PCA_with_sample_datasets

Humaun Farid Sohag

2025-08-13

Step 1 - Importing Required Libraries

```
library(caret)      # For scaling and train/test split
library(ggplot2)    # For plotting
library(gridExtra)  # For arranging plots
library(reshape2)   # For reshaping confusion matrix
library(dplyr)      # For data manipulation
```

Step 2 - Creating Sample Dataset

```
df <- data.frame(
  Height = c(170, 165, 180, 175, 160, 172, 168, 177, 162, 158),
  Weight = c(65, 59, 75, 68, 55, 70, 62, 74, 58, 54),
  Age    = c(30, 25, 35, 28, 22, 32, 27, 33, 24, 21),
  Gender = factor(c(1, 0, 1, 1, 0, 1, 0, 1, 0, 0), labels = c("Female", "Male"))
)
df
```

```
##    Height Weight Age Gender
## 1     170     65  30   Male
## 2     165     59  25 Female
## 3     180     75  35   Male
## 4     175     68  28   Male
## 5     160     55  22 Female
## 6     172     70  32   Male
## 7     168     62  27 Female
## 8     177     74  33   Male
## 9     162     58  24 Female
## 10    158     54  21 Female
```

Step 3 - Standardizing the Data

```
X <- df %>% select(-Gender);X
```

```
##    Height Weight Age
## 1     170     65  30
## 2     165     59  25
## 3     180     75  35
## 4     175     68  28
## 5     160     55  22
## 6     172     70  32
## 7     168     62  27
## 8     177     74  33
## 9     162     58  24
## 10    158     54  21
```

```
y <- df$Gender;y
```

```
##  [1] Male   Female Male   Male   Female Male   Female Male   Female Female
## Levels: Female Male
```

```
preProc <- preProcess(X, method = c("center", "scale"))
preProc
```

```
## Created from 10 samples and 3 variables
##
## Pre-processing:
##   - centered (3)
##   - ignored (0)
##   - scaled (3)
```

```
X_scaled <- predict(preProc, X)
X_scaled
```

```
##        Height     Weight         Age
## 1   0.1747456  0.1315587  0.48297827
## 2  -0.4973530 -0.6577935 -0.56697449
## 3   1.5189428  1.4471457  1.53293102
## 4   0.8468442  0.5262348  0.06299717
## 5  -1.1694516 -1.1840283 -1.19694614
## 6   0.4435851  0.7893522  0.90295937
## 7  -0.0940938 -0.2631174 -0.14699339
## 8   1.1156837  1.3155870  1.11294992
## 9  -0.9006121 -0.7893522 -0.77696504
## 10 -1.4382910 -1.3155870 -1.40693669
```

Step 4 - Applying PCA algorithm

```
pca_model <- prcomp(X_scaled, center = FALSE, scale. = FALSE)
X_pca <- pca_model$x[, 1:2]

# Train-test split (70% train, 30% test)
set.seed(42)
```

```r
trainIndex <- createDataPartition(y, p = 0.7, list = FALSE)
X_train <- X_pca[trainIndex, ]
X_test  <- X_pca[-trainIndex, ]
y_train <- y[trainIndex]
y_test  <- y[-trainIndex]

# Logistic Regression Model
train_data <- data.frame(PC1 = X_train[,1], PC2 = X_train[,2], Gender = y_train)
test_data  <- data.frame(PC1 = X_test[,1],  PC2 = X_test[,2],  Gender = y_test)

model <- glm(Gender ~ ., data = train_data, family = binomial)
y_pred_prob <- predict(model, test_data, type = "response")
y_pred_prob
```

```
##           1           2
## 1.763212e-01 2.220446e-16
```

```r
y_pred <- factor(ifelse(y_pred_prob > 0.5, "Male", "Female"), levels = c("Female", "Male"))
y_pred
```

```
##      1      2
## Female Female
## Levels: Female Male
```

