.1426603749 -0.07350144  0.007593824
                               PC7         PC8
Pregnancies             0.54942607 -0.32313143
Glucose                 0.05037221 -0.15944744
BloodPressure           0.02290027 -0.23772903
SkinThickness          -0.48021310 -0.43679905
Insulin                 0.02021017  0.02057815
BMI                     0.44720842  0.57375777
DiabetesPedigreeFunction 0.01221441 -0.08538178
Age                    -0.51372325  0.53473601
```

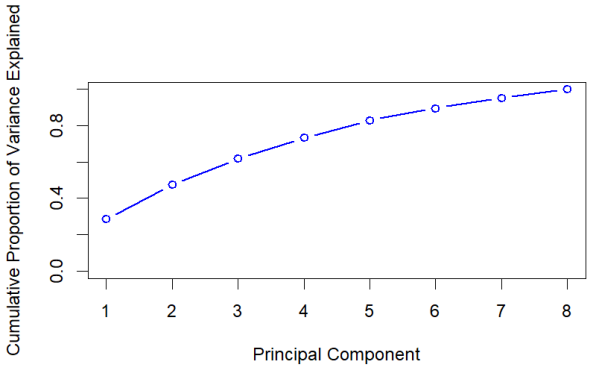
PC1 has relatively high absolute loading values for Glucose, Blood pressure, Skin thickness, BMI and age. The negative loadings on Glucose, Blood pressure, Skin thickness, Insulin, BMI, and Age indicate an inverse relationship with PC1, suggesting that higher values of these variables contribute to lower values of PC1.

PC2 shows a high loading for Pregnancies and Age, indicating that it captures variability related to these variables. It seems to represent demographic factors such as age and the number of pregnancies. Negative loading for Pregnancies indicates an inverse relationship with PC2, suggesting that a higher number of pregnancies contributes to lower values of PC2 while positive loading for Age indicates a positive relationship with PC2, suggesting that higher age values contribute to higher values of PC2.

Furthermore, we can see that diabetes pedigree function is relatively not as important as the other variables. As its loading scores are low in both PC1 and PC2.

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8
Standard deviation     1.5118 1.2275 1.0741 0.9578 0.87694 0.73276 0.66785 0.62010
Proportion of Variance 0.2857 0.1884 0.1442 0.1147 0.09613 0.06712 0.05575 0.04807
Cumulative Proportion  0.2857 0.4741 0.6183 0.7329 0.82906 0.89618 0.95193 1.00000
> #First two principal components capture 47.4% of v
```





We can see that the first 2 principal components capture 47.4% of variance. It is able to retain a significant proportion of information found in the original Pima Indian dataset but not a majority of it. The remaining 52.6% is spread across the other 6 PC loadings.

4. Classification

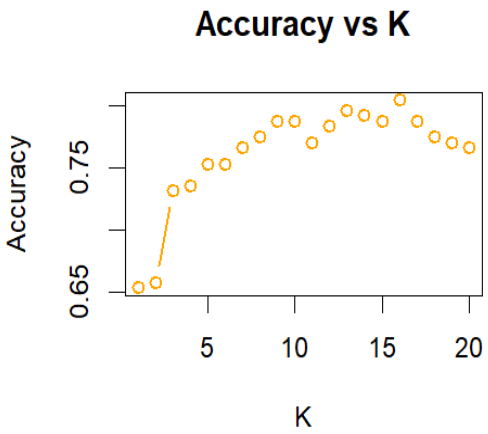
4.1 KNN

4.1.1 Methodology

To obtain the best k value, a loop is iterated over different values of K. The model is trained using training and testing data. The k value with the highest accuracy is selected. 3 different metrics will be used to evaluate the model along with plotting an ROC curve.

4.1.2 Substantive issue

RQ: To find the optimal value of K that provides us with the highest accuracy. Followed by evaluating the model's performance using accuracy, confusion matrix, precision, recall, and F1 score, and generate a ROC curve for further evaluation.

