Classification of Iris Dataset Using Machine Learning Algorithms

Hekma Magdy

Introduction

The **Iris dataset** was used to classify flowers into three species: *Setosa, Versicolor, and Virginica*. The objective is to compare different machine learning algorithms and evaluate their performance using training (70%) and testing (30%) data splits.

Dataset Description

```r
data("iris")
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
dim(iris)
```

```
[1] 150   5
```

```r
str(iris)
```

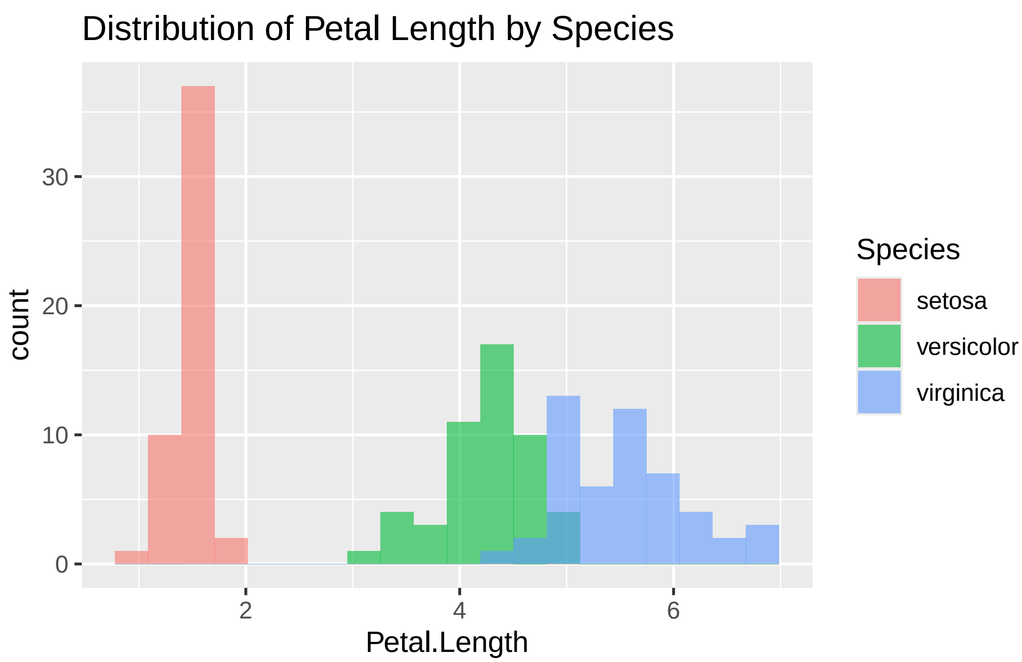
```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...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

```r
summary(iris)
```

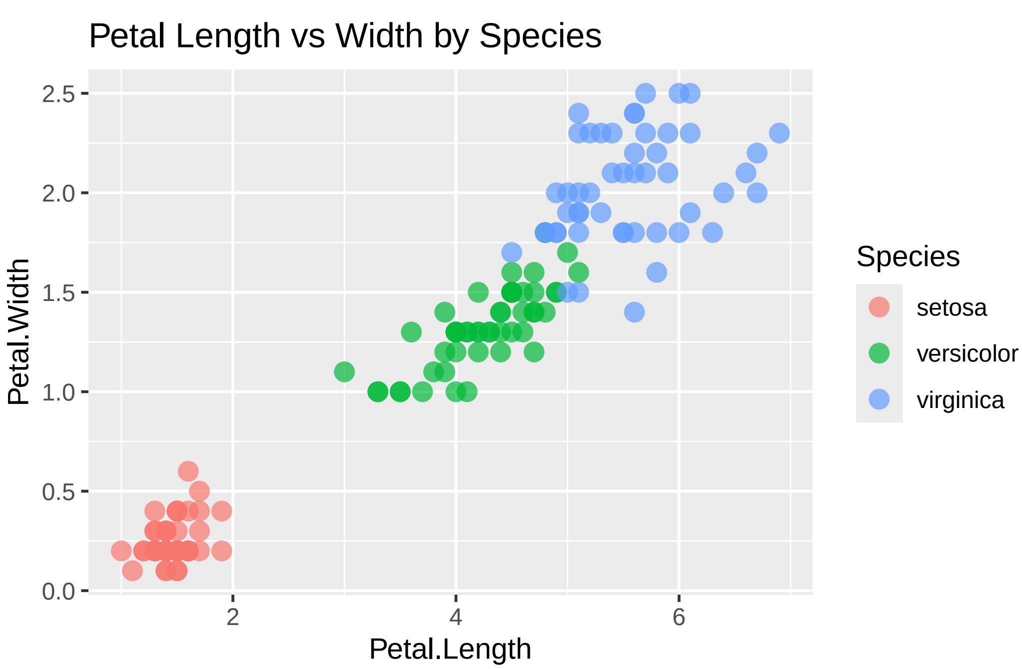
```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :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  
```

Data Preparation

