

Prediction of Diabetes

Nishant Rodrigues

April 22nd 2024

Introduction

The diabetes is threatening a lot of people nowadays, without having a perfect cure for it. There are actually two types of diabetes, namely Type 1 and Type 2. The type 2 diabetes is commonly called as diabetes mellitus. It can be defined as a chronic condition that affects the way the body processes blood sugar (glucose). We consider the mellitus here. After deep researches we found that, that some parameters are directly responsible for the mellitus to occur. By using the data of the people with and without diabetes, a dataset has been built. We use that dataset to classify the people who are in the risk of getting diabetes.

Objectives

- Investigate factors related to the risk of diabetes using a dataset containing observations of people with and without diabetes.
- Build a logistic regression model to classify individuals at risk of diabetes.
- Evaluate the model's performance using visualizations, including ROC curves.
- Identify areas for improvement, such as feature scaling and data cleaning, to enhance the model's predictive power.

Dataset Description

The dataset, stored in a CSV format as **diabetes.csv**, contains observations related to diabetes. Each row represents an individual and includes the following variables:

- **Pregnancies:** Number of times pregnant
- **Glucose:** Plasma glucose concentration after 2 hours in an oral glucose tolerance test
- **BloodPressure:** Diastolic blood pressure (mm Hg)
- **SkinThickness:** Triceps skin fold thickness (mm)
- **Insulin:** 2-Hour serum insulin (mu U/ml)
- **BMI:** Body mass index (weight in kg/(height in m)²)
- **DiabetesPedigreeFunction:** Diabetes pedigree function
- **Age:** Age in years
- **Outcome:** Class variable indicating whether the individual has diabetes (1) or not (0)

Loading the required libraries

```
library(ggplot2)
library(ggvis)
library(corrplot)
library(caTools)
library(ROCR)
```

Data Loading

The observations of the people are stored in a CSV format, named diabetes.csv. The data is loaded in the environment. Let's check how the data is structured.

```
data = read.csv("C:/Users/User/Downloads/diabetes.csv")
head(data)
```

```
##   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI
## 1           6    148             72           35         0 33.6
## 2           1     85             66           29         0 26.6
## 3           8    183             64            0         0 23.3
## 4           1     89             66           23        94 28.1
## 5           0    137             40           35       168 43.1
## 6           5    116             74            0         0 25.6
##   DiabetesPedigreeFunction  Age  Outcome
## 1                   0.627   50         1
```

```
## 2      0.351  31      0
## 3      0.672  32      1
## 4      0.167  21      0
## 5      2.288  33      1
## 6      0.201  30      0
```

```
summary(data)
```

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

```
str(data)
```

```
## 'data.frame':   768 obs. of  9 variables:
##  $ Pregnancies      : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose          : int  148 85 183 89 137 116 78 115 19
##  $ BloodPressure    : int  72 66 64 66 40 74 50 0 70 96 ..
##  $ SkinThickness    : int  35 29 0 23 35 0 32 0 45 0 ...
```

```
## $ Insulin      : int  0 0 0 94 168 0 88 0 543 0 ...
## $ BMI          : num  33.6 26.6 23.3 28.1 43.1 25.6 3
## $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 .
## $ Age          : int  50 31 32 21 33 30 26 29 53 54 .
## $ Outcome      : int  1 0 1 0 1 0 1 0 1 1
```

