Iris Data Analysis

2025-05-05

Data Analysis

Data Overview

```r
data(iris)

# Overview
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
str(iris)
```

```
## ’data.frame’:    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
summary(iris)
```

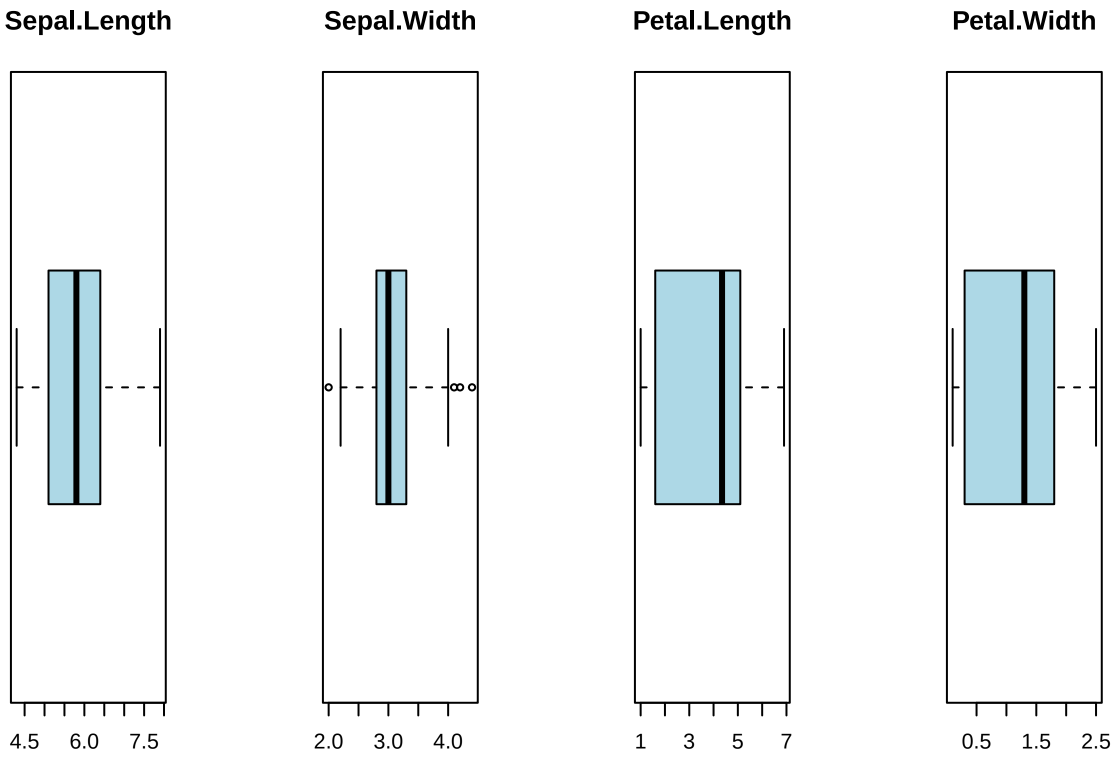
```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 
```

The dataset consists of 150 flower samples with 4 numeric features and 1 categorical target (Species). The task is a multiclass classification problem with 3 classes: setosa, versicolor, virginica.

Analyze Numeric Deatures

We visualize the distribution of each numeric feature using boxplots to detect outliers and variation. All features are continuous and measured in cm.

