Statistical Analysis for Developing a Diabetes Prediction Model Using R

Project by Subarna Biswas

29 April, 2024

Abstract

Diabetes, a metabolic disorder characterized by elevated blood sugar levels, affects millions of people worldwide. The objective of this project is to showcase the application of statistical analysis in predicting an individual's likelihood of having diabetes based on medical data. Additionally, the aim is to develop and compare several prediction models to identify the most effective one that can determine whether a patient has diabetes by analyzing specific diagnostic measurements included in the dataset. Various techniques will be explored to enhance the performance and accuracy of the prediction model.

1 Introduction:

Diabetes can be attributed to a combination of genetic susceptibility and environmental factors. Prolonged overweight can lead to diabetes, as well as being born into a family with a history of the condition. Furthermore, the risk of developing type 2 diabetes tends to increase gradually as we age. Several potential complications are associated with diabetes, including cardiovascular diseases such as high blood pressure, heart failure, stroke, and even death resulting from heart attacks or other conditions related to arterial hypertension.When it comes to women and diabetes, there are several unique considerations to explore.

There is an intriguing aspect regarding diabetes in Pima Indian heritage. The Pima Indians, particularly those living in the Gila River Indian Community in Arizona, have one of the highest rates of type 2 diabetes in the world. The selected data is on 768 pima Indian women which collected information about eight diabetes causing factors.

The dependent variable in the data is qualitative, indicating the presence or absence of diabetes. Predicting qualitative responses is known as classification. There are various classification techniques, or classifiers, that can be utilized for this purpose. In this project, I employed some widely-used classifiers, including

logistic regression, linear discriminant analysis, quadratic discriminant analysis, and naive Bayes, to predict diabetes.

2 Data Descriptions:

The dataset utilized in this project is known as the Pima Indigenous Diabetes database, which was sponsored and published by the National Institute of Diabetes, Digestive and Kidney Diseases in the United States of America. It is publicly accessible on the Kaggle website (https://www.kaggle.com/uciml/pima-indians-diabetes-database) and serves as an open-source dataset comprising records of female patients. The dataset encompasses a total of 768 cases, with each case representing a female participant from the Pima Indigenous community.
The dataset contains 500 cases classified as non-diabetic and 268 cases classified as diabetic. Additionally, the dataset includes the following eight features:

- Pregnancies : This numeric variable represents number of times a Pima Indigenous female got pregnant.
- Glucose : Plasma glucose concentration after 2hours in an oral glucose tolerance test.
- BloodPressure: Diastolic blood pressure (mm Hg). Also a numeric variable.
- SkinThickness: A numeric variable estimates triceps skinfold thickness, measured in millimeters (mm) within the dataset. This measurment offers a reliable estimate of both obesity and body fat distribution. It serves as a valuable indicator in assessing body composition and provides insights into the distribution of fat in the triceps region of the body.
- Insulin: In the context of the dataset, the variable labeled 'Insulin' represents the two-hour serum insulin level. By analyzing an individual's insulin levels following a meal, it is possible to identify the presence of a metabolic disorder and determine if there is a defect in islet function, both of which are associated with diabetes.
- BMI: Body mass index (BMI), a measure of obesity and health, is commonly used in statistical analysis. The degree of obesity cannot be judged directly by the absolute value of the weight, and it is naturally related to height. So, BMI is defined as the body mass divided by the square of the body height.

- DiabetesPedigreeFunction(DPF) : This numerical variable determines the genetic risk of diabetes based on family history. It considers the prevalence of diabetes among relatives to assess an individual's likelihood of developing the condition. A higher DPF score indicates a greater genetic predisposition to diabetes.
- Age (years): The age range of Pima Indigenous female in the dataset is from 21 to 81.
- Outcome : Binary classification variable where '0' means that a female does not have diabetes, and a '1' indicates the participant has diabetes.

3 Load Dataset and required libraries:

For data processing, model construction, and evaluation using R programming, I utilize the following libraries:

```
library(readr)
library(ggplot2)
library(corrplot)
library(caret)
library(e1071)
library(caTools)
library(MASS)
```

readr: for reading and loading .cvs data.
ggplot2 and corrplot: For data visualization and correlation plots to understand the data structure and correlations between dependent and independent variables.
caret: for confusion matrix analysis to assess accuracy and performance of each model fit.
caTools: provides utilities for partitioning data into training and testing sets.
Additionally, I will employ:
e1071: for implementing the Naive Bayes model on the diabetes data.
MASS: for fitting Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) models.

