Diabetes Accuracy Write Up

By: Ravinit Chand, Juan Diaz, Sai Sindura Vuppu, and Brandon Vargas

Background

Diabetes is rapidly increasing worldwide, leading to serious health issues like heart disease and kidney failure. It is crucial to study and understand the patterns related to this disease because early diagnosis and effective management can significantly improve patient lives and reduce healthcare costs. The Pima Indians Diabetes Dataset is a prominent dataset utilized for predicting the onset of diabetes in female patients based on various diagnostic measurements. Originating from the National Institute of Diabetes and Digestive and Kidney Diseases, this dataset comprises data from 768 female patients of Pima Indian heritage, aged 21 years and older. It includes eight exploratory variables: the number of pregnancies, plasma glucose concentration, blood pressure, triceps skin fold thickness, 2-hour serum insulin, body mass index (BMI), diabetes pedigree function, and age. The target variable is a binary outcome indicating whether the patient has diabetes (pos) or not (neg).

As we attempt to explore this dataset, we pose the following questions which will help drive our research: Which features in the dataset have the highest correlation with each other? Do the results from our model provide sufficient accuracy in predicting diabetes? If the model's accuracy is insufficient, does dimensionality reduction improve the accuracy of diabetes predictions? Our primary focus is to get a relatively high accuracy in predicting diabetes in a patient. In order to achieve this, we experiment with a method that reduces the dimensionality of the data called PCA (Principal Component Analysis) to check if this will lead to a higher accuracy than the original data.

Exploratory Data Analysis: Analyzing The Data

Pair Plots

Model 1 is a pair plots visualization which uses the relationships between every possible pair of explanatory variables to show us different insights into the variables' distribution and correlation. Each pair of variables is plotted against each other in the different squares in the plot with the scattered points colored based on diabetes status (green being positive and red being negative). The plots on the diagonal show the distribution of each variable, the scatter plots under the diagonal show whether the relationships are linear, and the plots above the diagonal are used to show the correlation coefficients to back up the scatter plots.

Correlation Matrix

The correlation plot (Model 2) goes further into showing the correlation coefficients between all of the pairs of variables. The color and value are indications of the strength and direction of the correlations. As an example, there is a weak negative correlation between age and the triceps measurement (-0.11) which shows that as women get older in the database, their tricep skinfold thickness decreases, but the decrease is very miniscule. The purpose of this plot is to explain the relationship and magnitude between the explanatory variables.

Logistic Regression Table

The logistic regression model (Model 3) is shown to provide the coefficients for each explanatory variable. These coefficients indicate the variables' strength and direction of association with diabetes status. The significant coefficients shown are glucose, BMI, and the number of pregnancies and all have small p-values which show their strong significance. For example, the coefficient of pedigree is 0.9452 which means that for each unit increase in pedigree, the log odds of having diabetes increase by 0.9452 while holding all other variables constant. This table is vital for understanding which variables are the strongest predictors of diabetes in the dataset.

Analyzing The Dataset

Part 1: Data Cleaning

When originally looking at our dataset containing eight explanatory variables and the binary response variable, we first wanted to check to see if there were any N/A values, which would've led us to delete the values or replace them with the mean values. N/A values in the context of this dataset is when one of the 768 female patients didn't have information about the explanatory or response variable, such as not inputting the age of a patient for example. There were no N/A values within the entire dataset, so no changes were made.

Part 2: Box-M Test For Original Data

We then performed the Box-M test, a technique used to determine if the covariance matrices for groups in a multivariate setting are the same. This test differentiates between Quadratic Discriminant Analysis (QDA) and Linear Discriminant Analysis (LDA). QDA assumes that covariance matrices are not equal, while LDA assumes they are equal. The test resulted in a p-value of 2.2e-16, which is significantly smaller than any common significance level, suggesting the use of QDA over LDA.

Part 3: Shapiro-Wilks Test

For further testing on our dataset, we also did shapiro-wilks test on all eight explanatory variables with and without diabetes, outputting sixteen values. The shapiro-wilks test is used to verify if the assumption of normality is utilized in our dataset, which occurs when the p-value is greater than the significance level. However, we got significantly small p-values for all variables, so we concluded that normality assumption is violated. Despite this violation, quadratic discriminant analysis (QDA) is still considered the optimal choice over linear discriminant analysis (LDA) for classification tasks in this context because QDA does not assume equal covariance matrices between the groups, making it more robust to such deviations from normality.