Correlations

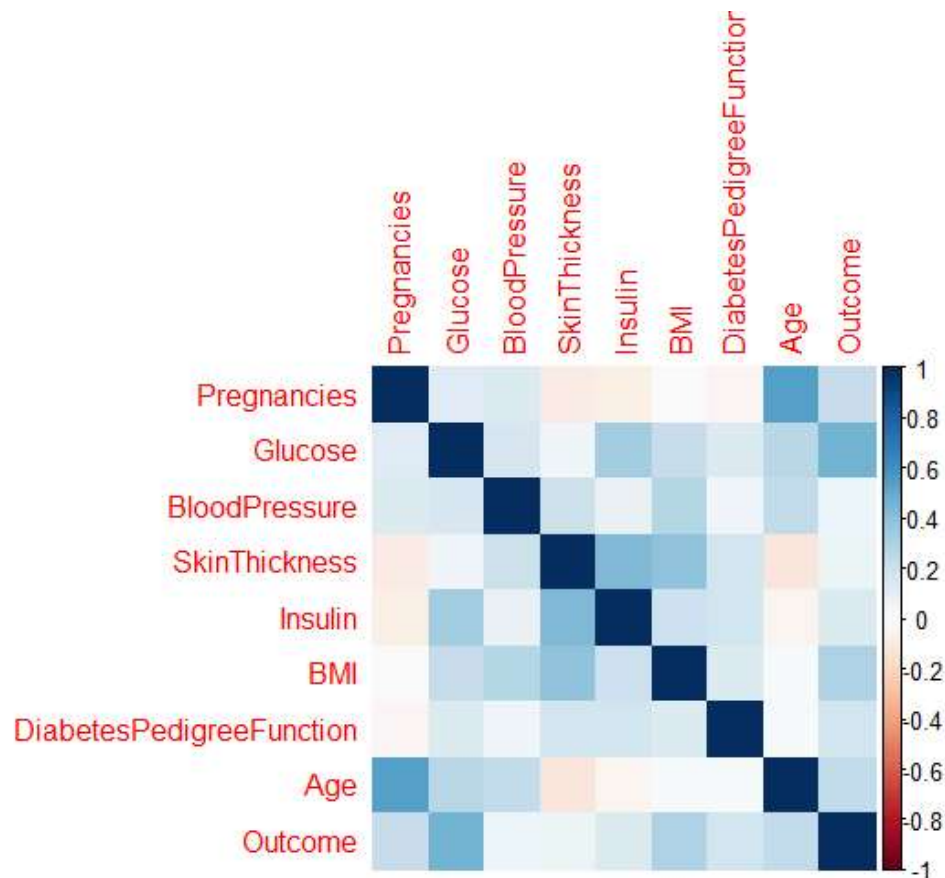
The proportionalities of the attributes of the data can be identified by the correlation coefficient, either numerically or visually. They help to know which attributes are highly dependent on the prediction variable: Outcome.

```
correlations <- cor(data)
correlations
```

```
##              Pregnancies    Glucose BloodPressure SkinThickness
## Pregnancies      1.00000000 0.12945867   0.14128198      -
## Glucose          0.12945867 1.00000000   0.15258959
## BloodPressure    0.14128198 0.15258959   1.00000000
## SkinThickness   -0.08167177 0.05732789   0.20737054
## Insulin         -0.07353461 0.33135711   0.08893338
## BMI             0.01768309 0.22107107   0.28180529
## DiabetesPedigreeFunction -0.03352267 0.13733730   0.04126495
## Age             0.54434123 0.26351432   0.23952795      -
## Outcome         0.22189815 0.46658140   0.06506836
##              Insulin      BMI DiabetesPedigreeFunction
## Pregnancies -0.07353461 0.01768309      -0.0
## Glucose     0.33135711 0.22107107      0.1
## BloodPressure 0.08893338 0.28180529      0.0
## SkinThickness 0.43678257 0.39257320      0.1
## Insulin      1.00000000 0.19785906      0.1
## BMI          0.19785906 1.00000000      0.1
## DiabetesPedigreeFunction 0.18507093 0.14064695      1.0
## Age         -0.04216295 0.03624187      0.0
## Outcome     0.13054795 0.29269466      0.1
##              Age      Outcome
## Pregnancies 0.54434123 0.22189815
## Glucose     0.26351432 0.46658140
## BloodPressure 0.23952795 0.06506836
## SkinThickness -0.11397026 0.07475223
```

```
## Insulin      -0.04216295  0.13054795
## BMI          0.03624187  0.29269466
## DiabetesPedigreeFunction 0.03356131  0.17384407
## Age          1.00000000  0.23835598
## Outcome      0.22825508  1.00000000
```

```
corrplot(correlations, method="color")
```

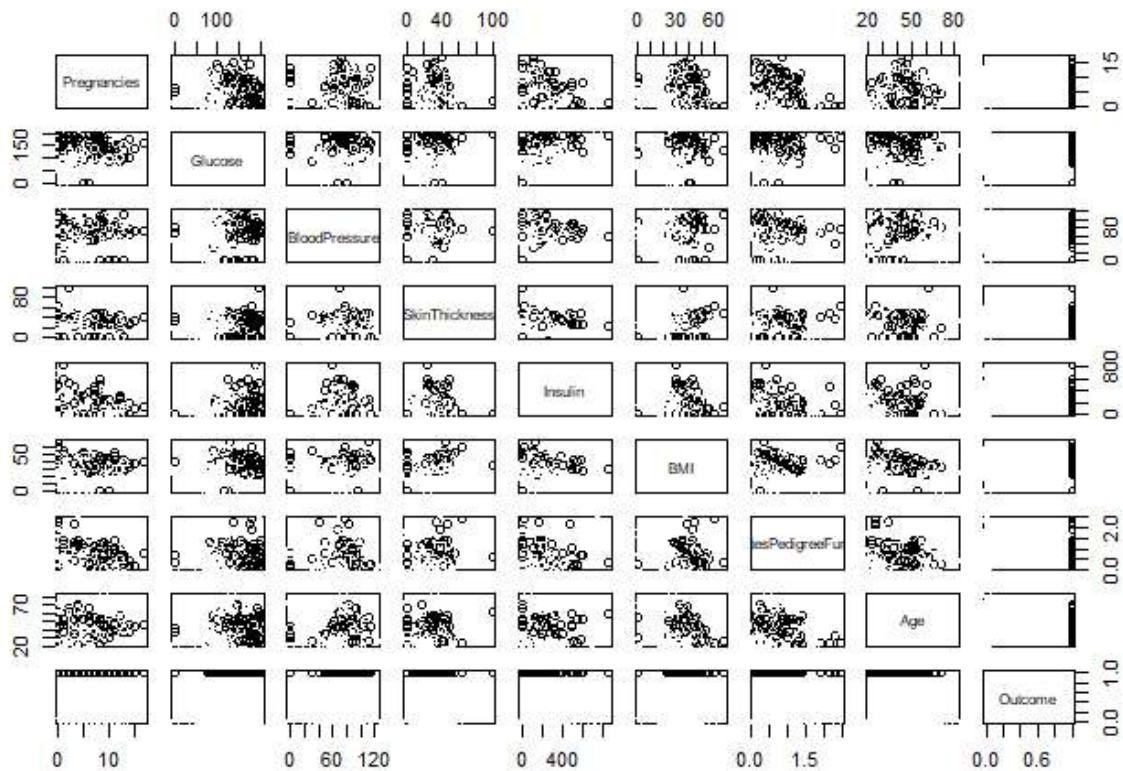


Visualization

Visualizations are used to grasp the structure of data and its relations, like how they vary and their relationships with the other data. They are said to be EDA.

