HarvardX: PH125.9x Data Science

Capstone Project:

Predict Diabetes Positiveness Using LDA Model

April 26, 2019

Philip K W Ng

I. Executive Summary

This project will build a **linear discriminant analysis model** ("LDA model") using the PimaIndiansDiabetes2 dataset to predict the probability of being diabetes positive based on multiple clinical variables.

The following steps are followed to perform the analysis and make the conclusion:

- Download R Package
- Summarize Dataset
- Analyze Dataset
- Build LDA Model
- Make Prediction
- Examine Prediction Accuracy

The PimaIndiansDiabetes2 dataset available in the mlbench package is used for this project. The dataset covers eight clinical variables from 392 female individuals, and is commonly used for binary classification case study, where the outcome variable can have only two possible values: negative or positive.

After summarizing the dataset, "logistic regression" is used to analyze the relationship between the variables. Then, a linear discriminant analysis model ("LDA model") is built to predict the probability of being diabetes positive based on multiple clinical variables. Finally, the "confusion matrix" is used to examine how many observations are correctly or incorrectly classified.

As the LDA model correctly predicts the individual outcome in **86.6%** of the cases, and the misclassification error rate (Type I Error and Type II Error) is low at **13.4%**. Furthermore, both the Sensitivity (True Positive Rate) and the Specificity (True Negative Rate) of the model are high at **75%** and **92.3%**, respectively. Therefore, it is concluded that this LDA model is likely a reliable prediction model for the PimaIndiansDiabetes2 dataset.

Details of the analysis will be explained in the subsequent sections.

II. Download R Package

The following R Packages are downloaded for the analysis in the project: tidyverse, caret, ggplot2, caTools, kLaR, data.table, dplyr, broom, MASS, corrplot.

```
## Loading required package: tidyverse

## -- Attaching packages ---------------------------------------------------------------------------------------------------------------------------
---------- tidyverse 1.2.1 --

## v ggplot2 3.1.1      v purrr   0.3.2
## v tibble  2.1.1      v dplyr   0.8.0.1
## v tidyr   0.8.3      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------------------------------------------------------------------
---- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## Loading required package: caret

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

## Loading required package: caTools

## Loading required package: klaR

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: data.table

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose

## Loading required package: broom

## Loading required package: corrplot

## corrplot 0.84 loaded
```

III. Summarize Dataset

The PimaIndiansDiabetes2 data is downloaded for the analysis in this project.

```
data("PimaIndiansDiabetes2", package = "mlbench")
PimaIndiansDiabetes2 <- na.omit(PimaIndiansDiabetes2)
model <- glm(diabetes ~., data = PimaIndiansDiabetes2,
             family = binomial)
probabilities <- predict(model, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")

mydata <- PimaIndiansDiabetes2 %>%
  dplyr::select_if(is.numeric)
predictors <- colnames(mydata)
```

The following ways are used to look at the raw data from different perspectives: shape, size, type, general layout. Inspecting data helps build up intuition and identify questions for the dataset.

```
dim(mydata)
```

```
## [1] 392   8
```

```
summary(mydata)
```

```
##     pregnant         glucose         pressure         triceps
##  Min.   : 0.000   Min.   : 56.0   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:21.00
##  Median : 2.000   Median :119.0   Median : 70.00   Median :29.00
##  Mean   : 3.301   Mean   :122.6   Mean   : 70.66   Mean   :29.15
##  3rd Qu.: 5.000   3rd Qu.:143.0   3rd Qu.: 78.00   3rd Qu.:37.00
##  Max.   :17.000   Max.   :198.0   Max.   :110.00   Max.   :63.00
##     insulin           mass          pedigree           age
##  Min.   : 14.00   Min.   :18.20   Min.   :0.0850   Min.   :21.00
##  1st Qu.: 76.75   1st Qu.:28.40   1st Qu.:0.2697   1st Qu.:23.00
##  Median :125.50   Median :33.20   Median :0.4495   Median :27.00
##  Mean   :156.06   Mean   :33.09   Mean   :0.5230   Mean   :30.86
##  3rd Qu.:190.00   3rd Qu.:37.10   3rd Qu.:0.6870   3rd Qu.:36.00
## Max.  :846.00  Max.  :67.10  Max.  :2.4200  Max.  :81.00
```

```
str(mydata)
```

```
## 'data.frame':    392  obs. of  8 variables:
##  $ pregnant:    num  1 0 3 2 1 5 0 1 1 3 ...
```

```
##  $ glucose :     num  89 137 78 197 189 166 118 103 115 126 ...
##  $ pressure:     num  66 40 50 70 60 72 84 30 70 88 ...
##  $ triceps :     num  23 35 32 45 23 19 47 38 30 41 ...
##  $ insulin :     num  94 168 88 543 846 175 230 83 96 235 ...
##  $ mass    :     num  28.1 43.1 31 30.5 30.1 25.8 45.8 43.3 34.6 39.3 ...
##  $ pedigree:     num  0.167 2.288 0.248 0.158 0.398 ...
##  $ age     :     num  21 33 26 53 59 51 31 33 32 27 ...
##  - attr(*, "na.action")  = 'omit' Named int  1 2 3 6 8 10 11 12 13 16 ...
##   ..- attr(*, "names")   = chr  "1" "2" "3" "6" ...
```

