Sauda Haywood

Predicting Diabetes Using Medical Records Data

Introduction:

For this final project, The goal is to develop a model for predicting whether an individual has diabetes or not, based on an individual medical record data. This project will leverage the power of data science to enhance the understanding of diabetes and improve early diagnosis. Someone would be interested in this topic, because diabetes is a widespread and chronic health condition that affects millions of people worldwide. With the use of data science, diabetes can be detected and prevented at early stage, which is crucial for managing and reducing the impact of diabetes on individuals and healthcare systems. This is a data science problem because it will be solved by following data science steps. Collecting, cleaning, processing, and analyzing datasets to get insights that will help identify individuals at risk of diabetes.

What is Diabetes

Diabetes is a prevalent and chronic medical condition that affects millions of people worldwide, making it a significant public health concern. It is essential to understand the nature of this disease to effectively address its prevention, management, and treatment. Diabetes, as defined by the American Diabetes Association, is a group of metabolic disorders characterized by elevated blood sugar levels over an extended period. These elevated blood sugar levels can result from either insufficient insulin production, ineffective utilization of insulin, or a combination of both (American Diabetes Association, 2022).

Project Statement:

The project aimed to predict the likelihood of someone having diabetes based on various medical history factors such as age, glucose level, pregnancies, insulin level, BMI, diabetes pedigree function, and skin thickness. This was achieved by constructing a predictive model.

Project Approach

To address this problem, a dataset containing information such as pregnancies, glucose levels, blood pressure, age, and other medical history was obtained from Kaggle. The dataset underwent initial processing, including cleaning steps to remove duplicates and outliers. Once the data was prepared, analysis and visualization were conducted. Subsequently, logistic regression was utilized to explore the relationships between predictor variables and the likelihood of diabetes. Research Questions:

1. How to predict diabetes based on medical records data?
2. Why is it important to use data science to help detect diabetes
3. How accurate are the predictions using the data science models.
4. Which are the trends in the daily activity data that can be linked to diabetes ?
5. Can Data Science provide tool that provides personalized recommendations for individuals to prevent diabetes

Required Packages:

For this project , I will use these packages dplyr, ggplot2, tidyr Metrics caTools

Data importing and cleaning

The primary data set used are “Pima Indians Diabetes Database” (https://www.kaggle.com/datasets/uciml/ pima-indians-diabetes-database)

This dataset were originally sourced from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), and they serve the purpose of diagnostically predicting the presence or absence of diabetes in patients. This dataset contains a set of independent variables, and one dependent outcome variable. The independent variables include:

- Pregnancies
- Glucose
- Blood Pressure
- Skin Thickness
- Insulin
- BMI (Body Mass Index)
- Diabetes Pedigree Function
- Age

The outcome variable is binary, with values of 1 or 0, 1 indicating the presence of diabetes (1) or o indicating the absence of diabetes (0) in the individual.

```
# Load necessary libraries
library(dplyr) library(readr)
data <- read_csv("diabetes data 1.csv")
```

Data Understanding

Pregnancies: Represents the number of times a woman has experienced pregnancy. Glucose: Denotes the plasma glucose concentration measured 2 hours into an oral glucose tolerance test. BloodPressure: Signifies the diastolic blood pressure, measured in millimeters of mercury SkinThickness: Reflects the thickness of the triceps skin fold, measured in millimeters Insulin: Represents the 2-hour serum insulin level, measured in micro international units per milliliter BMI: Stands for Body Mass Index, which is calculated as the weight in kilograms divided by the square of height in meters Age: Represents the age of the individual in years. DiabetesPedigreeFunction: This variable provides a score that assesses the likelihood of diabetes based on the individual’s family medical history. Outcome: This variable has two possible values, 0 indicating the absence of diabetes and 1 indicating the presence of diabetes.