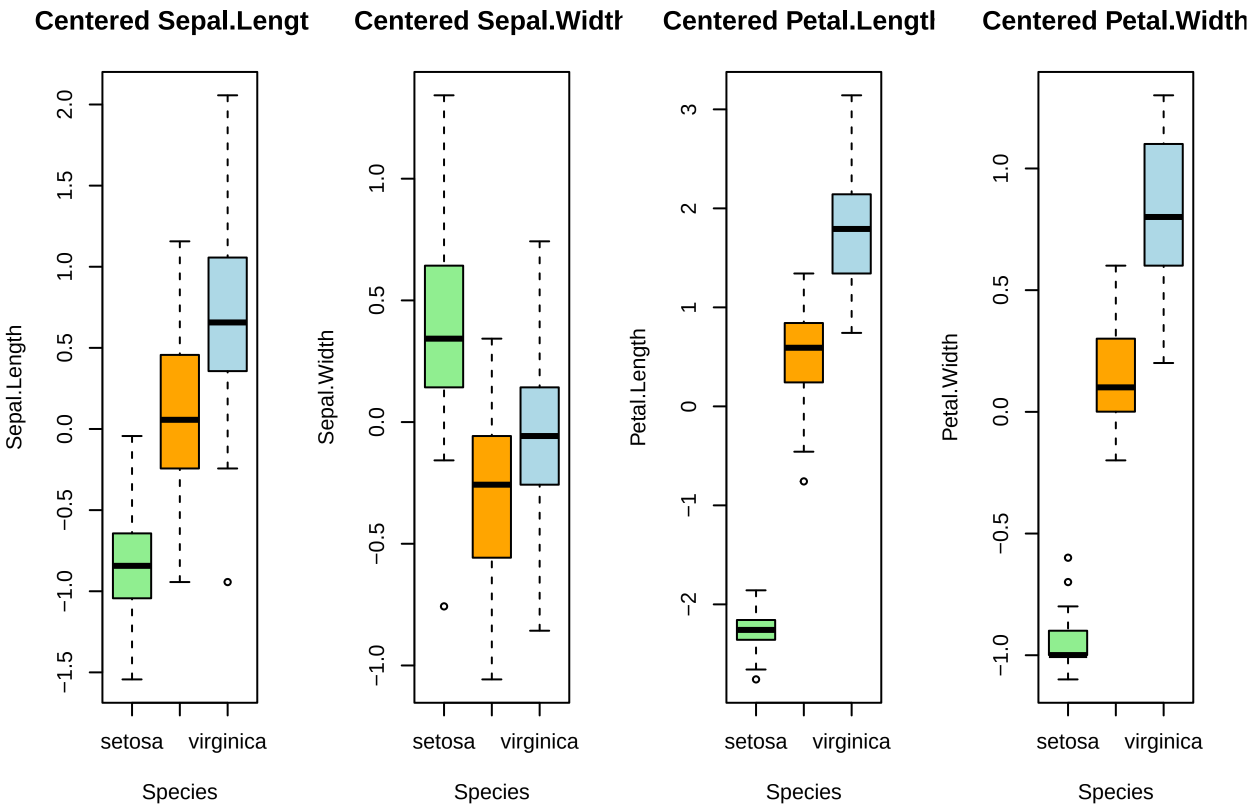
```r
# Select numeric features
iris_numeric <- iris[, 1:4]

# Boxplots of raw (uncentered) data
par(mfrow = c(1, 4))
for (col in names(iris_numeric)) {
  boxplot(iris_numeric[[col]],
          main = col,
          col = "lightblue",
          horizontal = TRUE)
}
```



Centered Data & Species-wise Boxplot

To reduce bias from scale differences, we center the numeric features. We then examine how the distributions vary across species using boxplots.

```r
iris_centered <- scale(iris_numeric, center = TRUE, scale = FALSE)
iris_centered_df <- data.frame(iris_centered, Species = iris$Species)

# Boxplots per feature by species
par(mfrow = c(1, 4))
for (col in names(iris_centered_df)[1:4]) {
  boxplot(
    iris_centered_df[[col]] ~ iris_centered_df$Species,
    col = c("lightgreen", "orange", "lightblue"),
    main = paste("Centered", col),
    ylab = col,
    xlab = "Species"
  )
}
```



Train a simple K-NN classifier

A basic k-NN model is trained (k = 3). The dataset is randomly split into 100 training and 50 testing samples. Accuracy and confusion matrix are shown below.

```r
library(class)

set.seed(42)
```

```r
features <- iris_centered_df[, 1:4]
labels <- iris_centered_df$Species

train_idx <- sample(1:nrow(features), 100)
train_features <- features[train_idx, ]
test_features <- features[-train_idx, ]
train_labels <- labels[train_idx]
test_labels <- labels[-train_idx]

predicted_labels <- knn(train = train_features,
                        test = test_features,
                        cl = train_labels,
                        k = 3)

# Output
conf_mat <- table(Predicted = predicted_labels, Actual = test_labels)
print(conf_mat)
```

```
##             Actual
## Predicted    setosa versicolor virginica
##   setosa         13          0         0
##   versicolor      0         17         2
##   virginica       0          0        18
```