4 Data Visualization

Out of the total subjects, 500 were identified as non-diabetic, while 268 individuals were diagnosed with diabetes (Fig-1). Therefore, about 35% of the women in the

dataset are diagnosed with diabetes.

```
[ggplot(outcome_df, aes(x=Outcome, y=Count)) +
  geom_bar(stat="identity", fill="purple") +
  labs(x="Outcome(1= diabetes,0= no diabetes)",
       y="Total Counts for each outcomes") +
  theme_minimal() +
  theme(axis.text.x = element_text(size=12),
        axis.text.y = element_text(size=12),
        axis.title = element_text(size=12),
        plot.title = element_text(size=12, face="bold"))]
```

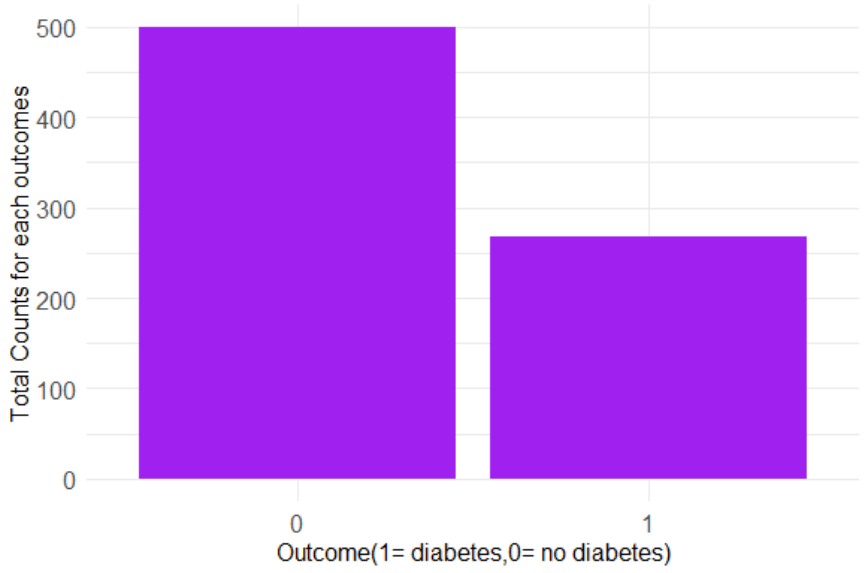

Figure 1: Distribution of Diabetes Outcomes

From the correlation matrix in Fig-2, we observe that the likelihood of having diabetes is highly correlated with the Glucose level. Additionally, we note that Skin Thickness and Blood Pressure have the lowest correlation with the outcome.

```
[num.var <- sapply(diabetes, is.numeric)
corr.matrix <- cor(diabetes[,num.var])
corrplot(corr.matrix, method="number")]
```