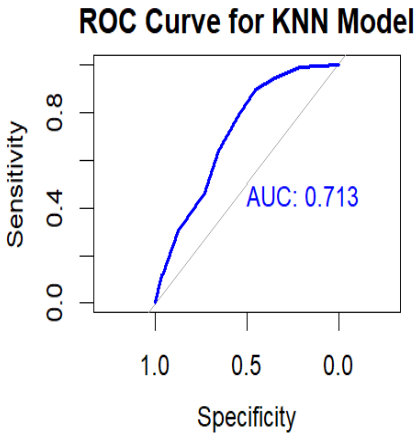



```
Confusion Matrix and Statistics

          knn_model
y_test    0   1
     0  134  11
     1   35  51

               Accuracy : 0.8009
                 95% CI : (0.7435, 0.8504)
    No Information Rate : 0.7316
    P-Value [Acc > NIR] : 0.009160


> cat("Precision Score:", precision, "\n")
Precision Score: 0.8225806
> cat("Recall Score:", recall, "\n")
Recall Score: 0.5930233
> cat("F1 Score:", f1, "\n")
F1 Score: 0.6891892
```



When k = 16, the model generates the highest accuracy of 80%. A precision score of 82.26% suggests that the KNN model has a relatively high accuracy in identifying individuals who truly have diabetes among those it predicts as positive. However, a 59.30% recall score suggest that the model misses some positive instances leading to false negatives. A f1 score of 68.91% suggest that there is a reasonable balance between precision and recall. Overall, indicating a relatively good performance in predicting diabetes. With an AUC of 0.713, this KNN model has a moderate discriminatory power in distinguishing between individuals with diabetes and those without it on the Pima Indian dataset. While not extremely high, it still demonstrates some ability to correctly classify those classes.

4.2 Logistic Regression

4.2.1 Methodology

First the logistic regression is fitted using glucose only to understand its influence. The p-values, odd ratios and confidence intervals are calculated to understand the impact of glucose on the probability of having diabetes. A logistic regression curve will be plotted to visualize the relationship between glucose and probability of having diabetes.

Moving to the second half of the codes, a logistic regression is fitted using all predictor variables. This allows us to asses the significance of each predicator. The accuracy of this model will be evaluated as well.

4.2.2 Substantive issue

RQ: To examine if the increase in blood plasma glucose concentration affects the probability of getting diabetes.

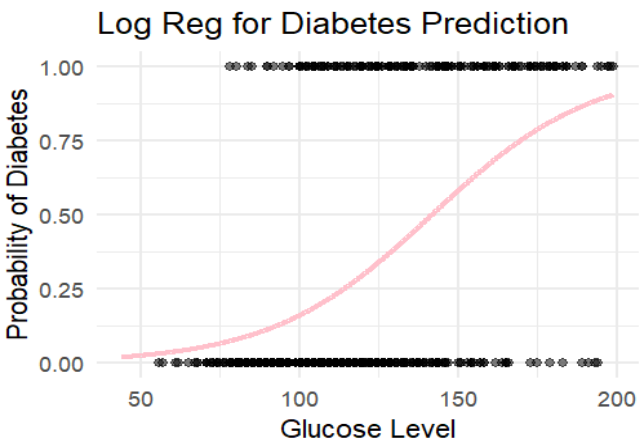
Logistic Regression Analysis

```
Call:
glm(formula = Outcome ~ Glucose, family = binomial, data = train_data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.603767   0.518680  -10.80   <2e-16 ***
Glucose      0.039581   0.004014    9.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
> OR <- exp(coef(logistic_model))
> OR
(Intercept)     Glucose
0.003683961 1.040374817
```



The model generated follows, **$\hat{Y}$ = -5.603767 + 0.039581 (Glucose).** Glucose is a statistically significant input variable. By looking at the odds ratio, we can see that by increasing glucose plasma concentration by 1 unit, the probability of getting diabetes increases by 1.04037 units.