Part 4: Accuracy With QDA And CV Using Full Data

To see how correct our research was in hopes of assisting citizens worried about the prevalence of diabetes, we ran an accuracy table using QDA. We predicted the data on both the test and training with our QDA model to see how accurate our results were. This led to a 70% accuracy on the test set and 78% accuracy on the training set, which was decent but we knew as researchers that having the highest accuracy possible was the ultimate goal.

With that, we also used a strong method in statistics called cross-validation with ten folds. Here, we take our full data and have ten equal portions of the data, where nine of the ten is

the training set and the last set is the test set, which will lead to an accuracy. This occurs ten times, with each set eventually becoming the test set, and the average of those ten accuracies is the accuracy from the cross validation. Here, the accuracy was 74.6%, indicating the model's reliability but also highlighting room for improvement.

The decision boundary was utilized to assign which class our data fell into. The decision boundary is modeled by the equation $\frac{1}{2}x^T Ax + x^T b + c$ where the matrix A, vector b, and constant c are calculated using the sample correlation matrices and the prior probabilities of the positive and negative subjects. If the new data (x) plugged into the boundary data was greater than 0, that new data would get assigned to class 1, labeled as "negative" diabetes. However, if the new data plugged into the decision boundary data is less than 0, the new data would get assigned to class 2 which is "positive" diabetes.

Part 5: PCA

In hopes of having a higher accuracy, we decided to do principal component analysis (PCA), which reduces the dimensions of the dataset but keeps most of the information since PCA's are a linear combination of the explanatory variables. When getting our PCA's, we did the eigenvalue test which is where you keep the PCA's that have eigenvalues greater than one into the new dataset. We kept the first three PCA's from that test and added the binary diabetes response variable from the original dataset, creating a new dataset with four columns.

Part 6: Box-M Test For Dimension-Reduced Data

We ran the Box M test again for the same reason as previously mentioned, which like before, led us to having a smaller p-value than the significance level, implying that QDA is a better fit in the model than LDA.

Part 7: Accuracy With QDA And CV With Dimension-Reduced Data

With our new model with reduced dimensionality, we wanted to see if our accuracy was higher in hopes of helping individuals have more accurate answers over if they have diabetes or not given data about them. We ran the same three tests that gave us accuracy as the model with the original dataset, which was the QDA test from the training set, QDA test from the testing set, and the cross-validation from the whole dataset. To our surprise, our accuracy didn't change much, with the QDA model with the test set from the PCA dataset having an accuracy of 70.67%, while the training set 's accuracy was at 75%. Finally the cross validation PCA dataset model has an accuracy of 73.9%. Overall the accuracy didn't change significantly with reduced dimensionality which is shown by Model 4, which we found surprising since we predicted that having reduced dimensionality would increase the accuracy of the dataset by a good amount.

Part 8: Interpreting PCA Models

The PCA plots show context on how well the principal components differentiate between diabetic and non-diabetic individuals. Model 5 shows that the first two principal components capture a significant portion of the variance, with visible clusters but some overlap, indicating moderate separation of classes. Model 6 captures less variance, with more overlap between classes, suggesting less effective separation. Model 7 demonstrates better separation than Model 6, reinforcing that PC1 is a strong differentiator. Overall, while PCA aids in dimensionality reduction, the separation is not perfect, highlighting the inherent complexity in the dataset.

Strength And Weaknesses

There are many strengths to our analysis. First, the absence of N/A values in the dataset meant the data was already clean and ready to work with. Having no N/A values is a strength since a problem that could arise from having a good amount of N/A values is if removing them will skew the data, which we fortunately didn't have to think about with no N/A values in the dataset. In terms of our process, we used the Box M Test twice: first with the original dataset then with the dataset with three principal components. The fact that the test returned a low p-value for both datasets shows that QDA is the appropriate classification method to use. In our principal component analysis, we were successful in reducing the dataset to three principal components compared to the original eight predictors. Finally, using the scree plot and elbow point method provides an advantage over other methods like the overall threshold and hypothesis testing approaches.

There are also some limitations to our study design. First, PCA is traditionally used for ranking, but no kind of sensible analysis would rank this kind of data. Given the response variable of subjects having or not having diabetes, it makes more sense to classify the data. However, we do use PCA to reduce the dimensionality, which would help if given a larger dataset. Finally, and probably the most impactful limitation, we cannot interpret the principal components practically. Since principal components are linear combinations of predictor variables, it's hard to say which of the predictor variables have more of an effect on classifying diabetes than do the others.