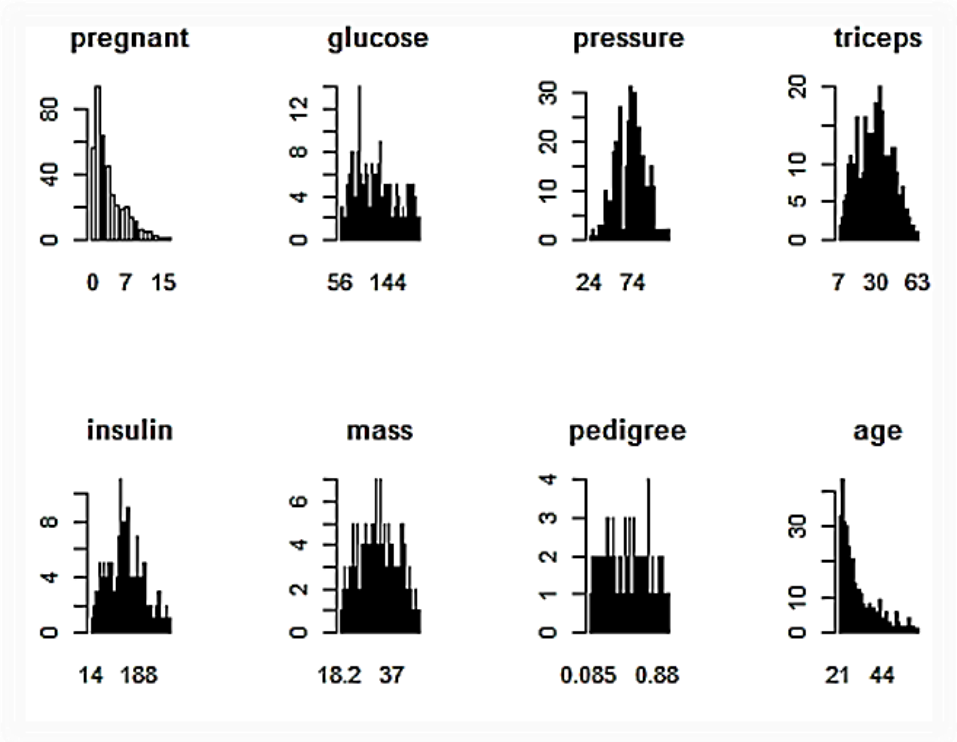
IV. Analyze Dataset

Data visualization is perhaps the fastest and most useful way to learn more about and summarize the data.

As the dataset outcome is a binary, that is, either diabetes positive or diabetes negative, logistic regression is used to help visualize the relationship between the variables and the logit of the outcome.

The barplots below give an idea of the proportion of instances that belong to each category.