4.2.3 Substantive issue

RQ: Using logistic regression, we then examined the statistically significant of the 8 input variables along with its accuracy.

```
Call:
glm(formula = Outcome ~ ., family = binomial, data = train_data)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -8.974831   0.976564  -9.190  < 2e-16 ***
Pregnancies                0.161438   0.038725   4.169 3.06e-05 ***
Glucose                    0.038628   0.004855   7.956 1.77e-15 ***
BloodPressure             -0.011567   0.010831  -1.068   0.2855
SkinThickness              0.001041   0.016031   0.065   0.9482
Insulin                   -0.001554   0.001411  -1.101   0.2707
BMI                        0.096786   0.021808   4.438 9.08e-06 ***
DiabetesPedigreeFunction   0.873909   0.363221   2.406   0.0161 *
Age                        0.006669   0.011266   0.592   0.5539
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 686.36  on 536  degrees of freedom
Residual deviance: 495.27  on 528  degrees of freedom
AIC: 513.27
```

```
  binary_predictions
     0   1
  0 131  13
  1  37  49
>
> # Calculate accuracy
> log_reg_accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
> log_reg_accuracy
[1] 0.7826087
```

From the results above, we can see that pregnancies, Glucose, BMI and DiabetesPedigreeFunction are statistically significant inputs. Age and SkinThickness have relatively high p-values, suggesting that there aren't statistically significant. This model generates an accuracy of 78.26%.

4.3 Naïve Bayes

4.3.1 Methodology

Naïve bayes assumes the predictor variables are independent of each other. This model is trained using the training data where the y variable is outcome. The accuracy of the model will be calculated. 3 different metrics will be used to evaluate the model along with plotting an ROC curve.

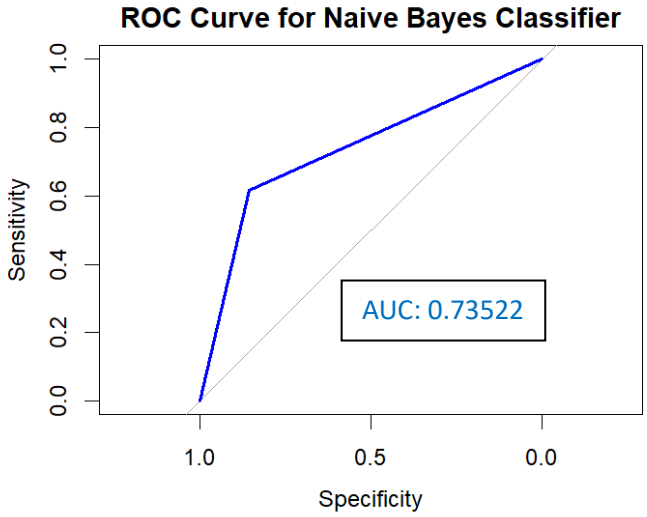
4.3.2 Substantive issue

RQ: To evaluate the model's performance using accuracy, precision, recall score and f1 score together with generating an ROC curve.

```
[1] "NB_Precision: 0.788461538461538"
> print(paste("NB_Recall:", NB_recall))
[1] "NB_Recall: 0.854166666666667"
> print(paste("NB_f1-Score:", NB_f1_score))
[1] "NB_f1-Score: 0.82"
> # Evaluate the accuracy of the model
> NB_accuracy <- mean(predictions == test_data$Outcome)
> print(paste("Accuracy:", NB_accuracy))
[1] "Accuracy: 0.765217391304348"
> # The accuracy is 76.52 %

          Reference
Prediction   0   1
         0 123  33
         1  21  53

               Accuracy : 0.7652
                 95% CI : (0.705, 0.8184)
    No Information Rate : 0.6261
    P-Value [Acc > NIR] : 4.652e-06
```



The naïve bayes model has an accuracy of 76.52%. A value of 0.788 suggest that out of all the instances predicted as positive by the Naive Bayes model, approximately 78.8% were correctly classified as positive. A recall value of 0.854 signifies that the Naive Bayes model correctly identified approximately 85.4% of all actual positive instances in the dataset. A f1-score of 0.82 suggest that the Naive Bayes model achieves a good balance between precision and recall. An AUC score of 0.73522 suggests that the naïve bayes model has a moderate ability to tell apart between the positive and negative classes