Conclusion

In conclusion, upon studying the Pima Indians Diabetes Dataset, we assessed how accurate our classification model was with and without reducing the dimensionality of the dataset. Our process consisted of training a QDA model, reducing the dimensions of the data using PCA, then training another QDA model on the reduced data. We also used methods like the Box-M test, Shapiro-Wilks test, and Cross Validation to ensure our model met assumptions along the way. We created a model with about 74% accuracy for the original data, and a model with about 73% accuracy for the PCA-reduced data.

These results reveal two main findings that answer our proposed research questions: First, our model does not lead to sufficient accuracy. An accurate model would likely achieve at least 80% accuracy. Second, using PCA did not change our classification accuracy much, which is illustrated by Model 4 in the appendix, with the accuracies before and after QDA having relatively similar numbers. Although PCA is a useful method for larger datasets since it reduces the dimensions, it did not prove successful on this dataset. We also proposed the question of correlation amongst variables, and this question is explored in our visualizations. For example the variables age and pregnant have a correlation of 0.54, which is a moderate positive correlation, implying that when age increases for the women in the dataset, pregnancy also typically increases.

The main practical implication of our analysis is to use it to determine whether or not someone has diabetes. For example, if someone were to come into a clinic wondering about whether or not they have diabetes, a doctor may take measurements such as glucose levels and triceps size. The doctor would then plug these measurements into our model to classify whether or not this person has or will have diabetes.

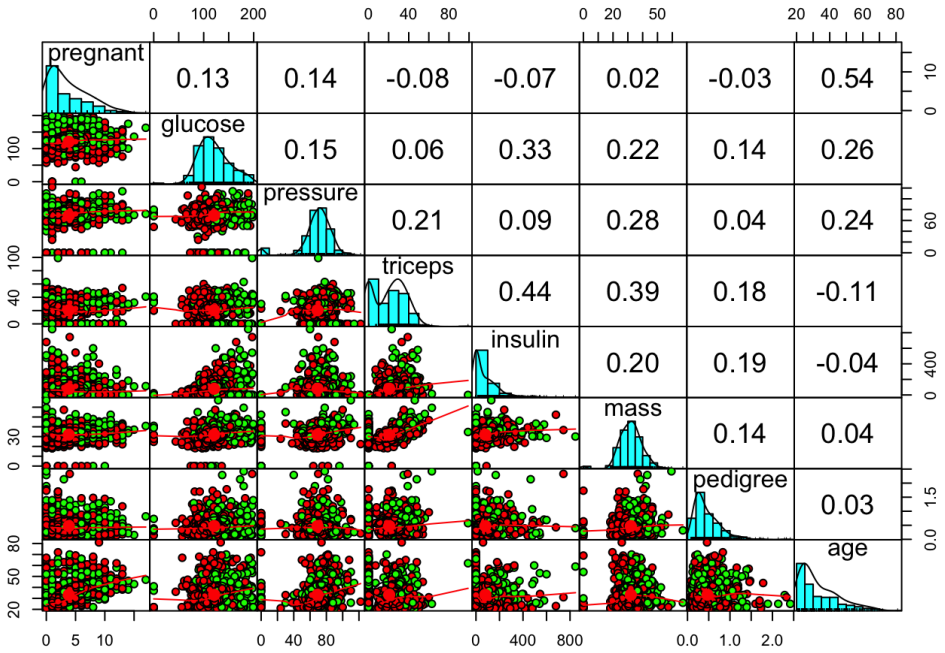
Group Member Contributions

- Ravinit Chand:
    - Write-up for analyzing the dataset parts 1, 4, 5, 6, and 7
    - Helped with write-up for strength/weakness, conclusion, and introduction
    - Did parts 1-7, 9-11 of code
- Juan Diaz
    - Write-up for exploratory data analysis
    - Write-up for analyzing the dataset part 8
    - Did parts 12-16 of code
- Sai Sindura Vuppu:
    - Write-up for introduction/background
    - Write-up for analyzing the dataset parts 2 and 3
    - Helped with plot interpretation and write-up for EDA
    - Helped with part 6 of the code
    - Refined the code
- Brandon Vargas:
    - Write-up for strength/weaknesses
    - Write-up for conclusion
    - Computed decision boundary calculation and write-up
    - Refined the code