A matrix of scatterplots is produced for this dataset.

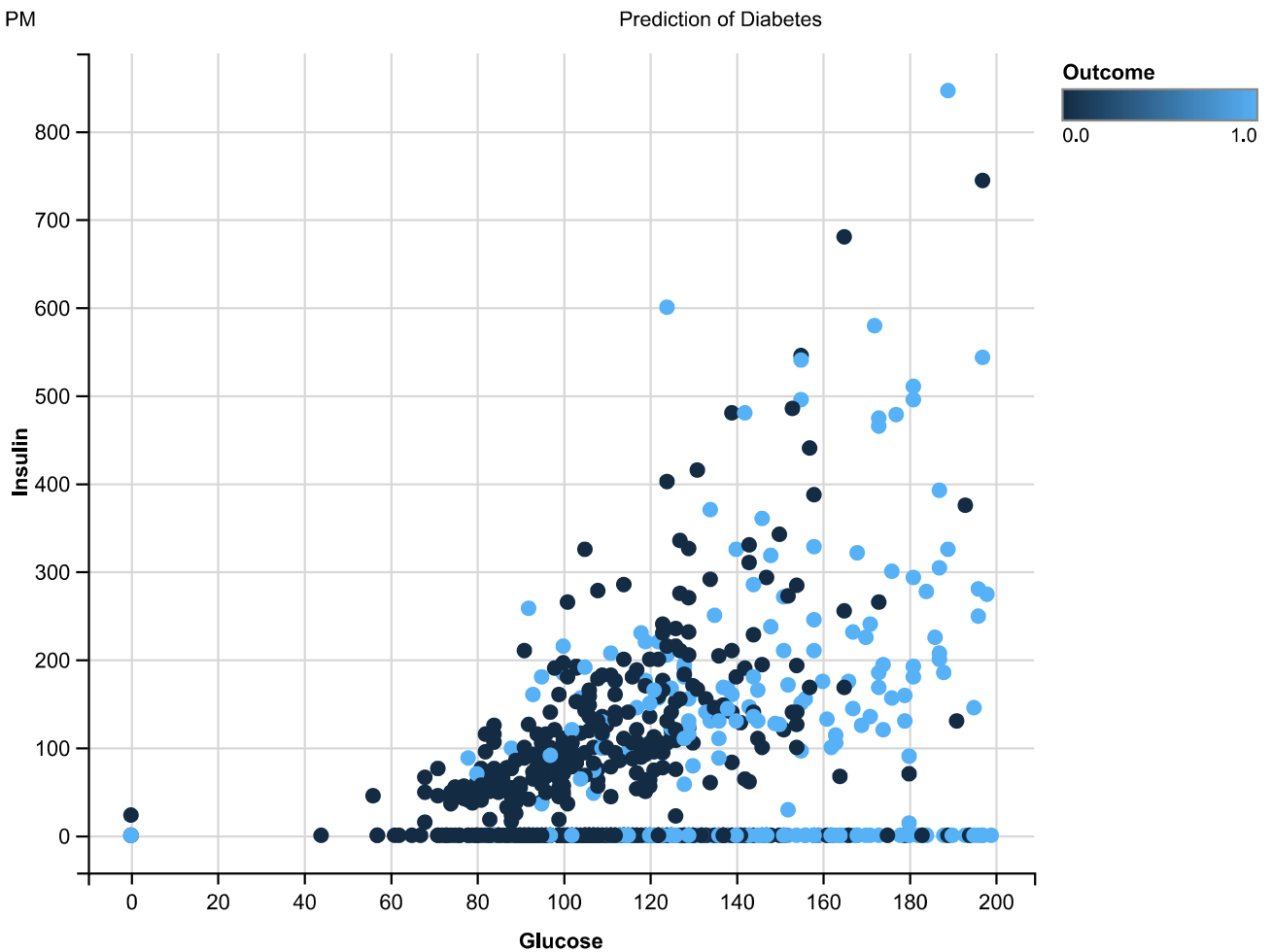
```
pairs(data, col=data$Outcome)
```



Glucose and Insulin

The glucose and the insulin are the major factors of the diabetes...which in turn have direct proportionality in the future during the diabetes. They are the major cause of the occurrence. They are strong correlated on each other.