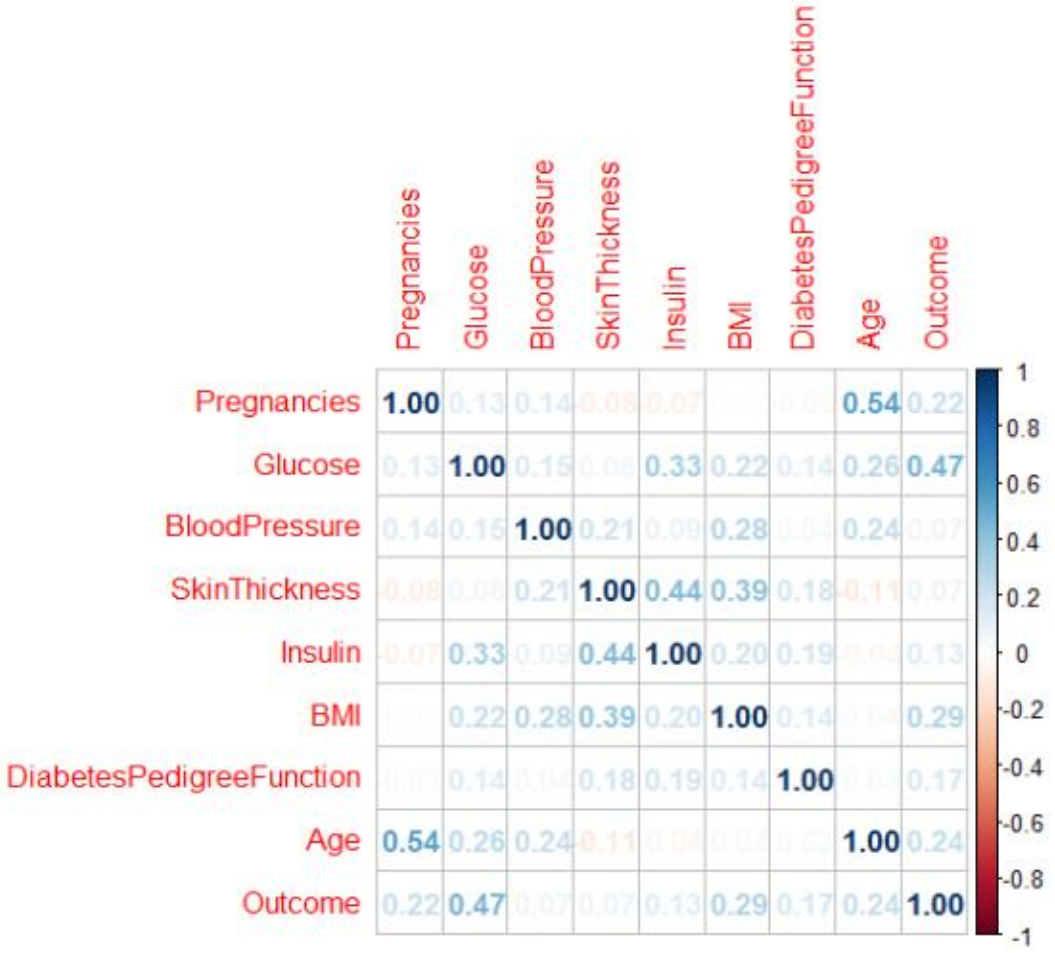

Figure 2: Co-relation between all the variables

Finally, the boxplots in Fig-3 are used to visualize the comparative relationship of each predictor in the case of having or not having diabetes. Blood pressure and skin thickness show little variation with diabetes, indicating weaker correlations. However, all other variables maintain more or less significant correlations with diabetes, as depicted in the boxplots.

```
[#Boxplots
attach(diabetes)
par(mfrow=c(2,4))
boxplot(Pregnancies~Outcome,
        main="Pregnancies vs. Diabetes",
        xlab="Outcome", ylab="Pregnancies",col="brown1")
boxplot(Glucose~Outcome,
        main="Glucose vs. Diabetes",
        xlab="Outcome", ylab="Glucose",col="deeppink")
boxplot(BloodPressure~Outcome,
```

```
        main="Blood Pressure vs. Diabetes",
        xlab="Outcome", ylab="Blood Pressure",col="green")
boxplot(SkinThickness~Outcome,
        main="Skin Thickness vs. Diabetes",
        xlab="Outcome", ylab="Skin Thickness",col="orange")
boxplot(Insulin~Outcome,
        main="Insulin vs. Diabetes",
        xlab="Outcome", ylab="Insulin",col="yellow")
boxplot(BMI~Outcome, main="BMI vs. Diabetes",
        xlab="Outcome", ylab="BMI",col="purple")
boxplot(DiabetesPedigreeFunction~Outcome,
        main="Diabetes Pedigree Function vs. Diabetes",
        xlab="Outcome", ylab="DiabetesPedigreeFunction",
        col="lightgreen")
boxplot(Age~Outcome, main="Age vs. Diabetes",
        xlab="Outcome", ylab="Age",col="cyan")
box(which = "outer", lty = "solid")]
```