Data Inspection

```
# Display basic information about the dataset str(data)
```

```
## spc_tbl_ [768 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ Pregnancies                : num [1:768] 6 1 8 1 0 5 3 10 2 8 ...
## $ Glucose                    : num [1:768] 148 85 183 89 137 116 78 115 197 125 ...
## $ BloodPressure              : num [1:768] 72 66 64 66 40 74 50 0 70 96 ...
## $ SkinThickness              : num [1:768] 35 29 0 23 35 0 32 0 45 0 ...
## $ Insulin                    : num [1:768] 0 0 0 94 168 0 88 0 543 0 ...
## $ BMI                        : num [1:768] 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
## $ DiabetesPedigreeFunction: num [1:768] 0.627 0.351 0.672 0.167 2.288 ...
## $ Age                        : num [1:768] 50 31 32 21 33 30 26 29 53 54 ...
```

```
## $ Outcome                                 : num [1:768] 1 0 1 0 1 0 1 0 1 1 ...
## - attr(*, "spec")= ##       ..
cols(
##       ..       Pregnancies = col_double(),
##       ..       Glucose = col_double(),
##       ..       BloodPressure = col_double(),
##       ..       SkinThickness = col_double(),
##       ..       Insulin = col_double(),
##       ..       BMI = col_double(),
##       ..       DiabetesPedigreeFunction = col_double(),
##       ..       Age = col_double(),
##       ..       Outcome = col_double()
##       .. )
## - attr(*, "problems")=<externalptr>
summary(data)

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


## function (object, ...)
## UseMethod("summary")
## <bytecode: 0x10c8802e8>
## <environment: namespace:base>

# Determining how many rows and columns dimensions <-
dim(df)
```

Summary: Minimum value for some colomums is zero which means some data are missing. and that need to be fixed during the data handling step. There is also outliers in insulin. And those need to be fixed in the next step.

Handling Missing Data

```
# Remove duplicate rows

library(dplyr) clean_data1 <-

distinct(data) # Remove missing

value

missing_counts <- colSums(is.na(clean_data1)) print(missing_counts)
```

```
##                 Pregnancies                     Glucose               BloodPressure
##                           0                           0                           0
##               SkinThickness                     Insulin                         BMI
##                           0                           0                           0
```

```
# Replace zero values in 'Glucose' with the mean clean_data1$Glucose[clean_data1$Glucose == 0] <-
mean(clean_data1$Glucose, na.rm = TRUE)

# Replace zero values in 'BloodPressure' with the mean clean_data1$BloodPressure[clean_data1$BloodPressure == 0
<- mean(clean_data1$BloodPressure,

# Replace zero values in 'SkinThickness' with the mean clean_data1$SkinThickness[clean_data1$SkinThickness == 0] <
mean(clean_data1$SkinThickness,

# Replace zero values in 'Insulin' with the mean clean_data1$Insulin[clean_data1$Insulin == 0] <-
mean(clean_data1$Insulin, na.rm = TRUE)

# Replace zero values in 'BMI' with the median clean_data1$BMI[clean_data1$BMI == 0] <-
mean(clean_data1$BMI, na.rm = TRUE) print(clean_data1)
```

```
##  DiabetesPedigreeFunction                     Age                 Outcome
##                         0                       0                       0
```

```
##
## # A tibble: 768 x 9
##    Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##          <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
## 1            6     148            72            35    79.8  33.6
## 2            1      85            66            29    79.8  26.6
```

```
## 3          8     183          64          20.5     79.8 23.3
## 4          1      89          66          23       94   28.1
## 5          0     137          40          35      168   43.1
## 6          5     116          74          20.5     79.8 25.6
## 7          3      78          50          32       88   31
## 8         10     115          69.1        20.5     79.8 35.3
## 9          2     197          70          45      543   30.5
## 10         8     125          96          20.5     79.8 32.0
```

i 758 more rows
i 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <dbl> Summary: :