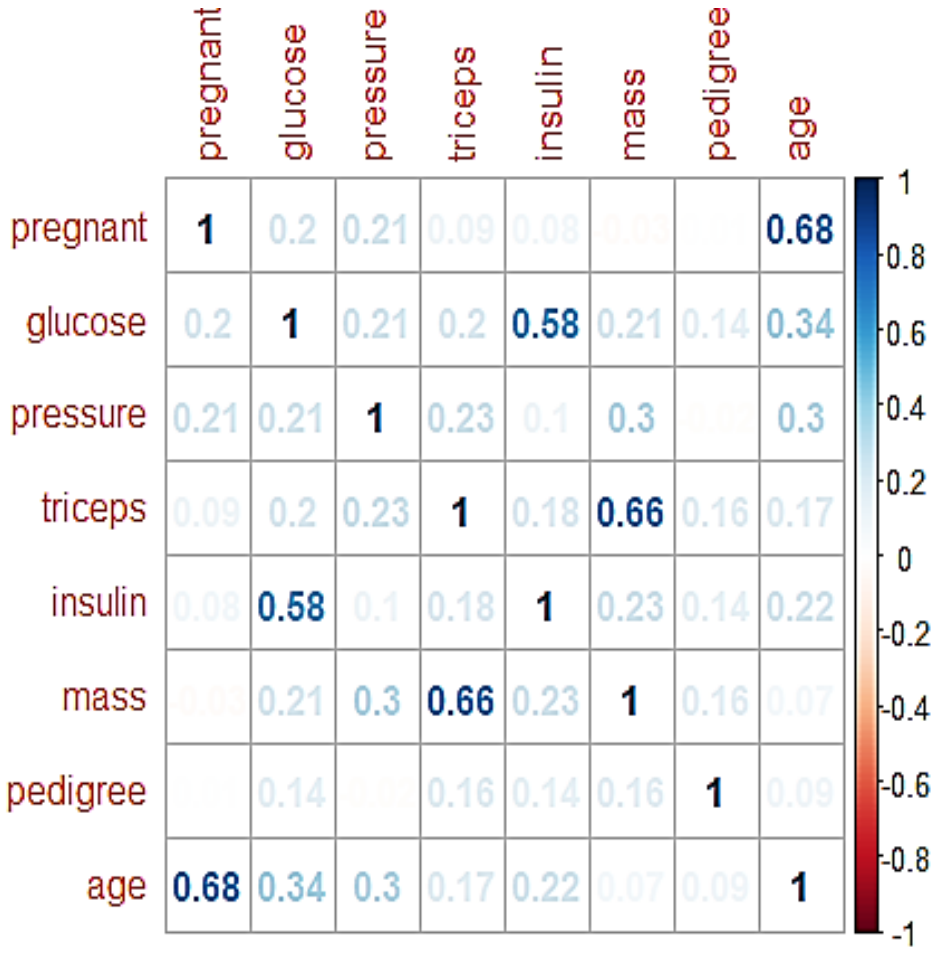
```
par(mfrow=c(2,4))
for(i in 1:8) {
  counts <- table(mydata[,i])
  name <- names(mydata)[i]
  barplot(counts, main=name)
}
```



The correlation plot below shows that that the following attributes tend to change together:

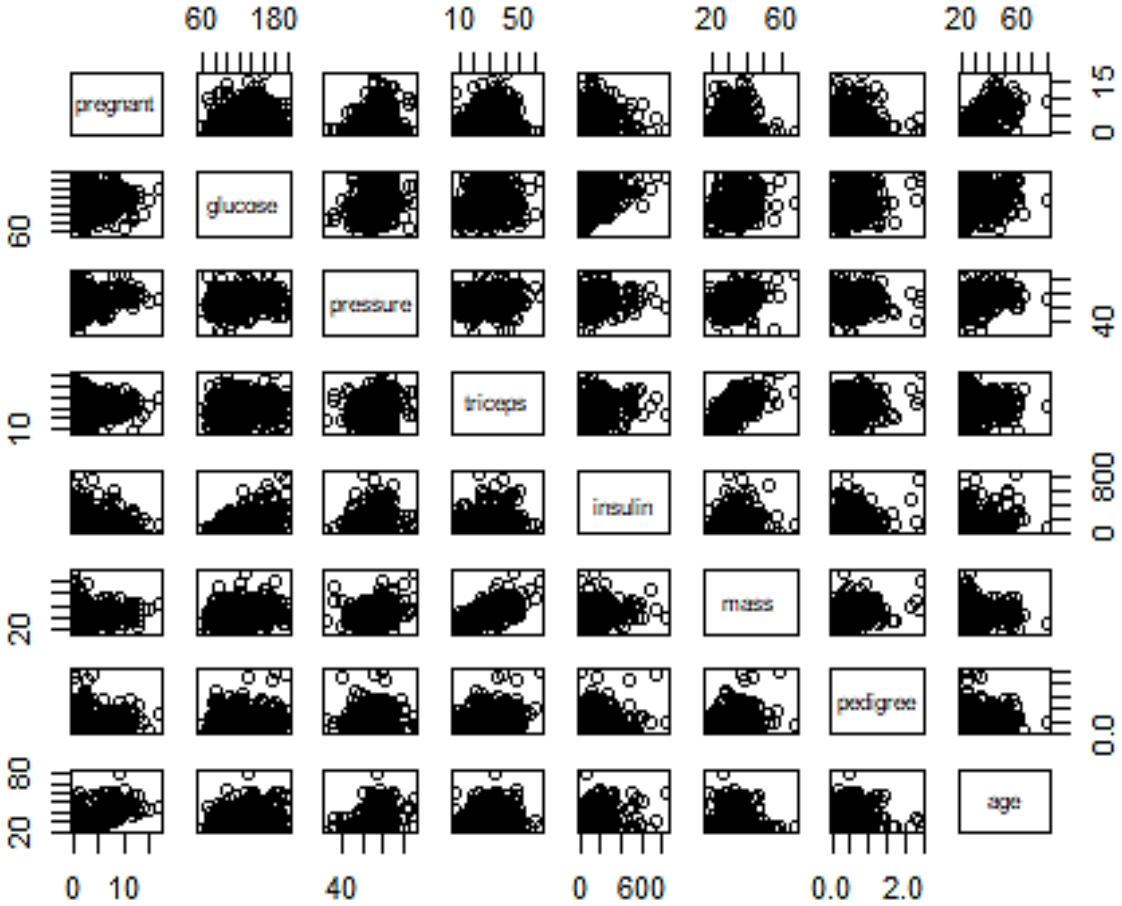
- pregnant and age
- glucose and insulin
- tricepes and mass

```
correlations <- cor(mydata[,1:8])
corrplot(correlations, method="number")
```