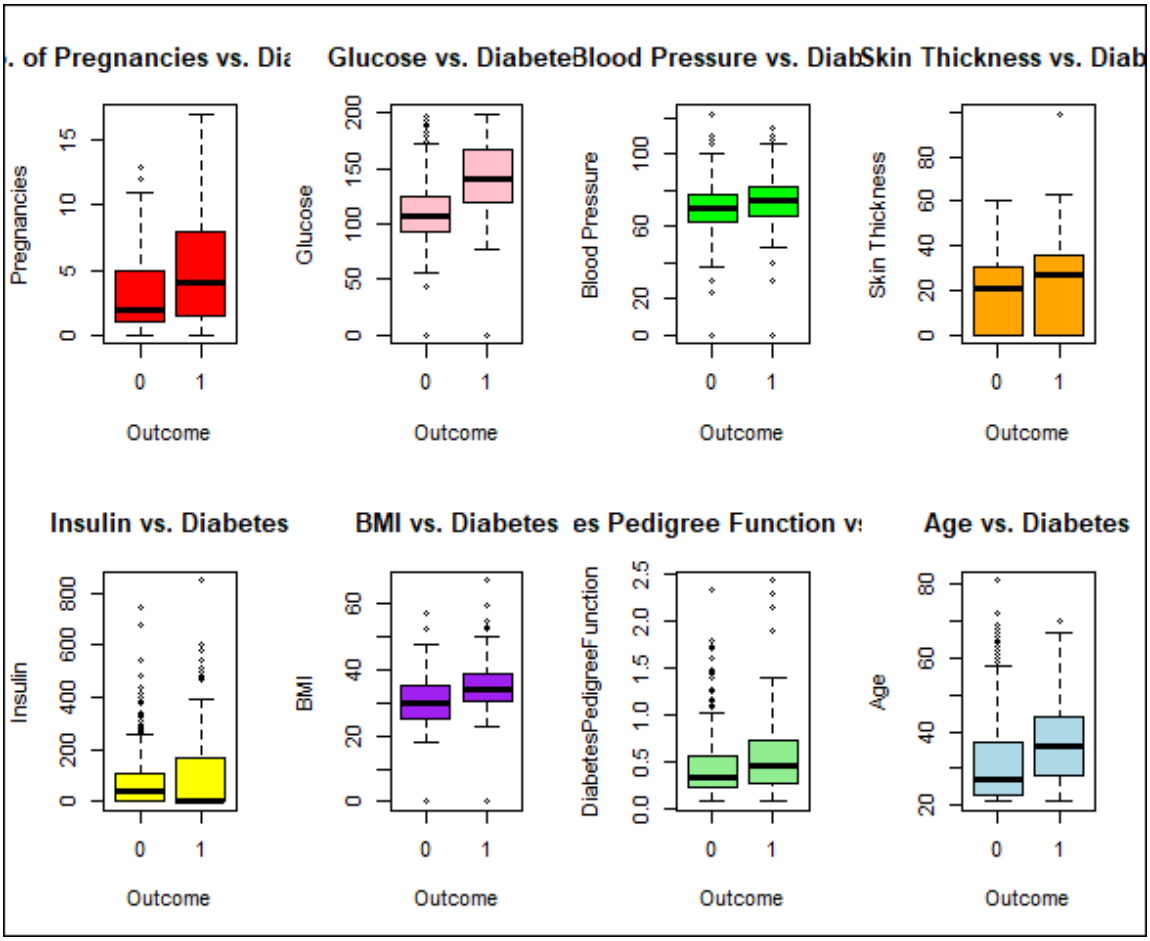

Figure 3: Relation of each predictors with the response.

Normality: According to the central limit theorem, in large samples (> 30 or 40), the sampling distribution tends to be normal, regardless of the shape of the data and the means of random samples from the distribution will themselves have normal distribution.With large enough sample sizes, the violation of the normality assumption should not cause major problems. [2] In the Diabetes dataset the sample size is 768. Therefore, consideration of normal distribution of the dataset will not affect the statistical analysis onward.

5 Methodology:

In this section, the diabetes data will be analyzed using various statistical tools in classification settings. A best model will be selected based on the highest accuracy to predict diabetes. In the selection of the best model, I will ensure that the model meets all assumptions to be reported as unbiased and generalizable outside the sample.

- The observations should be independent of each other. In other words, they should not arise from repeated measurements or matched data, a condition which is satisfied by our dataset.
- There should be little or no multicollinearity among the independent variables. This means that the independent variables should not be highly correlated with each other.
- The data should ideally be normally distributed, which typically requires a large sample size. With 768 observations in our dataset, we can reasonably consider the sample to be normally distributed. [1]

Create Training and Test Samples: To better assess the accuracy of the model, we can fit the model using part of the data and then examine how well it predicts the held-out data. In the following subsections, the data will be split into an 60% training dataset with 468 sample and a 40% testing dataset with 300 samples.

```
# split data
set.seed(123)
index <- sample(2, nrow(diabetes), prob = c(0.6, 0.4),
                replace = TRUE)
Diabetes_train <- diabetes[index==1, ]
Diabetes_test <- diabetes[index == 2, ]
outcome_test<- Diabetes_test$Outcome
```

```
outcome_train<- Diabetes_train$Outcome
print(dim(Diabetes_train))
print(dim(Diabetes_test))
```

5.1 Logistic regression model:

In the Diabetes dataset, the dependent variable "Outcome" exclusively consists of binary values, specifically 0 and 1. As a result, logistic regression emerges as the most straightforward approach to utilize. Logistic regression serves the purpose of forecasting the likelihood of certain conditions transpiring in binary scenarios. It enables the prediction of the probability of a categorical response transpiring based on the influence of one or more predictor variables.