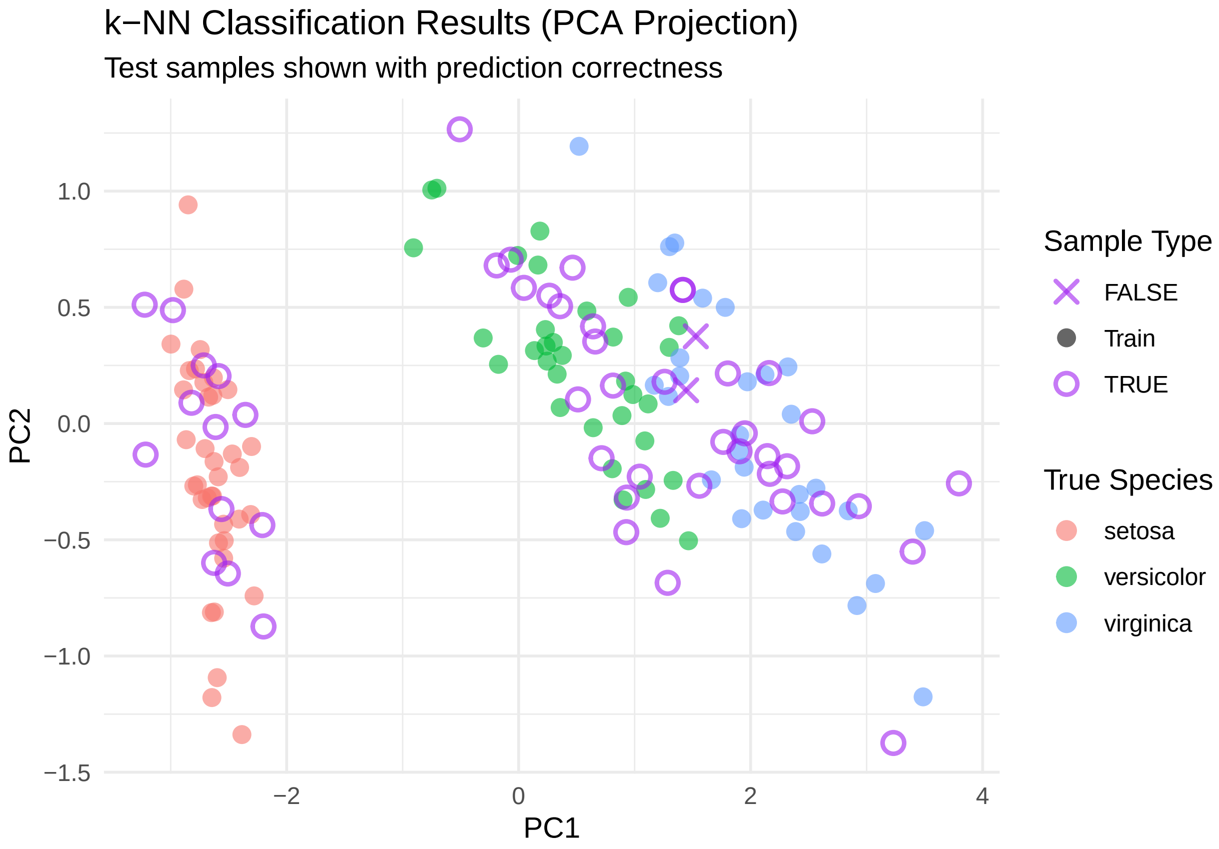
```r
# Accuracy
accuracy <- mean(predicted_labels == test_labels)
print(paste("Accuracy:", round(accuracy * 100, 2), "%"))
```

```
## [1] "Accuracy: 96 %"
```

visualize the Result with PCA

```r
library(ggplot2)

pca <- prcomp(iris_centered_df[, 1:4])
pca_data <- data.frame(pca$x[, 1:2],
                       Species = iris_centered_df$Species)

# Label data as Train/Test
pca_data$Set <- "Train"
pca_data$Set[-train_idx] <- "Test"

# Prediction results
pca_data$Predicted <- NA
pca_data$Predicted[-train_idx] <- as.character(predicted_labels)
pca_data$Correct <- NA
pca_data$Correct[-train_idx] <- pca_data$Species[-train_idx] == predicted_labels

# Plot
ggplot(pca_data, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = Species, shape = Set), size = 3, alpha = 0.6) +
  geom_point(data = subset(pca_data, Set == "Test"),
              aes(shape = Correct),
              size = 3,
              stroke = 1.2,
              color = "purple",
              alpha = 0.6) +
  scale_shape_manual(values = c("TRUE" = 1, "FALSE" = 4, "Train" = 16)) +
  labs(title = "k-NN Classification Results (PCA Projection)",
       subtitle = "Test samples shown with prediction correctness",
       color = "True Species", shape = "Sample Type") +
  theme_minimal()
```

```
## Warning: Removed 50 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


k−NN Classification Results (PCA Projection)
Test samples shown with prediction correctness
PC2
PC1
Sample Type
FALSE
Train
TRUE
True Species
setosa
versicolor
virginica