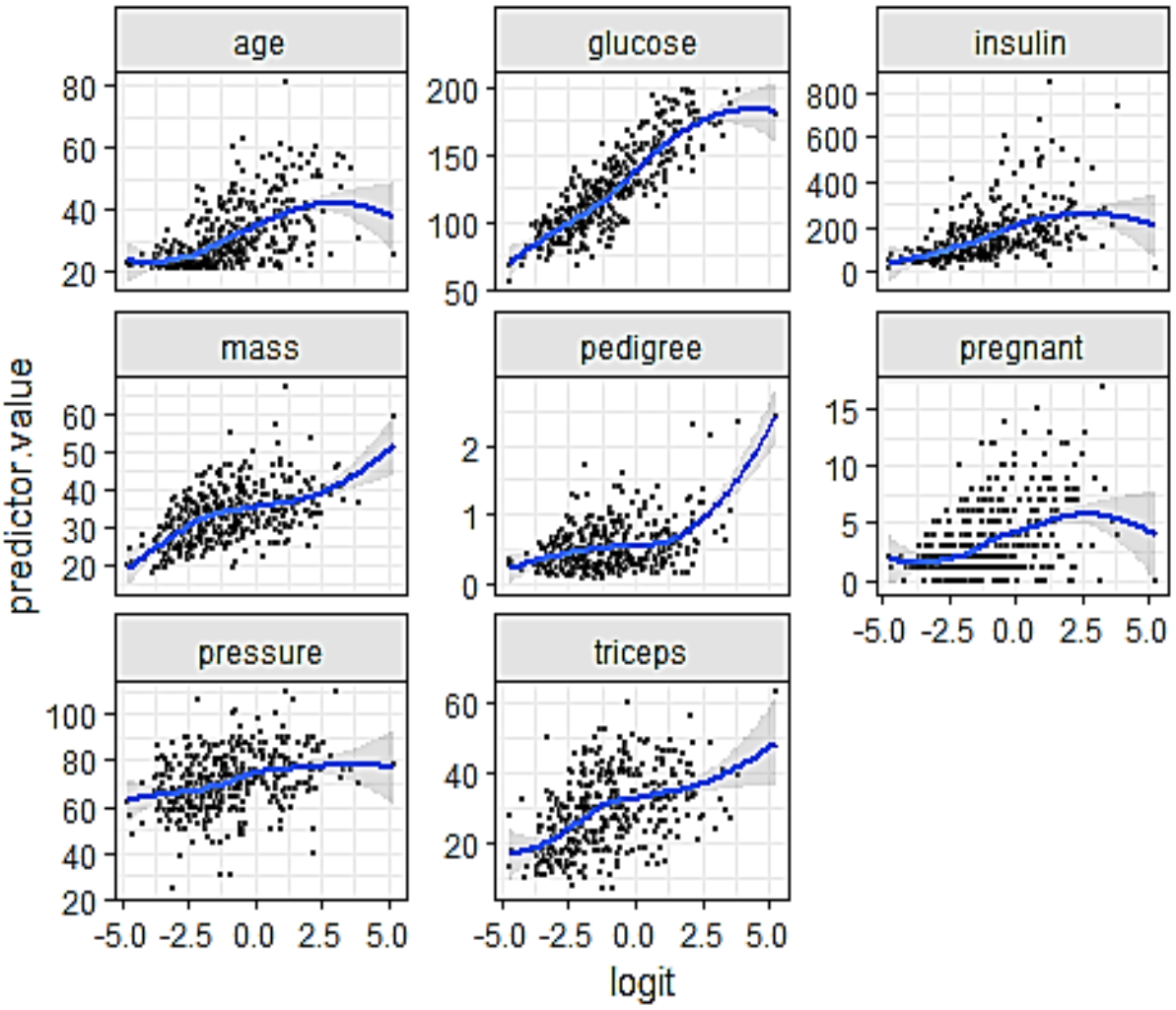
The scatter plot matrix below indicates the relationship between the variables. This aids in looking at the data from multiple perspectives.

```
#Scatter plot
pairs(mydata)
```