The equation of a logistic regression model takes the following form:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+....+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+....+\beta_p X_p}} \quad (1)$$

Where, $X = (X_1, X_2...., X_p)$ are p predictors and from equation above we use maximum likelihood method to estimate the coefficients $\beta_0, \beta_1, ..., \beta_p$. [1]

The glm() function can be used to fit glm() many types of generalized linear models, including logistic regression. So here I used the glm (general linear model) function and specify family =" binomial" so that R fits a logistic regression model to the dataset.

```
[#regression model
model.fit <- glm(Outcome ~.,
          data = Diabetes_train, family = binomial)
summary(model.fit)
#prediction_logistic
model.pred <-predict(model.fit, Diabetes_test,
          type = "response")
model.pred_class <- ifelse(model.pred > 0.5, 1, 0)
table(model.pred_class,Diabetes_test$Outcome)
mean(model.pred_class==Diabetes_test$Outcome)]
```

The p-values in the output provides an idea of how effective each predictor variable is at predicting the probability of diabetes (Fig-4). It appears that the variables

"Pregnancies," "Glucose," "Blood Pressure," "Age," "BMI," and "Diabetes Pedigree Function" are considered important predictors. This conclusion is drawn from the observation that these variables have low p-values. A low p-value typically indicates that the relationship between the predictor variable and the outcome variable is statistically significant. Therefore, these variables are more likely to have a meaningful impact on predicting the outcome of interest, which in this case is diabetes.On the other hand, the variables "Skin Thickness" and "Insulin" are noted as not being statistically significant in the model. This means that there is insufficient evidence to suggest a significant relationship between these variables and the outcome variable, at least based on the current dataset.

```
> summary(model.fit)

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

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -9.0299378  0.9628419  -9.378  < 2e-16 ***
Pregnancies              0.0903025  0.0447221   2.019  0.04347 *
Glucose                  0.0376932  0.0050204   7.508  6.0e-14 ***
BloodPressure           -0.0170087  0.0076809  -2.214  0.02680 *
SkinThickness           -0.0047440  0.0090696  -0.523  0.60092
Insulin                 -0.0007931  0.0012155  -0.653  0.51406
BMI                      0.0979239  0.0207622   4.716  2.4e-06 ***
DiabetesPedigreeFunction 1.1844060  0.4027269   2.941  0.00327 **
Age                      0.0264114  0.0128694   2.052  0.04014 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 597.15  on 467  degrees of freedom
Residual deviance: 420.07  on 459  degrees of freedom
AIC: 438.07

Number of Fisher Scoring iterations: 5
```

Figure 4: Summary of proposed logistic regression model

Finally, a confusion matrix is created to assess the model's performance by comparing the predicted values to the test labels.

```
#confusion matrix
predictions <- factor(ifelse(model.pred > 0.5, 1, 0),
        levels = levels(as.factor(Diabetes_test$Outcome)))
confusionMatrix(predictions, as.factor(Diabetes_test$Outcome))
```

Accuracy: The percentage of correctly classified occurrences, or the overall accu-

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 162  42
         1  27  69

               Accuracy : 0.77
                 95% CI : (0.7182, 0.8164)
    No Information Rate : 0.63
    P-Value [Acc > NIR] : 1.417e-07

                  Kappa : 0.4925

 Mcnemar's Test P-Value : 0.09191

            Sensitivity : 0.8571
            Specificity : 0.6216
         Pos Pred Value : 0.7941
         Neg Pred Value : 0.7187
             Prevalence : 0.6300
         Detection Rate : 0.5400
   Detection Prevalence : 0.6800
      Balanced Accuracy : 0.7394