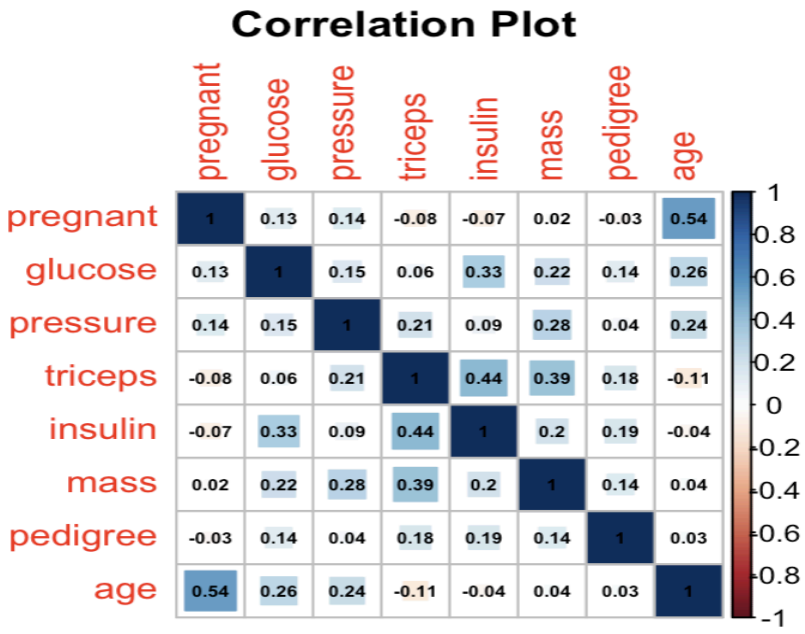
Dataset Used

https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

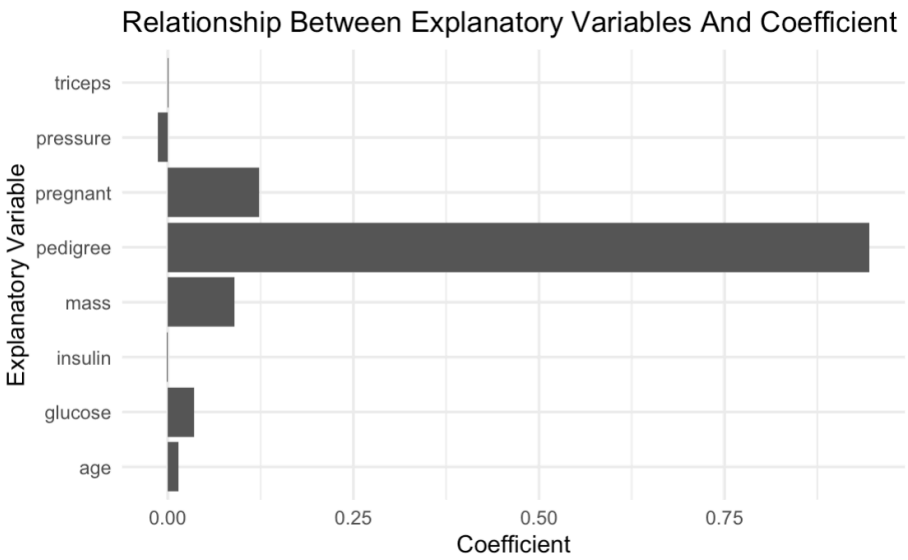
Appendix



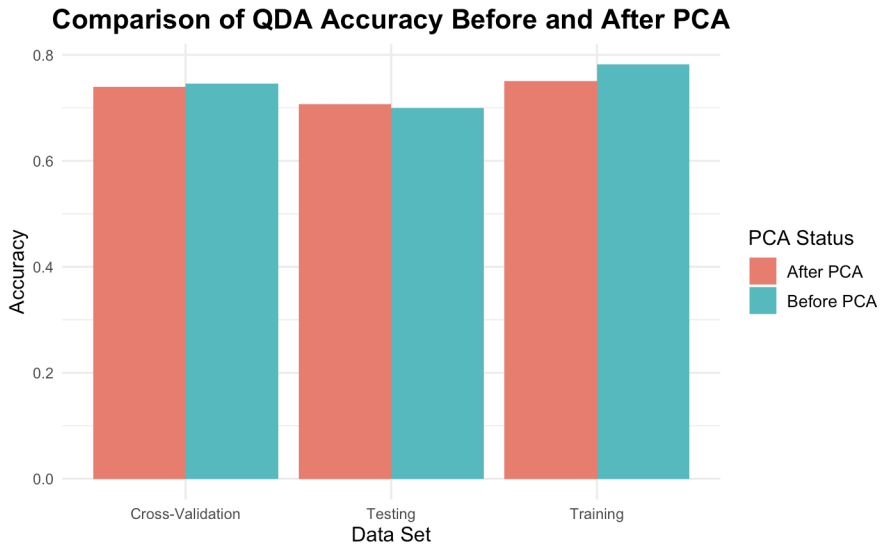
Model 1: Pair Plot



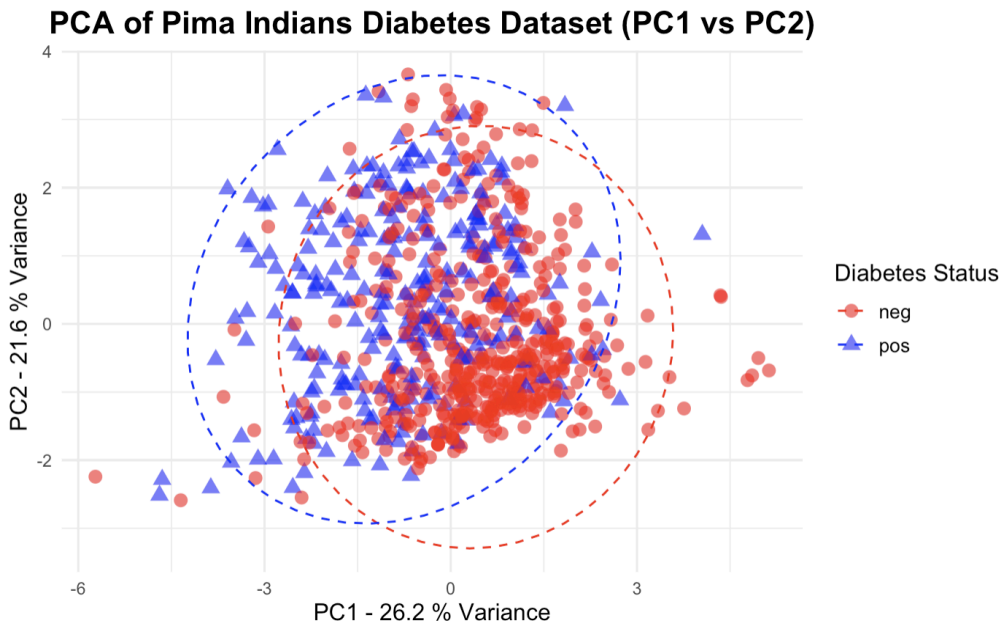
Model 2: Correlation Plot



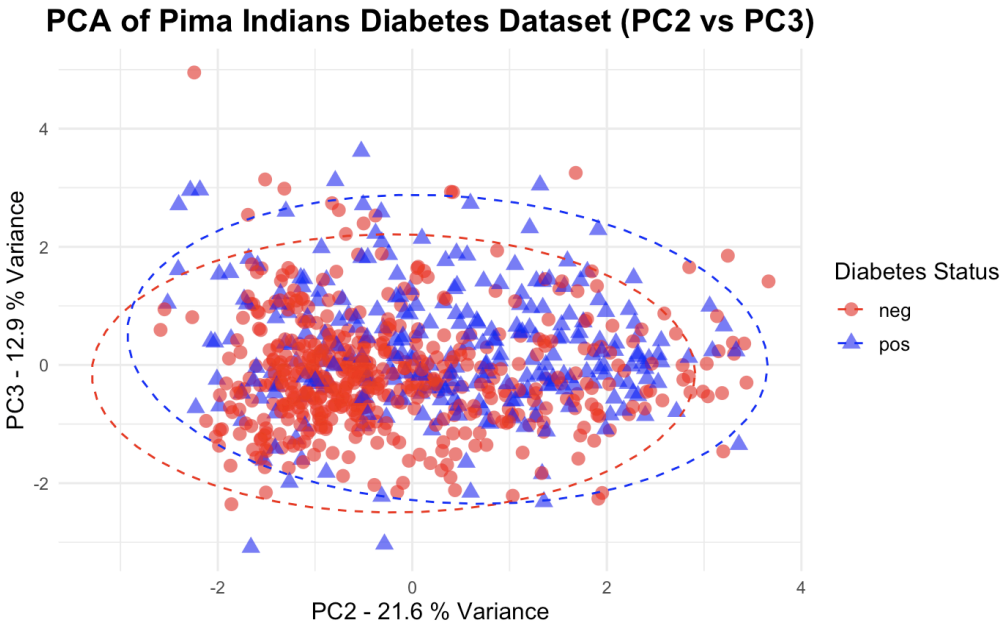
Model 3: Logistic Regression Model



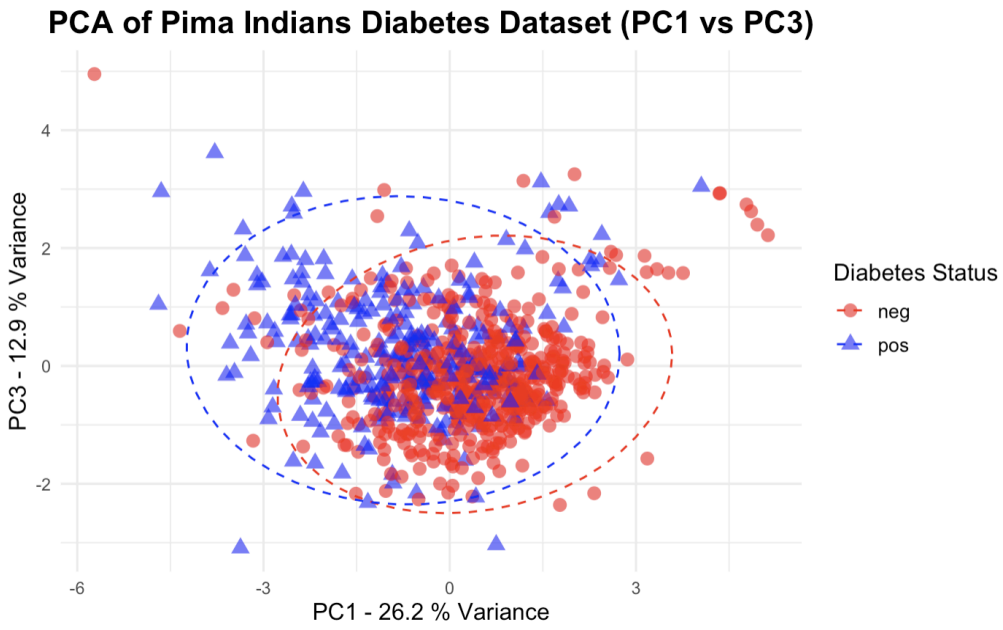
Model 4: Comparing Accuracy Before/After PCA



Model 5: PC1 Vs PC2 Comparison



Model 6: PC2 Vs PC3 Comparison



Model 7: PC1 Vs PC3 Comparison

Code

```
```{r}
# Importing Necessary Libraries
library(mvtnorm)
library(klaR)
library(psych)
library(MASS)
library(devtools)
library(heplots)
library(mlbench)
library(factoextra)
library(caret)
library(reshape2)
library(ggplot2)
library(ggfortify)
```

```
# Visualizing The Dataset (Part 1)
data(PimaIndiansDiabetes)
str(PimaIndiansDiabetes)
pairs.panels(PimaIndiansDiabetes[1:8],
             gap = 0,
             bg = c("red", "green", "blue", "pink", "yellow", "orange",
"brown", "black")[PimaIndiansDiabetes$diabetes],
             pch = 21)
```

```
# Check NA Values (Part 2)
total_na <- sum(is.na(PimaIndiansDiabetes))
print(total_na)
```