The scatter plot is then smoothed to show clearer relationship between each variable and the logit values.

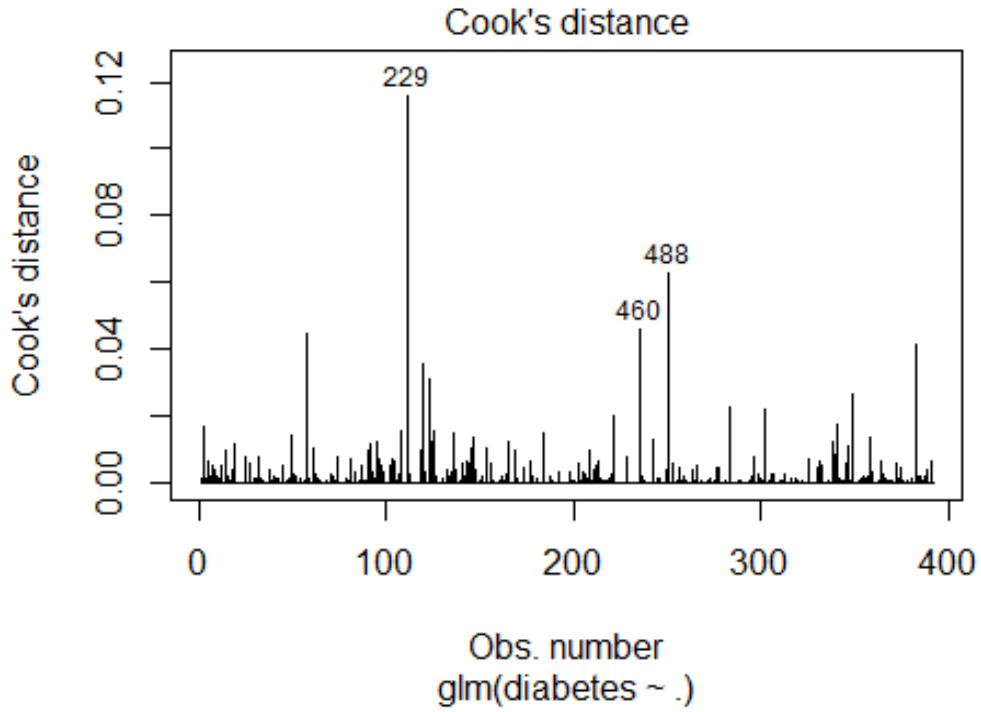
```
mydata <- mydata %>%
  mutate(logit = log(probabilities/(1-probabilities))) %>%
  gather(key = "predictors", value = "predictor.value", -logit)
ggplot(mydata, aes(logit, predictor.value))+
  geom_point(size = 0.5, alpha = 0.5) +
  geom_smooth(method = "loess") +
  theme_bw() +
  facet_wrap(~predictors, scales = "free_y")
```



The above smoothed scatter plots show that variables glucose, mass, pregnant, pressure and triceps are all quite linearly associated with the diabetes outcome in logit scale.