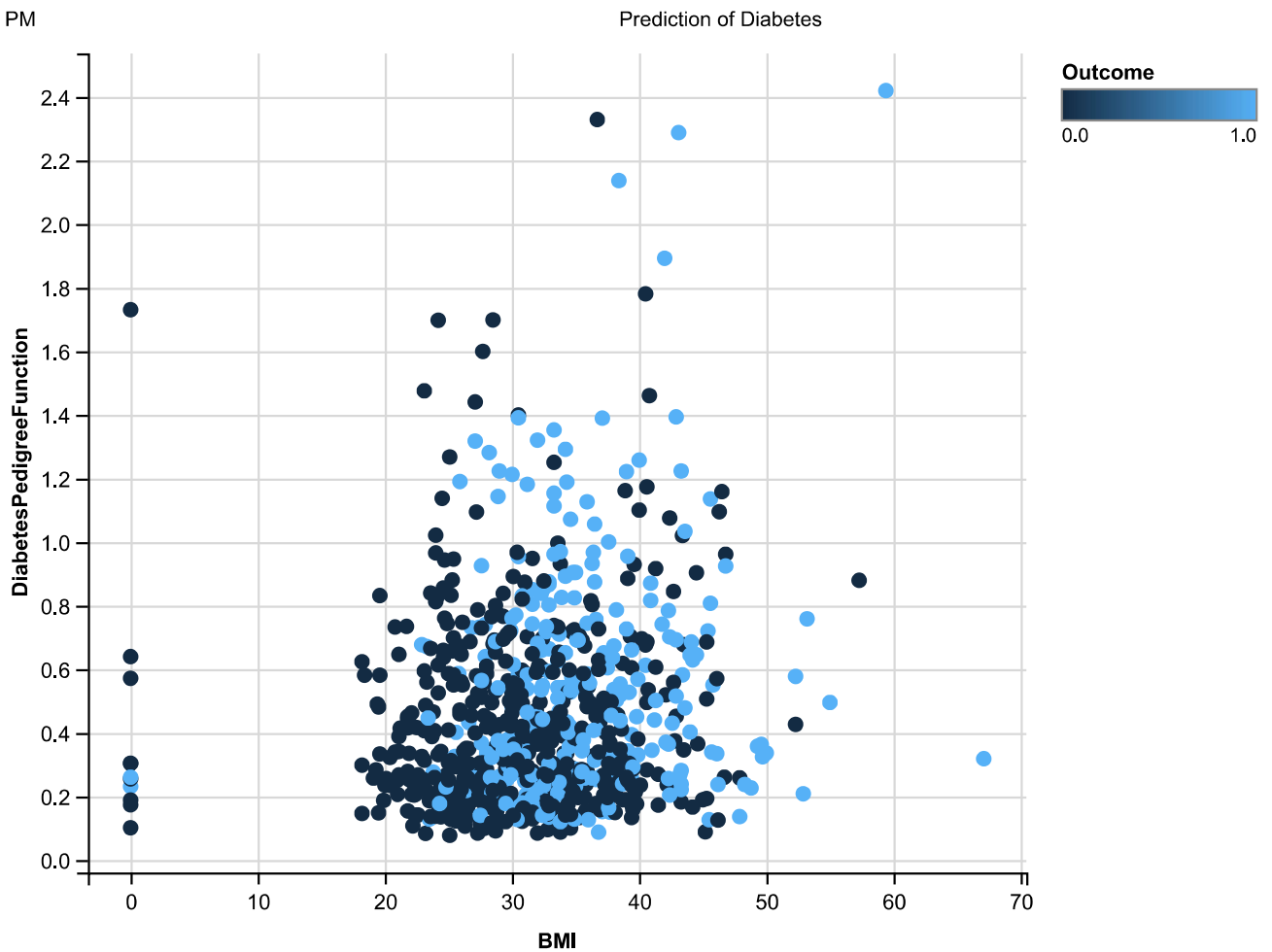
```
data %>% ggvis(~Glucose,~Insulin,fill =~Outcome) %>% layer_points()
```



BMI ad DiabetesPedigreeFunction

The BMI and DiabetesPedigreeFunction is plotted here.