4.4 Decision tree

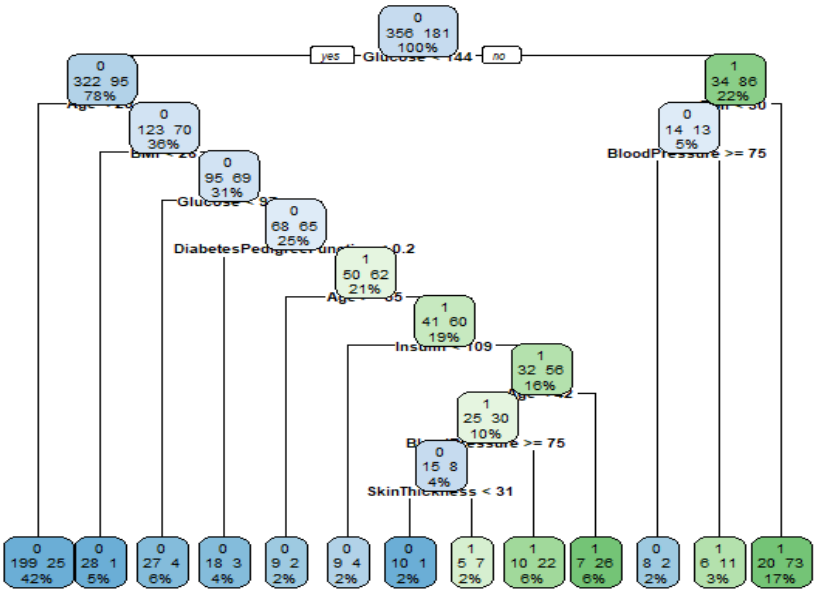
4.4.1 Methodology

A decision tree model is trained using the training data where the target variable is outcome. Predictions are generated on the test data and the accuracy of this model will be calculated. Using rpart.plot(), we will analyse the structure of the tree.

4.4.2 Substantive issue

RQ: To evaluate the accuracy of the decision tree model to identify patients with and without diabetes.

```
> # Print accuracy
> print(paste("Accuracy:", DT_accuracy))
[1] "Accuracy: 0.752173913043478"
```



The decision tree model correctly predicted the outcome for approximately 75.22% of the cases in the test dataset. It is important to note that variables that appear nearer to the root of the tree are considered to be more influential. As they have a larger impact on the final classification results. Based on the tree, glucose seems to be the most influential variable as it appears at the root of the tree. While Skin thickness isn't as important as it appears only once further down the tree. There are a total of 13 terminal nodes. To properly understand how the decision tree works, we'll take a look at 2 examples. Based on the graph, patients with glucose level of more than 144 along with a BMI of more than 30 were considered diabetic. In scenario 2, patients with blood pressure more than 75 and glucose more than 144 were considered diabetic although their BMI was less than 30. Patients with a blood pressure of less than 75 were classified as non-diabetic.

4.5 SVM

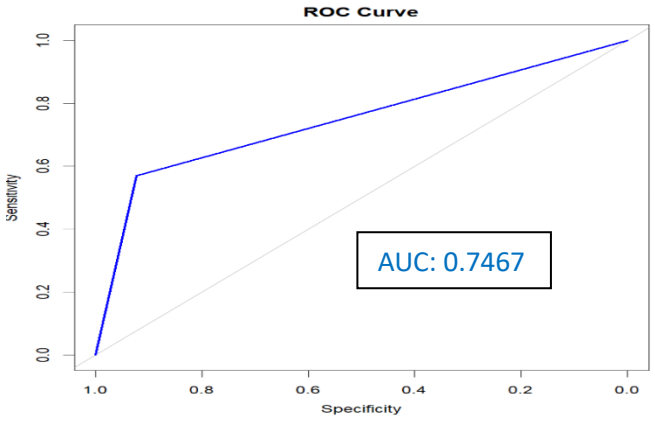
4.5.1 Methodology

A svm is trained using the target variable outcome. The kernal used will be Radial. The accuracy, f1 scores, recall scores and precision scores will be evaluated along with generating the ROC curve to assess the model's classification performance.