```
# LDA And QDA Classification (Part 3)
res <- boxM(PimaIndiansDiabetes[, 1:8], PimaIndiansDiabetes[, "diabetes"])
summary(res)
boxM(cbind(pregnant, glucose, pressure, triceps, insulin, mass, pedigree,
age) ~ diabetes, data=PimaIndiansDiabetes)
```
```

```r
# Form Test/Training Sets (Part 4)
set.seed(123)
ind <- sample(2, nrow(PimaIndiansDiabetes), replace = TRUE, prob = c(0.6,
0.4))
training <- PimaIndiansDiabetes[ind == 1, ]
testing <- PimaIndiansDiabetes[ind == 2, ]
dim(PimaIndiansDiabetes)
dim(training)
dim(testing)
```

```r
# Shapiro-Wilks Test For Normality (Part 5)
shapiro_test_results <- lapply(names(training)[1:8], function(var) {
  diabetes_neg <- shapiro.test(training[training$diabetes == "neg",
var])$p.value
  diabetes_pos <- shapiro.test(training[training$diabetes == "pos",
var])$p.value
  return(c(diabetes_neg, diabetes_pos))
})
shapiro_results_df <- do.call(rbind, shapiro_test_results)
colnames(shapiro_results_df) <- c("diabetes_neg_p_value",
"diabetes_pos_p_value")
shapiro_results_df$variable <- names(training)[1:8]
print(shapiro_results_df)
```

```r
# QDA Model And Accuracy (Part 6)
quadratic <- qda(diabetes ~ ., data = training)
p1_qda <- predict(quadratic, training)$class
tab_qda <- table(Predicted = p1_qda, Actual = training$diabetes)
p2_qda <- predict(quadratic, testing)$class
tab1_qda <- table(Predicted = p2_qda, Actual = testing$diabetes)
accuracy_train_before <- sum(diag(tab_qda)) / sum(tab_qda)
accuracy_test_before <- sum(diag(tab1_qda)) / sum(tab1_qda)
print(paste("Training Accuracy:", accuracy_train_before))
print(paste("Testing Accuracy:", accuracy_test_before))
```

```r
# CV For OG Data (Part 7)
num_folds <- 10
```

```r
ctrl <- trainControl(method = "cv", number = num_folds)
model_before <- train(diabetes ~ ., data = training, method = "qda",
trControl = ctrl)
print(model_before)
cv_accuracy_before <- model_before$results$Accuracy
```

```r
# Decision Boundary For QDA (Part 8)
pi1 <- quadratic$prior[[1]]
pi2 <- quadratic$prior[[2]]
pos <- training[training$diabetes == "pos", ][, 1:8]
neg <- training[training$diabetes == "neg", ][, 1:8]
mu1 <- matrix(unname(colMeans(neg)))
mu2 <- matrix(unname(colMeans(pos)))
d <- dim(mu2)
S1 <- cor(neg)
S2 <- cor(pos)
n1 <- dim(neg)[1]
n2 <- dim(pos)[1]
Spl <- ((n1 - 1) * S1 + (n2 - 1) * S2) / (n1 + n2 - 2)
A <- solve(S2) - solve(S1)
b <- solve(S1) %*% mu1 - solve(S2) %*% mu2
c <- log(pi1/pi2) + 1/2 * (t(mu2) %*% solve(S2) %*% mu2 - t(mu1) %*%
solve(S1) %*% mu1) + 1/2 * log(det(S2)/det(S1))
```

```r
# Correlation Plot (Part 9)
library(corrplot)
correlation_matrix <- cor(PimaIndiansDiabetes[, 1:8])
corrplot(correlation_matrix, method = "square", addCoef.col = "black",
number.cex = 0.5, title = "Correlation Plot", mar = c(0, 0, 2, 0))
```

```r
# Logistic Regression Table (Part 10)
model <- glm(diabetes ~ ., data = PimaIndiansDiabetes, family = binomial)
coef_table <- coef(summary(model))
print(coef_table)
dataframe_coefficients <- data.frame(Explanatory_Variable =
names(coef(model))[-1], Coefficient = coef(model)[-1])
print(dataframe_coefficients)
```

```r
# PCA (Part 11)
res.pca <- prcomp(PimaIndiansDiabetes[, 1:8], scale = TRUE)
principal_components <- res.pca$x[, 1:3]
response_variable <- as.factor(PimaIndiansDiabetes$diabetes)
New_PimaIndiansDiabetes <- data.frame(principal_components, diabetes =
response_variable)
```

```r
# PCA Plots (Part 12)
pca_plot_1_2 <- ggplot(New_PimaIndiansDiabetes, aes(x = PC1, y = PC2, color
= diabetes, shape = diabetes)) +
  geom_point(size = 3, alpha = 0.6) +
  stat_ellipse(type = "norm", linetype = 2, level = 0.95) +
  scale_color_manual(values = c("neg" = "red", "pos" = "blue")) +
  scale_shape_manual(values = c("neg" = 16, "pos" = 17)) +
  labs(title = "PCA of Pima Indians Diabetes Dataset (PC1 vs PC2)",