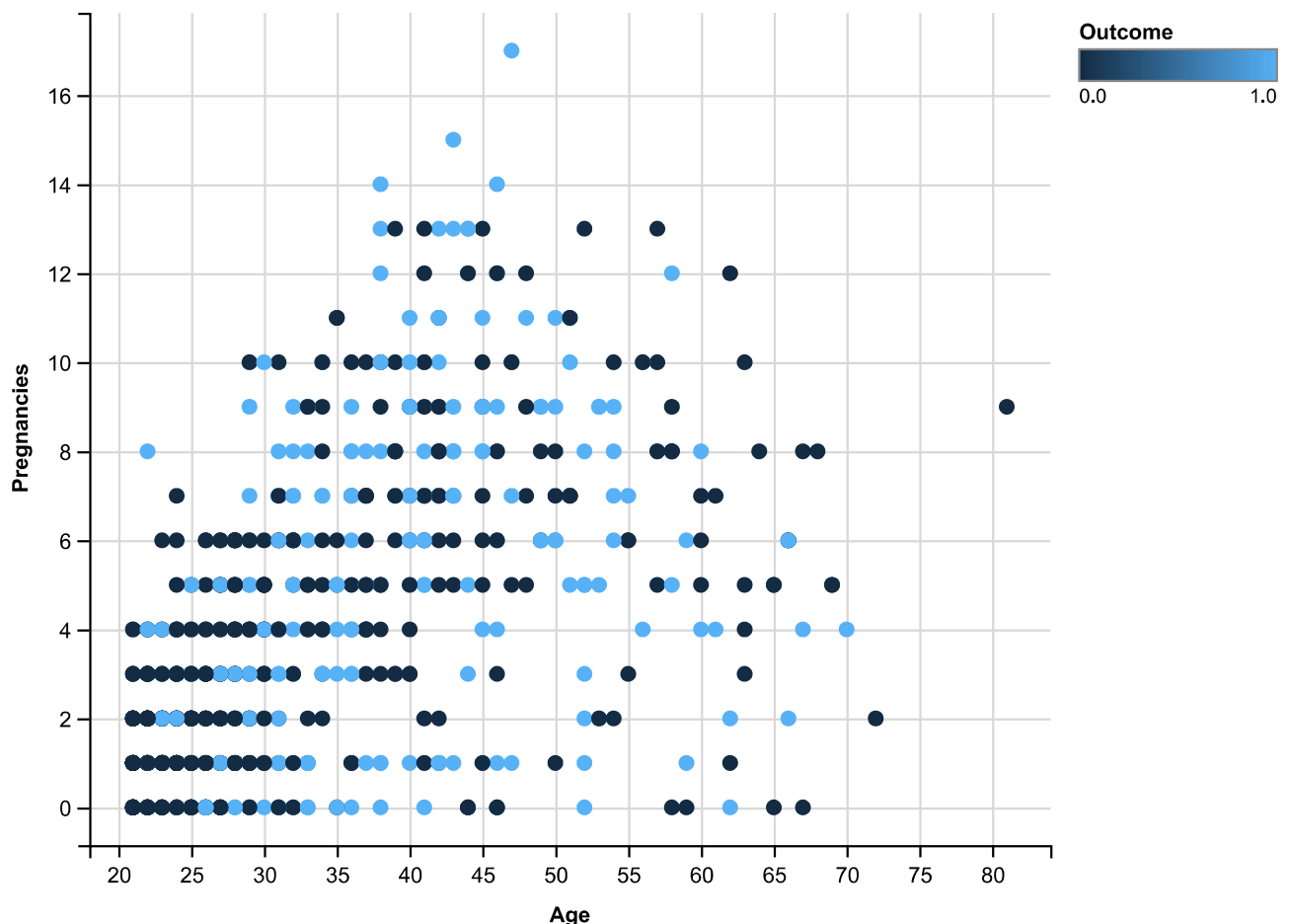
```
data %>% ggvis(~BMI,~DiabetesPedigreeFunction,fill =~Outcome) %>% la
```



Age and Pregnancies

The males have 0 for the pregnancy attribute, which is why we find a lot of values plottinh zero in this grpah.

```
data %>% ggvis(~Age,~Pregnancies,fill =~Outcome) %>% layer_points()
```

Preparing the data

The dataset is divided as two parts, training data and testing data, with a Splitratio of 0.75. It means that 2/3rds of the data is labelled by training set and the rest 1/3rd of data is the testing set. The division of the dataset is by means of a random order generated by the seed.

```
set.seed(88)
split <- sample.split(data$Outcome, SplitRatio = 0.75)
data_train <- subset(data, split == TRUE)
data_test <- subset(data, split == FALSE)
```

Logistic regression

The Logistic regression helps to classify the concern person will get diabetes or not. Since we are using the logistic regression we have to mention that, family = binomial. We are using all the attributes we have in the dataset. Let us take a look at the summary.

```
model <- glm (Outcome ~ .-Pregnancies + Glucose + BloodPressure + Sk
summary(model)
```

```
##
## Call:
## glm(formula = Outcome ~ . - Pregnancies + Glucose + BloodPressure
##      SkinThickness + Insulin + BMI + DiabetesPedigreeFunction +
##      Age, family = binomial, data = data_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -8.3339721    0.8159489  -10.214  < 2e-16 *
## Glucose         0.0382162    0.0044235   8.639  < 2e-16 *
## BloodPressure  -0.0088309    0.0060059  -1.470   0.1415
## SkinThickness   0.0007624    0.0081902   0.093   0.9258
## Insulin        -0.0017095    0.0010823  -1.580   0.1142
## BMI             0.0792632    0.0169318   4.681 2.85e-06 *
## DiabetesPedigreeFunction 0.7386714    0.3332368   2.217  0.0266 *
## Age            0.0204344    0.0095270   2.145  0.0320 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 745.11  on 575  degrees of freedom
## Residual deviance: 552.82  on 568  degrees of freedom
## AIC: 568.82
##
## Number of Fisher Scoring iterations: 5
```

Prediction

The trained model is used to predict the data for the testing data and for the training data(For checking accuracy purposes and for ROC curve)

```
predict_train <- predict(model, type = 'response')
```

```
predict_test <- predict(model, newdata = data_test, type = 'response')
```

Interpretation of Logistic Regression Model

The logistic regression model is used to predict the likelihood of a person developing diabetes based on various attributes. Here's the interpretation of the coefficients:

- **Glucose:** For every unit increase in glucose level, the log-odds of having diabetes increase by 0.0382. This suggests that higher glucose levels are associated with an increased risk of diabetes.
- **BloodPressure:** The coefficient for blood pressure is -0.0088, but it is not statistically significant (p-value = 0.1415). This suggests that blood pressure may not be a significant predictor of diabetes in this model.
- **SkinThickness** and **Insulin:** Both SkinThickness and Insulin have coefficients that are not statistically significant (p-values > 0.05). This indicates that these variables may not be good predictors of diabetes in this model.
- **BMI:** For every one unit increase in BMI, the log-odds of having diabetes increase by 0.0793. This suggests that higher BMI is associated with an increased risk of diabetes.
- **DiabetesPedigreeFunction:** For every one unit increase in the diabetes pedigree function, the log-odds of having diabetes increase by 0.7387. This suggests that a higher diabetes pedigree function value is associated with an increased risk of diabetes.
- **Age:** For every one year increase in age, the log-odds of having diabetes increase by 0.0204. This suggests that older individuals are more likely to have diabetes.