Cook's distance is used to examine the extreme values (outliners) in the data. Below is the identified top 3 outliners.

```
plot(model, which = 4, id.n = 3)
```



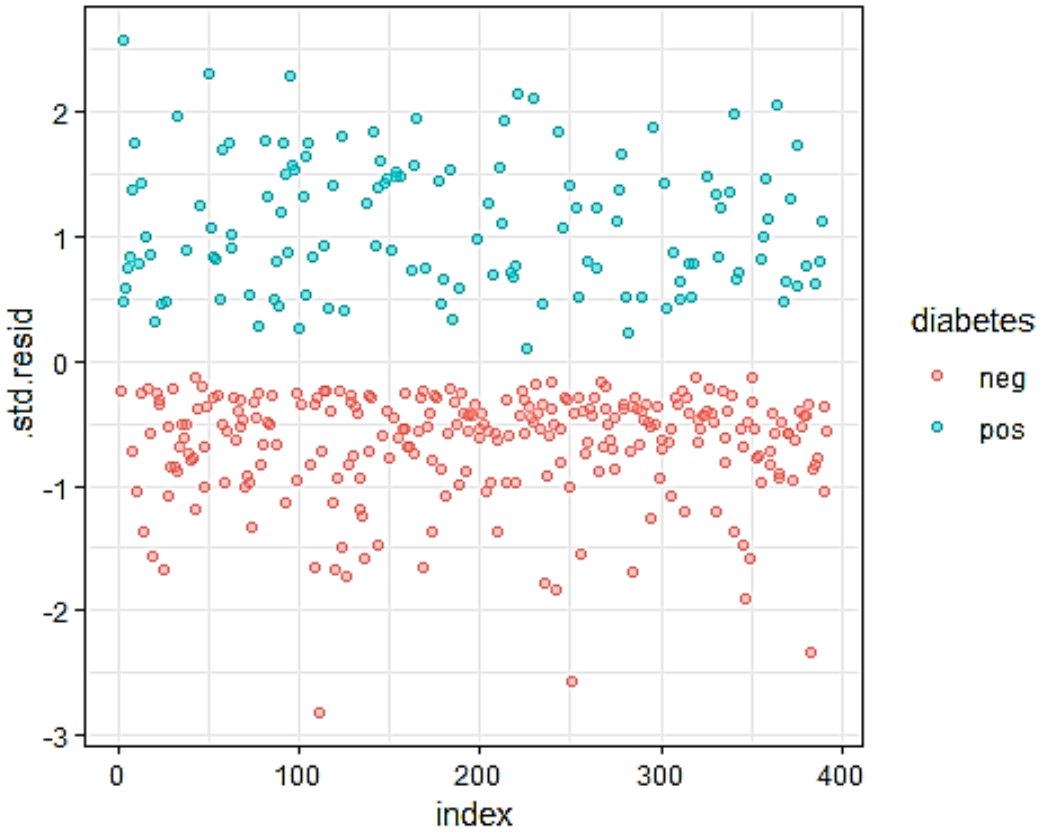
Further details of the top 3 outliners are shown as follow:

```
model.data <- augment(model) %>%
  mutate(index = 1:n())
model.data %>% top_n(3, .cooksd)

## # A tibble: 3 x 18
##   .rownames diabetes pregnant glucose pressure triceps insulin  mass
##         <chr>  <fct>    <dbl>       <dbl>  <dbl> <dbl>  <dbl> <dbl>
## 1       229     neg        4         197     70    39    744    36.7
## 2       460     neg        9         134     74    33     60    25.9
## 3       488     neg        0         173     78    32    265    46.5
## # ... with 10 more variables: pedigree <dbl>, age <dbl>, .fitted <dbl>,
## #   .se.fit <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,
## #   .std.resid <dbl>, index <int>
```

Although outliners may impact the quality of the logistic regression analysis, not all of them are influential observations. To check if the data contains potential influential observations, the standardized error of residuals is inspected. The standardized residuals are plotted as below.

```
ggplot(model.data, aes(index, .std.resid)) +
  geom_point(aes(color = diabetes), alpha = .5) +
  theme_bw()
```



The filter below is used to identify if there are any influential data points with abs (.std.res) > 3.

```
model.data %>%
  filter(abs(.std.resid) > 3)

## # A tibble: 0 x 18
## # ... with 18 variables: .rownames <chr>, diabetes <fct>, pregnant <dbl>,
## #   glucose <dbl>, pressure <dbl>, triceps <dbl>, insulin <dbl>,
## #   mass <dbl>, pedigree <dbl>, age <dbl>, .fitted <dbl>, .se.fit <dbl>,
## #   .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,
## #   .std.resid <dbl>, index <int>
```

The above analysis indicates that there is no influential observation in the data.