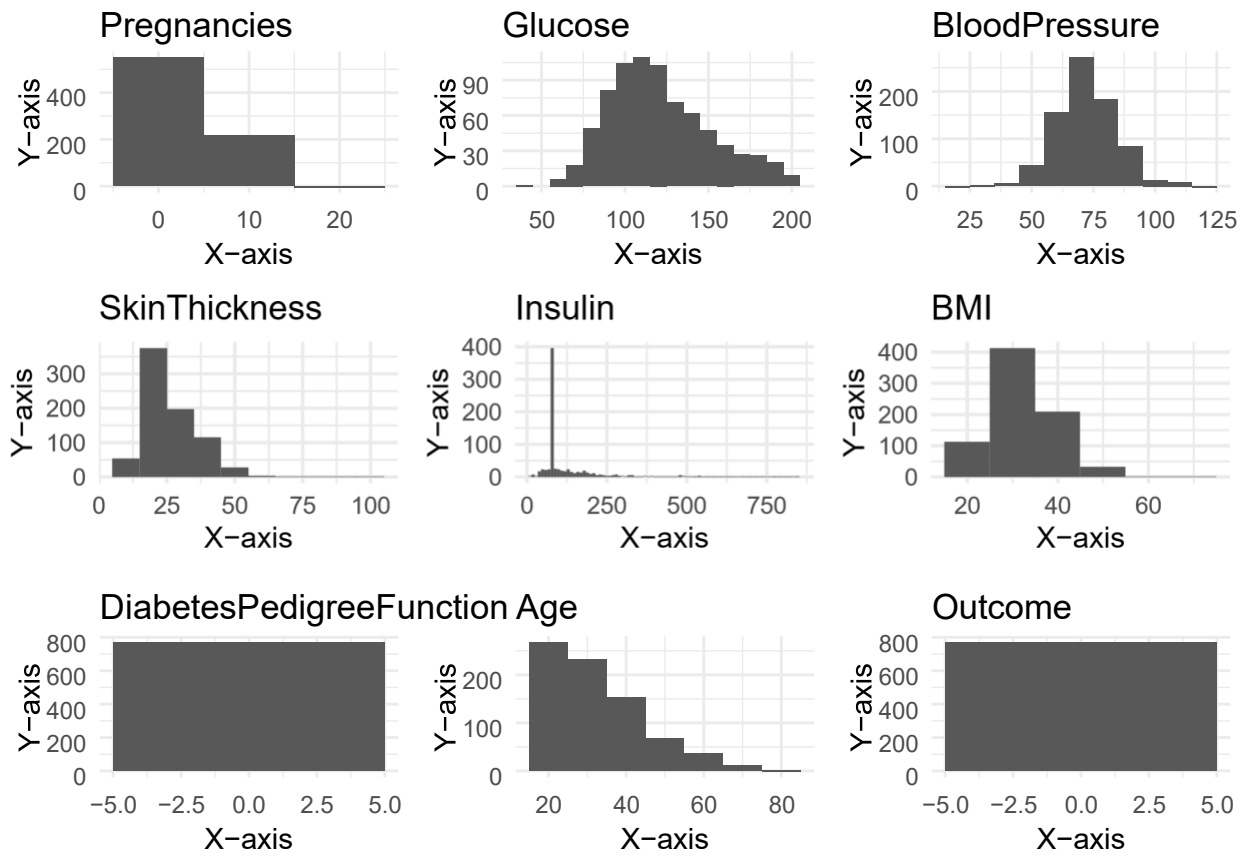
- Duplicate rows were removed
- There is no missing value in the new dataset
- Zero values have been replaced with the mean value

Data Visualization

```
library(ggplot2) library(dplyr)
library(tidyr)


# Melt the dataframe to make it longer for plotting
melted_df <- clean_data1 %>% gather(key = "variable",
   value = "value")

# Plotting histograms
plots <- list() for (col in unique(melted_df$variable))
{
   p <- ggplot(melted_df %>% filter(variable == col), aes(x = value)) +
      geom_histogram(binwidth = 10) + ggtitle(col) + theme_minimal() +
      labs(x = "X-axis", y = "Y-axis")
   plots[[col]] <- p
}

# Displaying all histograms using grid.arrange library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##         combine
```

```
grid.arrange(grobs = plots, ncol = 3)
```