       'Positive' Class : 0
```

Figure 5: Confusion Matrix and statistics for Logistic Regression model

racy of the model. The accuracy in this instance is 77%.
No Information rate: The precision attained by consistently forecasting the most common class. The NIR in this instance is 63%.
P-Value $[Acc > NIR]$: The statistical test's p-value, which assesses how accurate the model is in relation to the NIR. The accuracy of the model is statistically substantially better than the NIR when the p-value is less than 0.05. The accuracy of the model in this instance is statistically considerably superior than the NIR, as indicated by the p-value of $1.4e^{-}7$.
Kappa: 0.49 (moderate agreement, -1 to 1) between the actual and predicted categories
P-Value for Mcnemar's Test: 0.09191 indicates that the gap between the model's predicted and actual classifications is not statistically significant.
Sensitivity: 85.71% accurately classifying as positive the proportion of real positive cases.
Specificity: Percentage of real negative cases that were appropriately labeled as

negative is 62.2%
Value of Pos Pred (PPV): The percentage of positive cases that are really anticipated to be positive is 79.41%
Neg Pred Value (NPV): The percentage of negative cases that are actually anticipated to be negative is 71.87%
Prevalence: The percentage of instances that are positive in the dataset are 63%
Detection Rate: Accurate classification of positive cases, independent of class imbalance is 54%.
Detection Prevalence: Percentage of all cases, independent of actual class, classed as positive 68%.
Balanced accuracy: Mean of specificity and sensitivity is 73.94%.

5.2 Linear Discriminant Analysis:

Now Linear discriminant analysis will be performed on the diabetes data. In R, we fit an LDA model using the lda() function, which is part of the MASS library. We fit the model using same train data that was used for previous model.

```
#LDA model
lda.fit<-lda(Outcome ~.,
             data = Diabetes_train)
lda.fit
#prediction
lda.pred <- predict(lda.fit, Diabetes_test)
lda.pred_class <- lda.pred$class
mean(lda.pred_class==Diabetes_test$Outcome)

#confusion matrix
table(lda.pred_class,Diabetes_test$Outcome)
confusionMatrix(table(lda.pred_class,Diabetes_test$Outcome))
```

This model provides 76.33% accuracy in predicting the diabetes which is good(Fig-6). If we see the Mcnemar's p-value, that specifies the gap between the model's predicted and actual classifications is 0.057 which is not significant but not so far from the significance level. All other model accessment parameters looks good.

```
Confusion Matrix and Statistics


lda.pred_class   0   1
             0 162  44
             1  27  67

               Accuracy : 0.7633
                 95% CI : (0.7111, 0.8103)
    No Information Rate : 0.63
    P-Value [Acc > NIR] : 5.424e-07

                  Kappa : 0.4758

 Mcnemar's Test P-Value : 0.05758

            Sensitivity : 0.8571
            Specificity : 0.6036
         Pos Pred Value : 0.7864
         Neg Pred Value : 0.7128
             Prevalence : 0.6300
         Detection Rate : 0.5400
   Detection Prevalence : 0.6867
      Balanced Accuracy : 0.7304

       'Positive' Class : 0
```

Figure 6: Confusion Matrix and statistics for LDA model

5.3 Quadratic Discriminant Analysis:

Next, QDA model fit considered to analyse the Diabetes data. QDA is implemented in R using the qda() function, which is also part of the MASS library.

```
#QDA model
qda.fit <- qda(Outcome ~., data = Diabetes_train)
qda.fit
#prediction
qda.pred <- predict(qda.fit, Diabetes_test)$class
mean(qda.pred == Diabetes_test$Outcome)
#confusion matrix
table(qda.pred, Diabetes_test$Outcome)
confusionMatrix(table(qda.pred, Diabetes_test$Outcome))
```

This model provides 70% accuracy in predicting the outcome which is good but less effective than the first model (fig-7). Also the gap between prediction and actual classification is significant in this case which compromise the model effectiveness.

```
Confusion Matrix and Statistics


qda.pred   0   1
       0 146  47
       1  43  64

               Accuracy : 0.7
                 95% CI : (0.6447, 0.7513)
    No Information Rate : 0.63
    P-Value [Acc > NIR] : 0.00653

                  Kappa : 0.3517

 Mcnemar's Test P-Value : 0.75183

            Sensitivity : 0.7725
            Specificity : 0.5766
         Pos Pred Value : 0.7565
         Neg Pred Value : 0.5981
             Prevalence : 0.6300
         Detection Rate : 0.4867
   Detection Prevalence : 0.6433
      Balanced Accuracy : 0.6745

       'Positive' Class : 0
```

Figure 7: Confusion Matrix and statistics for QDA model

5.4 Naive Bayes:

Finally, I fit a naive Bayes model to the diabetes data. Naive Bayes is implemented in R using the naiveBayes() function, which is part of the e1071 library. The syntax is identical to that of lda() and qda(). By default, this implementation of the naive Bayes classifier models each quantitative feature using a Gaussian distribution.