Multicollinearity corresponds to a situation where the data contains highly correlated predictor variables and should be removed in regression analysis. The R function *vif()* is used to identiy such situation, and a value that exceeds 5 indicates a problematic amount of collinearity.

```
car::vif(model)

## pregnant     glucose     pressure     triceps     insulin     mass     pedigree     age
## 1.892387     1.378937    1.191287     1.638865    1.384038    1.832416 1.031715     1.974053
```

As all variables show a VIF value of well below 5, there is no collinearity.

V. Build LDA Model

The linear discriminant analysis model is used to predict the probability of diabetes test positively based on clinical variables.

The PimaIndiansDiabetes2 dataset is split into training set (75% used to build the model) and test set (25% used to evaluate the model performance).

```
pima.data <- na.omit(PimaIndiansDiabetes2)
# Inspect the data
# Split the data into training and test set
set.seed(123)
training.samples <- pima.data$diabetes %>%
  createDataPartition(p=0.75, list=FALSE)
train.data <- pima.data[training.samples, ]
test.data <- pima.data[-training.samples, ]
```

VI. Make Prediction

The LDA model is fitted on the training set and make predictions on the test set.

```
# Fit LDA
fit <- lda(diabetes ~., data = train.data)
# Make predictions on the test data
predictions <- predict(fit, test.data)
prediction.probabilities <- predictions$posterior[,2]
predicted.classes <- predictions$class
observed.classes <- test.data$diabetes

accuracy <- mean(observed.classes == predicted.classes)
accuracy

## [1] 0.8659794

error <- mean(observed.classes != predicted.classes)
error

## [1] 0.1340206
```

From the output above, the LDA Model correctly predicted the individual outcome in 86.6% of the cases, whereas the misclassification error rate (Type I Error and Type II Error) is low at 13.4%.

VII. Examine Prediction Accuracy

Two metrics are used to examine the performance of the LDA Model:

Sensitivity – which is the True Positive Rate or the proportion of identified positives among the diabetes-positive population.

Specificity – which is the True Negative Rate or the proportion of identified negatives among the diabetes-negative population.

Sensitivity and *Specificity* are computed using the function confusionMatrix().

```
# Confusion matrix, proportion of cases
table(observed.classes, predicted.classes) %>%
  prop.table() %>% round(digits = 3)

## predicted.classes
## observed.classes   neg   pos
##              neg   0.619 0.052
##              pos   0.082 0.247

confusionMatrix(predicted.classes, observed.classes,
                positive = "pos")

## Confusion Matrix and Statistics
##          Reference
## Prediction neg pos
##       neg  60   8
##       pos   5  24
##
##             Accuracy : 0.866
##               95% CI : (0.7817, 0.9267)
##    No Information Rate : 0.6701
##    P-Value [Acc > NIR] : 9.078e-06
##
##                  Kappa : 0.6895
##
##  Mcnemar's Test P-Value : 0.5791
##
##             Sensitivity : 0.7500
##             Specificity : 0.9231
##          Pos Pred Value : 0.8276
##          Neg Pred Value : 0.8824
##              Prevalence : 0.3299
##          Detection Rate : 0.2474
##    Detection Prevalence : 0.2990
##       Balanced Accuracy : 0.8365
##
##       'Positive' Class : pos
```

From the output above, the Sensitivity is high at 75%, that is, 75% of diabetes-positive individuals are correctly identified by the model as diabetes-positive. On the other hand, the Specificity is also high at 92.3%, that is, 92.3% of diabetes-negative individuals are correctly identified by the model as diabetes-negative.

VIII. Conclusion

As the LDA model correctly predicts the individual outcome in **86.6%** of the cases, and the misclassification error rate (Type I Error and Type II Error) is low at **13.4%**. Furthermore, both the Sensitivity (True Positive Rate) and the Specificity (True Negative Rate) of the model are high at **75%** and **92.3%**, respectively. Therefore, it is concluded that this LDA model is likely a reliable prediction model for the PimaIndiansDiabetes2 dataset.