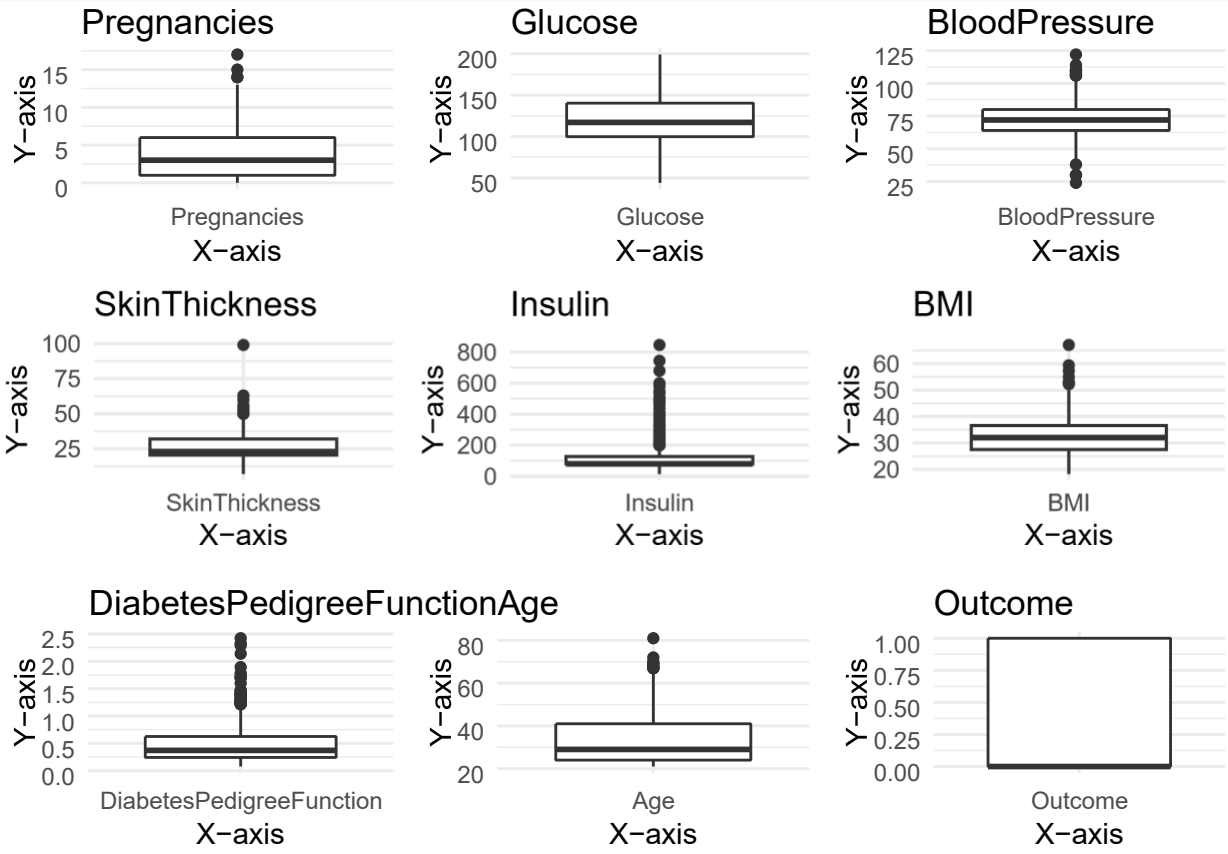


Summary: :

- From the plot, only blood pressure and glucose are normally distributed

```
library(ggplot2) library(dplyr)
library(tidyr)


# Melt the dataframe to make it longer for plotting
melted_df <- clean_data1 %>% gather(key = "variable",
   value = "value")

# Creating box plots
plots <- list() for (col in unique(melted_df$variable))
{
   p <- ggplot(melted_df %>% filter(variable == col), aes(x = variable, y = value)) + geom_boxplot() +
      ggtitle(col) + theme_minimal() +
```

```
    labs(x = "X-axis", y = "Y-axis")
  plots[[col]] <- p
}

# Displaying all box plots using grid.arrange
library(gridExtra) grid.arrange(grobs = plots,
ncol = 3)
```



Summary: :

- From the plot, only blood pressure and glucose are normally distributed. Others are skewed and present outliers.