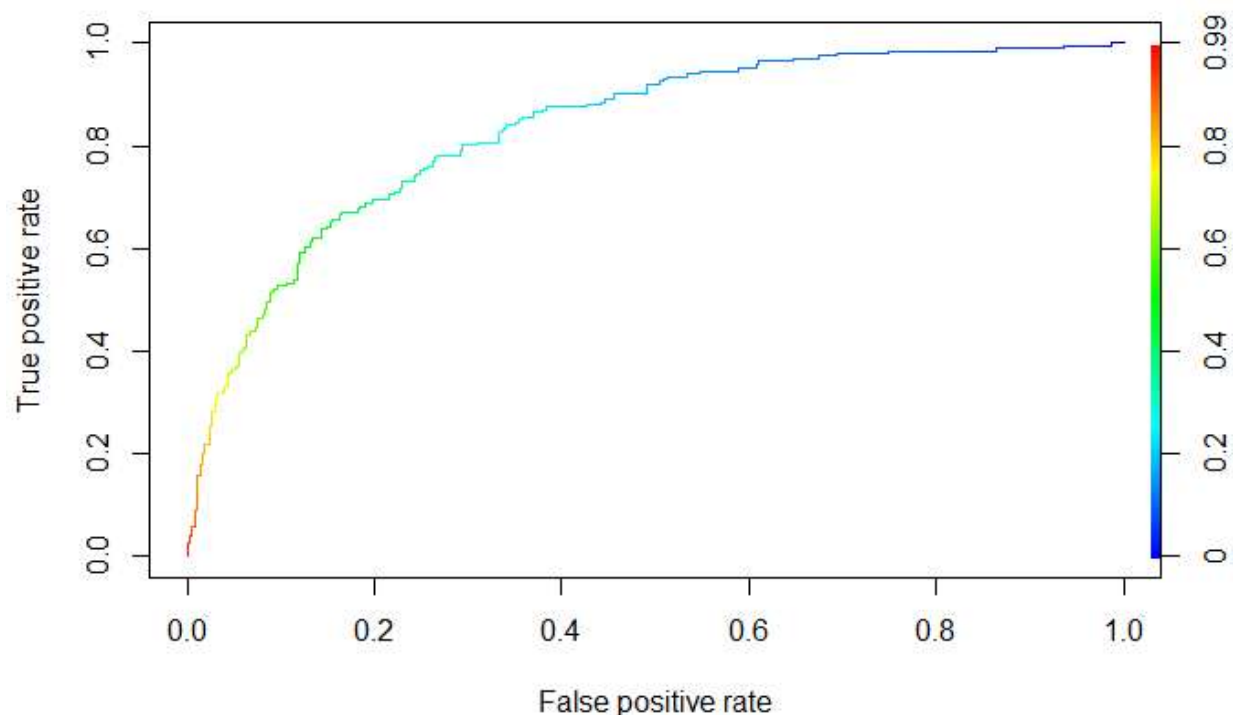
Model Performance

The model's performance can be assessed using the residual deviance, which is 552.82 on 568 degrees of freedom. Lower deviance indicates a

better fit of the model to the data. The null deviance, which represents the deviance of the null model (with no predictors), is 745.11 on 575 degrees of freedom. The difference between the null and residual deviance is used to assess the model's goodness of fit.

ROC Curve

```
ROCRpred <- prediction(predict_train, data_train$Outcome)
ROCRperf <- performance(ROCRpred, 'tpr', 'fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2, 1.7))
```