Step 5 - Evaluating with Confusion Matrix

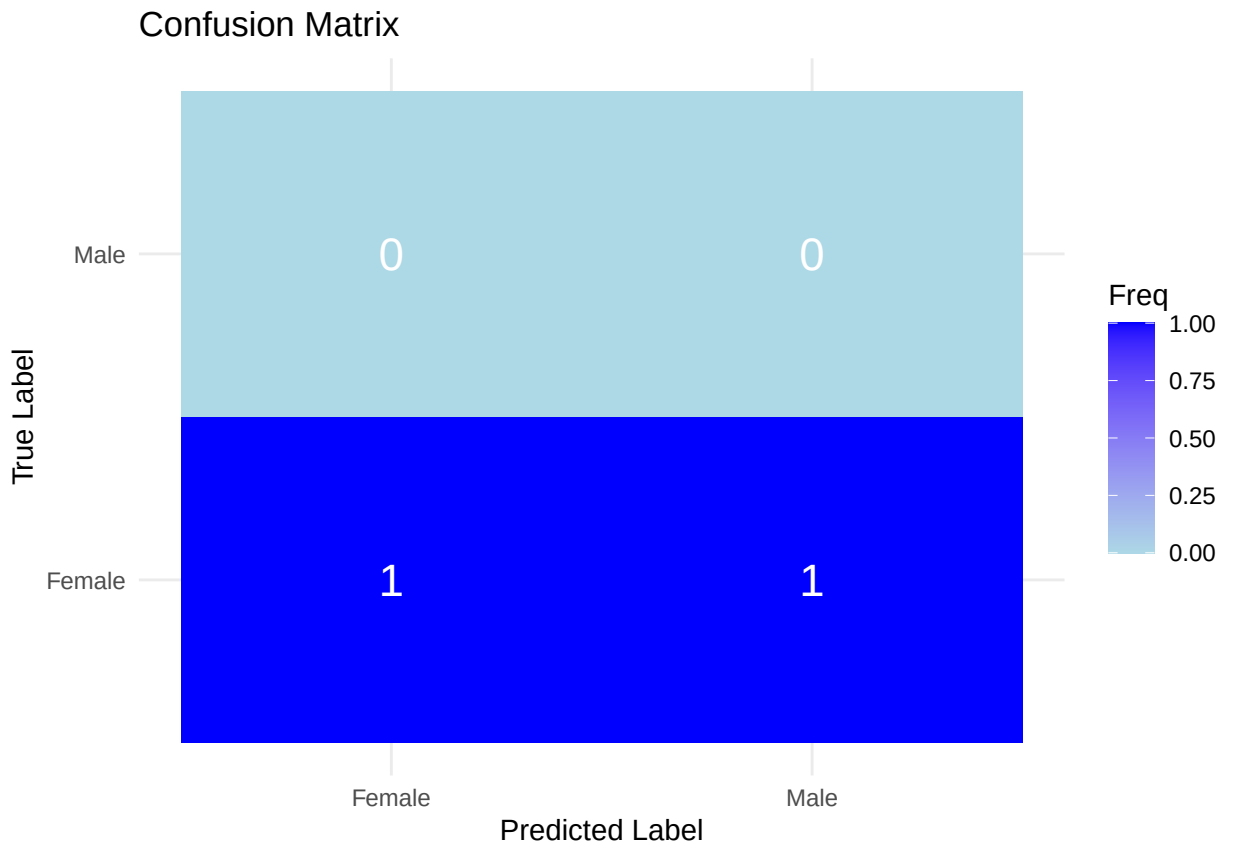
```r
cm <- confusionMatrix(y_pred, y_test)
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Female Male
##     Female      1    1
##     Male        0    0
##
##                Accuracy : 0.5
##                  95% CI : (0.0126, 0.9874)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.75
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 1.00
##
##             Sensitivity : 1.0
##             Specificity : 0.0
##          Pos Pred Value : 0.5
##          Neg Pred Value : NaN
```

```
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 1.0
##       Balanced Accuracy : 0.5
##
##        'Positive' Class : Female
##
```

```r
# Create a heatmap-like plot
cm_table <- as.data.frame(cm$table)
colnames(cm_table) <- c("True", "Predicted", "Freq")

ggplot(cm_table, aes(x = Predicted, y = True, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 6) +
  scale_fill_gradient(low = "lightblue", high = "blue") +
  labs(title = "Confusion Matrix", x = "Predicted Label", y = "True Label") +
  theme_minimal()
```



Step 6 - Visualizing PCA Result

```r
# Before PCA: first 2 standardized features
scaled_df <- data.frame(Feature1 = X_scaled[,1], Feature2 = X_scaled[,2], Gender = y)
```

```r
p1 <- ggplot(scaled_df, aes(x = Feature1, y = Feature2, color = Gender)) +
  geom_point(size = 3) +
  labs(title = "Before PCA: Using First 2 Standardized Features") +
  theme_minimal()+
  theme(plot.title = element_text(size = 11))

# After PCA
pca_df <- data.frame(PC1 = X_pca[,1], PC2 = X_pca[,2], Gender = y)

p2 <- ggplot(pca_df, aes(x = PC1, y = PC2, color = Gender)) +
  geom_point(size = 3) +
  labs(title = "After PCA: Projected onto 2 Principal Components") +
  theme_minimal()+
  theme(plot.title = element_text(size = 11))

# Arrange side by side
grid.arrange(p1, p2, ncol = 2)
```