```
#Naivebayes model
model_naive <- naiveBayes(Outcome ~., data = Diabetes_train)
```

```
model_naive
#prediction
preds_naive <- predict(model_naive, Diabetes_test)
mean(preds_naive==Diabetes_test$Outcome)
#confusion matrix
table(preds_naive, Diabetes_test$Outcome)
confusionMatrix(table(preds_naive,Diabetes_test$Outcome))
```

Model accuracy is 71% which is an improvement compared to the QDA model but still this model is less effective than the logistic model.

```
Confusion Matrix and Statistics


preds_naive   0   1
          0 147  45
          1  42  66

               Accuracy : 0.71
                 95% CI : (0.6551, 0.7607)
    No Information Rate : 0.63
    P-Value [Acc > NIR] : 0.002163

                  Kappa : 0.3745

 Mcnemar's Test P-Value : 0.830218

            Sensitivity : 0.7778
            Specificity : 0.5946
         Pos Pred Value : 0.7656
         Neg Pred Value : 0.6111
             Prevalence : 0.6300
         Detection Rate : 0.4900
   Detection Prevalence : 0.6400
      Balanced Accuracy : 0.6862

       'Positive' Class : 0
```

Figure 8: Confusion Matrix and statistics for Naive Bayes model

6 RESULTS:

Comparing the performance of Logistic Regression, Linear discriminant, Quadratic Discriminant and Naive Bayes models and found that the Logistic Regression performed better among all of them. After, Logistic Regression model with an accuracy level of 77% there comes LDA model with accuracy of 76%, than the Naive Bayes model with 71% accuracy and lastly the QDA model with 70% accuracy. Therefore the can consider the logistic regression model in prediction of diabetes using the coefficients from the output given in fig-4.

7 Model Assessment

In the diagnosis of diabetes, one might be interested in understanding the importance of variables that contribute to diabetes. We can compute that using VarImp fuction.Higher values indicate more importance. These results match up nicely with the p-values from the model. Glucose is the most important predictor variable, followed by BMI, Diabetes Pedigree Function, Blood Pressure, Age, Pregnancies, Insulin.

```
> varImp(model.fit)
                          Overall
Pregnancies             2.0191923
Glucose                 7.5080174
BloodPressure           2.2144317
SkinThickness           0.5230711
Insulin                 0.6525301
BMI                     4.7164578
DiabetesPedigreeFunction 2.9409659
Age                     2.0522711
```

In the presence of multicollinearity, the solution of the regression model becomes unstable. Therefore, VIF values of each variable are estimated using to examine if multicollinearity is a problem in this model. VIF values above 5 indicate the existence of multicollinearity. Since none of the predictor variables in our models have a VIF over 5, it is assumed that multicollinearity is not an issue in the suggested logistic model. [3] [4]

```
> car::vif(model.fit)
            Pregnancies                 Glucose           BloodPressure
               1.525948                1.217609                1.244069
          SkinThickness                 Insulin                     BMI
               1.535913                1.503685                1.289865
DiabetesPedigreeFunction                     Age
               1.053363                1.568127
```

Possible improvement(if any): Considering higher importance of the 'Glucose' variable in prediction of the 'Outcome',I tried logistic regression model again taking Glucose level as a single predictor.