Comparison

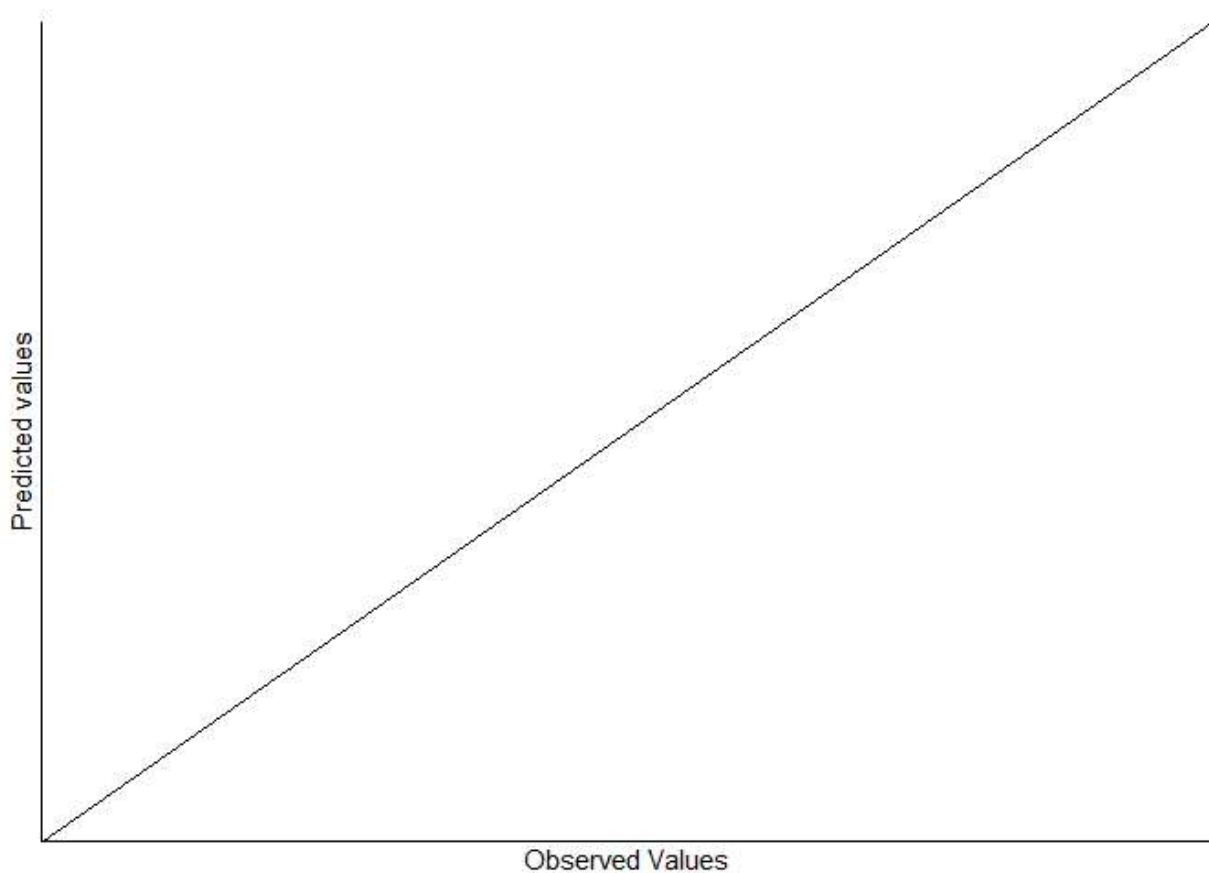
By comparing the real values with the real data, we can see the how our machine learning algorithm performs.

```
predict_test_c = predict_test
i = 1
while(i <= length(predict_test))
{
```

```

if(predict_test[i] < 0.5)
  predict_test_c[i] = 0
else
  predict_test_c[i] = 1
i = i + 1;
}
compare <- data.frame(data_test$Outcome,predict_test_c)
colnames(compare) <- c("Observed Values","Predicted values")
ggplot(data = compare,aes(x = "Observed Values", y = "Predicted values")) +
  labs(x = "Observed Values", y = "Predicted values") + theme_classic()

```



compare

##	Observed Values	Predicted values
## 6	0	0
## 8	0	0
## 9	1	1
## 10	1	0
## 11	0	0
## 14	1	1

## 16	1	0
## 32	1	1
## 33	0	0
## 37	0	0
## 38	1	0
## 41	0	1
## 44	1	1
## 45	0	1
## 46	1	1
## 49	1	0
## 56	0	0
## 60	0	0
## 62	1	0
## 70	0	0
## 73	1	1
## 77	0	0
## 78	0	0
## 84	0	0
## 86	0	0
## 89	1	1
## 91	0	0
## 93	0	0
## 94	1	0
## 95	0	0
## 102	0	0
## 103	0	0
## 105	0	0
## 110	1	0
## 111	1	1
## 114	0	0
## 124	0	0
## 128	0	0
## 130	1	0
## 142	0	0
## 143	0	0
## 150	0	0
## 153	1	1
## 163	0	0
## 164	0	0
## 168	0	0
## 182	0	0
## 192	0	0

## 195	0	0
## 199	1	0
## 201	0	0
## 204	0	0
## 209	0	0
## 216	1	1
## 219	1	0
## 225	0	0
## 227	0	0
## 228	1	1
## 236	1	1
## 239	1	1
## 240	0	0
## 244	1	0
## 256	1	0
## 262	1	1
## 264	0	1
## 272	0	0
## 280	0	0
## 281	1	1
## 283	0	0
## 285	1	0
## 291	0	0
## 292	1	0
## 299	1	0
## 304	1	1
## 312	0	0
## 315	1	0
## 323	1	0
## 326	0	0
## 327	1	0
## 341	0	0
## 342	0	0
## 343	0	0
## 344	0	0
## 346	0	0
## 350	1	0
## 356	1	1
## 357	1	0
## 358	1	1
## 363	0	0
## 364	1	1