Data Analysis Using a Predictive Model

```
# Load necessary libraries
library(dplyr) library(caTools)

# Split the data into training and testing sets
set.seed(123)
split <- sample.split(clean_data1$Outcome, SplitRatio = 0.7) train_data <-
subset(clean_data1, split == TRUE) test_data <- subset(clean_data1, split ==
FALSE)
```

Logistic Regression - Model Trainin

```
# Train a logistic regression model using the training data model <- glm(Outcome ~ ., data = train_data, family = binomial) # Summary of the model summary(model)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = binomial, data = train_data)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -9.691066   1.011411 -9.582 < 2e-16 ***
## Pregnancies                   0.109199   0.040340   2.707 0.00679 **
## Glucose                       0.044479   0.005060   8.790 < 2e-16 ***
## BloodPressure                -0.014586   0.010563 -1.381 0.16732
## SkinThickness                 0.006890   0.015100   0.456 0.64818
## Insulin                      -0.003756   0.001455 -2.580 0.00987 **
## BMI                           0.106359   0.022703   4.685 2.8e-06 ***
## DiabetesPedigreeFunction 0.760805        0.378096   2.012 0.04420 *
## Age                           0.015792   0.011680   1.352 0.17637
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##                 Null deviance: 696.28 on 537 degrees of freedom
## Residual deviance: 476.93 on 529 degrees of freedom
## AIC: 494.93
##
## Number of Fisher Scoring iterations: 5
```

Evaluation and Prediction of the Model

```
# Predict using the test data predicted_values <- predict(model, newdata = test_data, type = "response") # Convert predicted probabilities to binary outcomes (0 or 1) predicted_classes <- ifelse(predicted_values > 0.5, 1, 0) # Compare predicted classes with actual classes to evaluate the model
confusion_matrix <- table(predicted_classes, test_data$Outcome) print(confusion_matrix)
```

```
##
## predicted_classes   0   1
##                 0 127 35
```

```
##                          1 23 45
# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy:", accuracy))
```

```
## [1] "Accuracy: 0.747826086956522"
```

Summary

- Based on their coefficients and significance levels, Glucose, BMI, and Pregnancies appear to be more significant in predicting the outcome of diabetes
- DiabetesPedigreeFunction has the highest ablsolute value of coefficient (0.76080522). Which indicates that a one-unit increase in the Diabetes Pedigree Function score leads to a high increase of having diabetes.
- Also Glucose has a relatively large coefficient (0.044479074), which indicates that increases in glucose levels significantly increase the likelihood of having diabetes.
- BMI has 0.106358794 coefficient, which indicates that higher BMI values leads to diabetes.

P Values

- Glucose, BMI, DiabetesPedigreeFunction have low p-values which means they are statistically significant in predicting diabetes
- Pregnancies and Insulin have relatively low p-values. Thus they are still significance, but slightly less significant compared to Glucose, BMI, and DiabetesPedigreeFunction.
- BloodPressure, SkinThickness and Age have higher p-values, indicating that they are not significantly in predicting the diabete in this model.
- The model's accuracy is approximately 74.78%.

Insights:

The analysis indicated that variables like Glucose, BMI, and the Diabetes Pedigree have high significance in predicting diabetes. these was concluded based on their coefficients in the model.Also, it was noted that the model has 74.78% accuracy, indicating its capability to predict diabetes status from the provided data.

Implications:

This analysis can be applied by healthcare professionals during the diabetes screening process or by individuals concerned about diabetes who wish to know their status. This information can empower them to make lifestyle changes.

Limitations and Recommendations for Improvement:

First, obtaining data to use was a challenge. And when data was obtained, the data set was not representing the entire population, which can limit the model's generalizability. To improve in the future, acquiring more diverse data

from various sources and exploring other predictive models like Random Forests to enhance predictive accuracy, could be beneficial.

Conclusion:

In conclusion, this logistic regression model provided valuable insights into predicting diabetes. The model indicates that Glucose, BMI, and DiabetesPedigreeFunction have a significant impact on predicting the likelihood of diabetes.

References

Pima Indians Diabetes Database: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

American Diabetes Association. “Diabetes Basics.” 2022. [https://www.diabetes.org/diabetes]

Centers for Disease Control and Prevention. “National Diabetes Statistics Report, 2020.” 2021. [https://www.cdc.gov/diabetes/pdfs/data/statistics/national-diabetes-statistics-report.pdf]