4.5.2 Substantive issue

RQ: To evaluate the model's performance using accuracy, precision, recall score and f1 score together with generating an ROC curve.

```
SVM Model Evaluation:
> cat("F1 Score:", svm_f1, "\n")
F1 Score: 0.6712329
> cat("Recall Score:", svm_recall, "\n")
Recall Score: 0.5697674
> cat("Precision Score:", svm_precision, "\n")
Precision Score: 0.8166667
> svm_accuracy
[1] 0.7913043
```



The SVM model generates an accuracy of 79.13%. We've used an SVM model in this particular dataset as SVM models are particularly good for datasets of smaller sizes. A precision score of 79.13 % suggests that the SVM model has a pretty high accuracy in identifying individuals who have diabetes among those it predicted as positive. A precision score of approximately 0.81667 indicates

that when the SVM model predicts a positive case, it is correct about 82% of the time. In summary, the SVM model has a decent F1 score, indicating a good overall performance in terms of both precision and recall. However, the recall score suggests that there is room for improvement in identifying positive cases, while the precision score indicates that the model's positive predictions are highly accurate. An AUC score of 0.7467 suggests that the SVM model has a moderate ability to tell apart between the positive and negative classes.

5. Regression

5.1 Linear Regression

5.1.1 Methodology

A model is built to examine the effects on glucose on blood pressure. Using p-value, will we examine whether glucose is a significant variable. A scatterplot is created to visualize the relationship between the variables.

On the second half of the codes, we built another regression model using all variables with the train data. We do this to obtain the RMSE which will be evaluated against the OOB of random forest.

5.1.2 Substantive issue

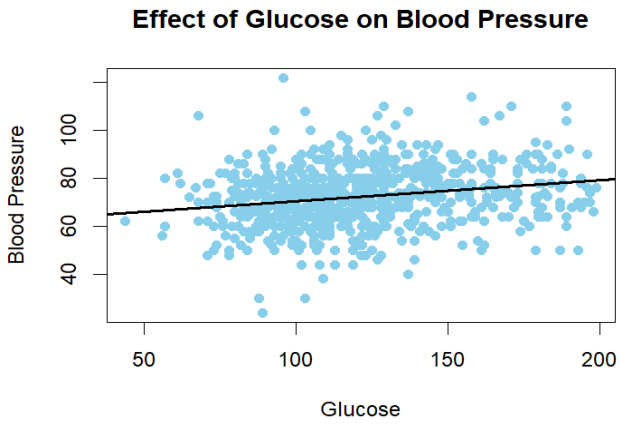
RQ: To examine the effect glucose has on blood pressure.

```
lm(formula = BloodPressure ~ Glucose, data = diabetes_data_new)

Residuals:
    Min      1Q  Median      3Q     Max
-45.549  -7.593  -0.093   7.477  51.837

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 61.73970    1.76357  35.008  < 2e-16 ***
Glucose      0.08774    0.01407   6.235 7.48e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.82 on 765 degrees of freedom
Multiple R-squared:  0.04835,   Adjusted R-squared:  0.04711
F-statistic: 38.87 on 1 and 765 DF,  p-value: 7.483e-10
```



The model generated is as follows, **$\hat{Y}$ = 61.73970 + 0.08774 (Glucose).** As glucose level increases, blood pressure increases as well. There is a positive correlation between blood pressure and glucose levels. From the results above, we see that glucose is a significant predictor as the p-value is less than $\alpha = 0.05$. Hence, we reject the null hypothesis and conclude that glucose has a significant effect on blood pressure.