## 367	1	0
## 374	0	0
## 379	1	1
## 381	0	0
## 382	0	0
## 388	1	0
## 391	0	0
## 392	1	1
## 395	1	1
## 396	0	0
## 408	0	0
## 414	0	0
## 417	0	0
## 419	0	0
## 422	0	0
## 424	0	0
## 431	0	0
## 432	0	0
## 433	0	0
## 436	1	1
## 437	0	1
## 439	0	0
## 448	0	0
## 449	1	0
## 450	0	0
## 451	0	0
## 453	0	0
## 456	1	1
## 463	0	0
## 466	0	0
## 473	0	0
## 478	0	0
## 493	0	0
## 498	0	0
## 500	0	1
## 504	0	0
## 508	0	0
## 509	0	0
## 513	0	0
## 531	0	0
## 532	0	0
## 533	0	0

## 536	1	1
## 538	0	0
## 542	1	0
## 543	1	0
## 548	0	0
## 550	0	1
## 562	1	1
## 563	0	0
## 567	0	0
## 573	0	0
## 577	0	0
## 580	1	1
## 583	0	0
## 585	1	0
## 586	0	0
## 592	0	0
## 599	1	1
## 606	0	0
## 608	0	0
## 610	0	0
## 623	0	1
## 625	0	0
## 627	0	0
## 636	1	0
## 639	1	0
## 640	0	0
## 652	0	0
## 655	0	0
## 663	1	1
## 664	1	1
## 665	1	0
## 671	0	1
## 673	0	0
## 675	0	0
## 680	0	0
## 681	0	0
## 691	0	0
## 694	1	1
## 695	0	0
## 700	0	1
## 703	1	1
## 711	0	0

## 714	0	0
## 719	0	0
## 721	0	0
## 724	0	0
## 728	0	0
## 736	0	0
## 740	1	0
## 741	1	1
## 744	1	1
## 746	0	0
## 747	1	1
## 749	1	1
## 753	0	0
## 757	0	0
## 759	0	0
## 760	1	1
## 764	0	0
## 766	0	0

Interpretation of Model Predictions

The logistic regression model was trained on the diabetes dataset to predict the likelihood of an individual having diabetes based on several features such as Glucose, BloodPressure, BMI, etc. Here's a brief interpretation of the model and its predictions:

1. Model Coefficients:

- Glucose has a positive coefficient of 0.038, indicating that as Glucose levels increase, the odds of having diabetes also increase.
- BMI has a positive coefficient of 0.079, suggesting that higher BMI is associated with higher odds of diabetes.
- DiabetesPedigreeFunction also has a positive coefficient of 0.739, indicating that a higher genetic predisposition to diabetes increases the odds.

2. Model Performance:

- The model's performance was evaluated using the ROC curve, which shows the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity).

- The area under the ROC curve (AUC) provides a measure of the model's ability to discriminate between positive and negative cases. A higher AUC indicates better performance.

3. Predictions:

- The model was used to predict the likelihood of diabetes in both the training and test datasets.
- The predictions were compared with the actual outcomes to assess the model's accuracy.

4. Comparison of Observed vs. Predicted:

- The comparison table shows the observed values (actual diabetes outcomes) and the predicted values by the model.
- The model's predictions were generally accurate, but there were some cases where the predictions differed from the actual outcomes.

5. Future Improvements:

- The model could be further improved by incorporating additional features or by fine-tuning the model parameters.
- Regular updates and retraining of the model using new data can help maintain its accuracy over time.

```
# Confusion Matrix
conf_matrix <- table(data_test$Outcome, predict_test_c)
conf_matrix
```

```
##      predict_test_c
##           0      1
## 0 116      9
## 1  30     37
```

```
# Accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
## [1] 0.796875
```

```
# Precision
precision <- conf_matrix[2, 2] / sum(conf_matrix[, 2])
precision
```

```
## [1] 0.8043478
```

```
# Recall
recall <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
recall
```

```
## [1] 0.5522388
```

```
# F1-score
f1_score <- 2 * (precision * recall) / (precision + recall)
f1_score
```

```
## [1] 0.6548673
```

Model Evaluation Metrics

1. Accuracy:

- The accuracy of the model is calculated as the ratio of correctly predicted observations to the total observations.
- The accuracy of the model is approximately 0.797 or 79.7%.

2. Precision:

- Precision measures the proportion of correctly predicted positive cases out of all predicted positive cases.
- The precision of the model is approximately 0.804 or 80.4%.

3. Recall (Sensitivity):

- Recall measures the proportion of correctly predicted positive cases out of all actual positive cases.
- The recall of the model is approximately 0.552 or 55.2%.

4. F1-Score:

- The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.
- The F1-score of the model is approximately 0.655.

Interpretation

- The confusion matrix shows that the model predicted 116 true negatives, 9 false positives, 30 false negatives, and 37 true positives.
- The model's accuracy indicates that it correctly predicted the outcome for approximately 79.7% of the cases in the test set.
- Precision suggests that when the model predicted an individual to have diabetes, it was correct about 80.4% of the time.
- Recall indicates that the model correctly identified 55.2% of all actual positive cases.
- The F1-score provides a balanced measure of the model's precision and recall, giving an overall performance metric of approximately 0.655.

Conclusion

Based on the logistic regression model, glucose, BMI, diabetes pedigree function, and age are significant predictors of diabetes, while blood pressure, skin thickness, and insulin may not be significant predictors. This information can be used to identify individuals at risk of developing diabetes and implement preventive measures. The results can be improved by applying the feature scaling and data cleaning. From this analysis we predict the type 2 diabetes, commonly called as diabetes mellitus. As a result it can help to improve their health conditions.