```
#diabetes vs Glucose
ggplot(data = diabetes) +
  aes(x = Outcome, y = Glucose) +
  geom_jitter()
```

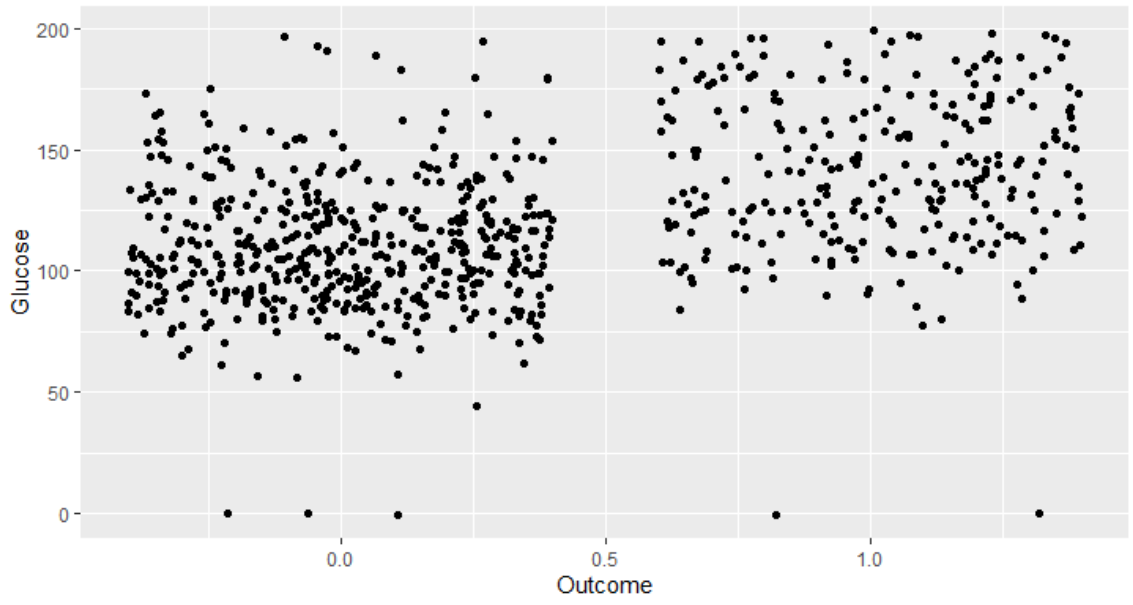

Figure 9: Glucose vs Outcome

[For better visualization I used geom_jitter instead of genom_points under ggplot library. It adds a small amount of random variation to the location of each point, and is a useful way of handling overplotting caused by discreteness.]

Even with high significant importance the Glucose variable as a predictor perform less effective in prediction of diabetes. The model can predict diabetes 73.33% correctly. The confusion matrix and statistics is given in Fig(10).

```
#Glucose as only predictor
logit.fit <- glm(Outcome ~Glucose,
      data = Diabetes_train, family = binomial)

logit.pred <- predict(logit.fit, Diabetes_test,
          type = "response")
predictions <- factor(ifelse(logit.pred > 0.5, 1, 0),
         levels = levels(as.factor(Diabetes_test$Outcome)))
confusionMatrix(predictions, as.factor(Diabetes_test$Outcome))
```

```
        Confusion Matrix and Statistics

                  Reference
        Prediction   0   1
                 0 163  54
                 1  26  57

                      Accuracy : 0.7333
                        95% CI : (0.6795, 0.7825)
           No Information Rate : 0.63
           P-Value [Acc > NIR] : 9.619e-05

                       Kappa : 0.3966

         Mcnemar's Test P-Value : 0.002539

                    Sensitivity : 0.8624
                    Specificity : 0.5135
                 Pos Pred Value : 0.7512
                 Neg Pred Value : 0.6867
                     Prevalence : 0.6300
                 Detection Rate : 0.5433
           Detection Prevalence : 0.7233
              Balanced Accuracy : 0.6880

               'Positive' Class : 0
```

Figure 10: Confusion matrix and statistics taking Glucose as only predictor.

One possible explanation, employing medical terminology, is that high glucose levels, known as hyperglycemia, are characterized by elevated blood sugar levels

commonly associated with type 1 or type 2 diabetes. However, it can also manifest in individuals without these conditions such as significant illness, chronic medical conditions, hormonal disorders, or certain medications.

Thus, solely relying on the variable Glucose for predicting diabetes is inadequate. Effective prediction requires consideration of all relevant variables.

8 CONCLUSIONS:

Logistic regression outperforms discriminant analysis in analyzing categorical response variables due to its adaptability and versatility. Unlike discriminant analysis, logistic regression doesn't assume normality of independent variables. Therefore, comparing performance of logistic regression, LDA, QDA, and Naive Bayes analyses, the logistic regression model emerges as the most effective for diagnosing diabetes. It's also crucial to consider all predictors and their potential interactions for better assessment.

References

[1] Trevor Hastie Robert Tibshirani Gareth James, Daniela Witten. *An introduction to statistical learning : with applications in R.* New York :Springer, (2013).

[2] Zahediasl S. Ghasemi, A. Normality tests for statistical analysis: a guide for non-statisticians. *International journal of endocrinology and metabolism*, 10(2):486–489, 2012.

[3] Leslie O. Schulz, Peter H. Bennett, Eric Ravussin, Judith R. Kidd, Kenneth K. Kidd, Julian Esparza, and Mauro E. Valencia. Effects of Traditional and Western Environments on Prevalence of Type 2 Diabetes in Pima Indians in Mexico and the U.S. *Diabetes Care*, 29(8):1866–1871, 08 2006.

[4] Everhart J. E. Dickson W. C. Knowler W. C. Johannes R. S. (1988). Smith, J. W. Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 261–265.