       x = paste("PC1 -", round(summary(res.pca)$importance[2, 1] * 100,
1), "% Variance"),
       y = paste("PC2 -", round(summary(res.pca)$importance[2, 2] * 100,
1), "% Variance"),
       color = "Diabetes Status", shape = "Diabetes Status") +
  theme_minimal()
print(pca_plot_1_2)

pca_plot_1_3 <- ggplot(New_PimaIndiansDiabetes, aes(x = PC1, y = PC3, color
= diabetes, shape = diabetes)) +
  geom_point(size = 3, alpha = 0.6) +
  stat_ellipse(type = "norm", linetype = 2, level = 0.95) +
  scale_color_manual(values = c("neg" = "red", "pos" = "blue")) +
  scale_shape_manual(values = c("neg" = 16, "pos" = 17)) +
  labs(title = "PCA of Pima Indians Diabetes Dataset (PC1 vs PC3)",
       x = paste("PC1 -", round(summary(res.pca)$importance[2, 1] * 100,
1), "% Variance"),
       y = paste("PC3 -", round(summary(res.pca)$importance[2, 3] * 100,
1), "% Variance"),
       color = "Diabetes Status", shape = "Diabetes Status") +
  theme_minimal()
print(pca_plot_1_3)

pca_plot_2_3 <- ggplot(New_PimaIndiansDiabetes, aes(x = PC2, y = PC3, color
```

```
= diabetes, shape = diabetes)) +
  geom_point(size = 3, alpha = 0.6) +
  stat_ellipse(type = "norm", linetype = 2, level = 0.95) +
  scale_color_manual(values = c("neg" = "red", "pos" = "blue")) +
  scale_shape_manual(values = c("neg" = 16, "pos" = 17)) +
  labs(title = "PCA of Pima Indians Diabetes Dataset (PC2 vs PC3)",
       x = paste("PC2 -", round(summary(res.pca)$importance[2, 2] * 100,
1), "% Variance"),
       y = paste("PC3 -", round(summary(res.pca)$importance[2, 3] * 100,
1), "% Variance"),
       color = "Diabetes Status", shape = "Diabetes Status") +
  theme_minimal()
print(pca_plot_2_3)
```

```
# Form Test/Training Sets using PCA Data (Part 13)
set.seed(123)
ind <- sample(2, nrow(New_PimaIndiansDiabetes), replace = TRUE, prob =
c(0.6, 0.4))
training_pca <- New_PimaIndiansDiabetes[ind == 1, ]
testing_pca <- New_PimaIndiansDiabetes[ind == 2, ]
dim(New_PimaIndiansDiabetes)
dim(training_pca)
dim(testing_pca)
table(training_pca$diabetes)
table(testing_pca$diabetes)
boxM_result <- boxM(training[, 1:3], training_pca$diabetes)
print(boxM_result)
```

```
# Run QDA on the training set (Part 14)
quadratic_pca <- qda(diabetes ~ ., data = training_pca)
partimat(diabetes ~ ., data = training_pca, method = "qda", mar = c(2, 2,
2, 2))
p1_qda_pca <- predict(quadratic_pca, training_pca)$class
tab_qda_pca <- table(Predicted = p1_qda_pca, Actual =
training_pca$diabetes)
p2_qda_pca <- predict(quadratic_pca, testing_pca)$class
tab1_qda_pca <- table(Predicted = p2_qda_pca, Actual =
testing_pca$diabetes)
accuracy_train_after <- sum(diag(tab_qda_pca)) / sum(tab_qda_pca)
```

```
accuracy_test_after <- sum(diag(tab1_qda_pca)) / sum(tab1_qda_pca)
print(paste("Training Accuracy after PCA:", accuracy_train_after))
print(paste("Testing Accuracy after PCA:", accuracy_test_after))
```

```
# CV For PCA Data (Part 15)
model_after <- train(diabetes ~ ., data = training_pca, method = "qda",
trControl = ctrl)
print(model_after)
cv_accuracy_after <- model_after$results$Accuracy
```

```
# Accuracy Comparison (Part 16)
accuracy_data <- data.frame(
  Set = rep(c("Training", "Testing", "Cross-Validation"), each = 2),
  Accuracy = c(accuracy_train_before, accuracy_train_after,
accuracy_test_before, accuracy_test_after, cv_accuracy_before,
cv_accuracy_after),
  PCA = rep(c("Before PCA", "After PCA"), times = 3)
)
accuracy_plot <- ggplot(accuracy_data, aes(x = Set, y = Accuracy, fill =
PCA)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Comparison of QDA Accuracy Before and After PCA", x = "Data
Set", y = "Accuracy", fill = "PCA Status") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
axis.title = element_text(size = 12), legend.title = element_text(size =
12), legend.text = element_text(size = 10))
print(accuracy_plot)
```