5.2 Random Forest

5.2.1 Methodology

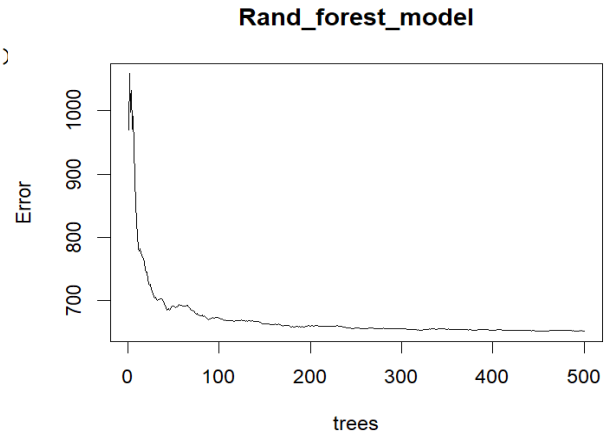
A random forest model is built to predict glucose based on all other variables. The importance of each variable is calculated to reflect its contributions. A random forest graph will be generated to show error rate as a function of the number of trees, allowing us to assess the stability of the model. The Out-of-bag RMSE is calculated. It is then compared against the RMSE of the linear regression model above.

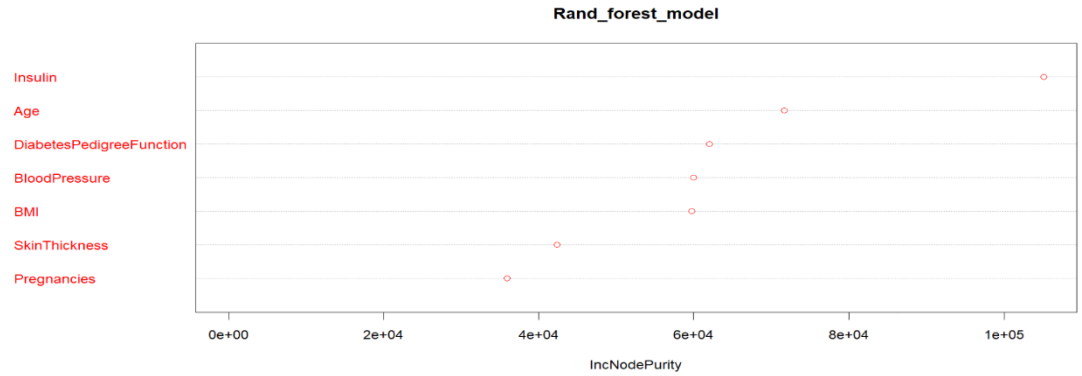
5.2.2 Substantive issue

RQ: To calculate the importance of 7 other input variables where the random forest outcome is glucose.

```
Call:
 randomForest(formula = Glucose ~ ., data = train_data[, -9]) 
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 652.1047
                    % Var explained: 26.91
>
> importance <- importance(Rand_forest_model)
> print(importance)
                          IncNodePurity
Pregnancies                     36511.45
BloodPressure                   59122.80
SkinThickness                   41552.28
Insulin                        105752.58
BMI                             60422.09
DiabetesPedigreeFunction        62146.25
Age                             71222.83
```





From the Random Forest model, we can see that when using glucose as the outcome variable, insulin is deemed the most important and influential variable. Number of pregnancies were deemed the least important. Furthermore, the OOB error rate seems to stabilize before 500 trees as seen from the graph above. This suggest that we do not have to increase the number of trees as it will most likely not increase our accuracy. In addition, a value of 25.53488 for the OOB error means that, on average, the random forest model is expected to have an error of approximately 25.53488 units when making predictions on unseen diabetes data.

6. RMSE comparison

```
> print(paste("RMSE for Random Forest:", OOB_RMSE))
[1] "RMSE for Random Forest: 25.536340739398"
> print(paste("RMSE for Linear Regression:",LR_RMSE ))
[1] "RMSE for Linear Regression: 25.8607444444431"
```

By comparing the RMSE of Random Forest and Linear regression, we can see that the RMSE for Random Forest is slightly lesser at 25.536 compared to Linear regression at 25.8607. Suggesting that Random Forest is a better predictive model.

7. Reference

BridgetChapple (2024) *Diabetes and blood pressure*, *Diabetes UK*. Available at: https://www.diabetes.org.uk/guide-to-diabetes/managing-your-diabetes/blood-pressure (Accessed: 03 April 2024).