```r
library(ggplot2)
ggplot(iris, aes(x = Petal.Length, fill = Species)) +
  geom_histogram(alpha = 0.6, bins = 20, position = "identity") +
  labs(title = "Distribution of Petal Length by Species")
```




```r
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Petal Length vs Width by Species")
```



```r
set.seed(42)
library(caret)
data_index =createDataPartition(iris$Species , p = 0.7 , list = FALSE)
train_data = iris[data_index,]
test_data = iris[-data_index ,]
```

Model Training

```r
#Logistic Regression
model_log =train(Species~. , data = train_data , method ="multinom")
# Decision Tree
model_tree =train(Species~. , data = train_data , method ="rpart")
# Random Forest
model_rf = train(Species~. , data = train_data , method ="rf")
```

Model Evaluation

```r
#Logistic Regression
log_predict= predict(model_log, test_data)
confusionMatrix(log_predict , test_data$Species)
# Decision Tree
tree_predict = predict(model_tree ,test_data , type = "raw")
confusionMatrix(tree_predict, test_data$Species)
#Random Forest
rf_predict = predict(model_rf , test_data)
confusionMatrix(rf_predict, test_data$Species)
```

Model Performance Comparison

```r
results <- data.frame(
  Model = c("Logistic Regression", "Decision Tree", "Random Forest"),
  Accuracy = c(0.978, 0.889, 0.956),
  Kappa = c(0.967, 0.833, 0.933),
  AUC = c(0.995, 0.950, 0.989)
)
knitr::kable(results, caption = "Comparison of Model Performance")
```

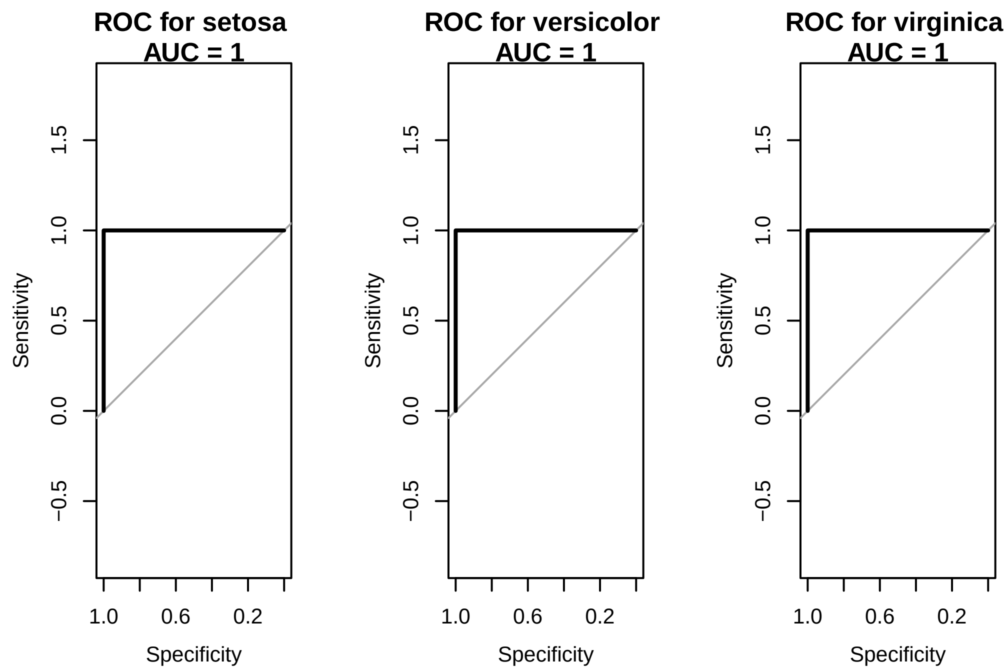
Table 1: Comparison of Model Performance

Model	Accuracy	Kappa	AUC
Logistic Regression	0.978	0.967	0.995
Decision Tree	0.889	0.833	0.950
Random Forest	0.956	0.933	0.989

ROC & AUC

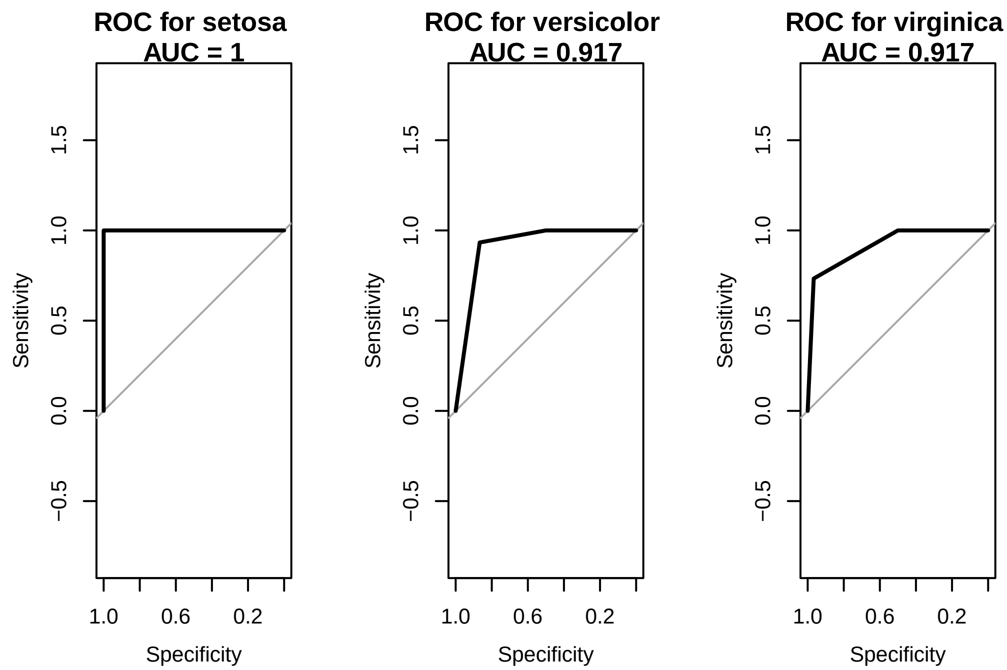
1- Logistic Regression

```r
library(pROC)

log_probs <- predict(model_log, test_data, type = "prob")

par(mfrow=c(1,3))

for (cls in levels(test_data$Species)) {
  binary_labels <- ifelse(test_data$Species == cls, 1, 0)
    roc_curve <- roc(binary_labels,
                     log_probs[, cls],
                     levels = c(0,1),
                     direction = "<")
    auc_value <- auc(roc_curve)
    plot(roc_curve,
       main = paste("ROC for", cls, "\nAUC =", round(auc_value, 3)))
}
```



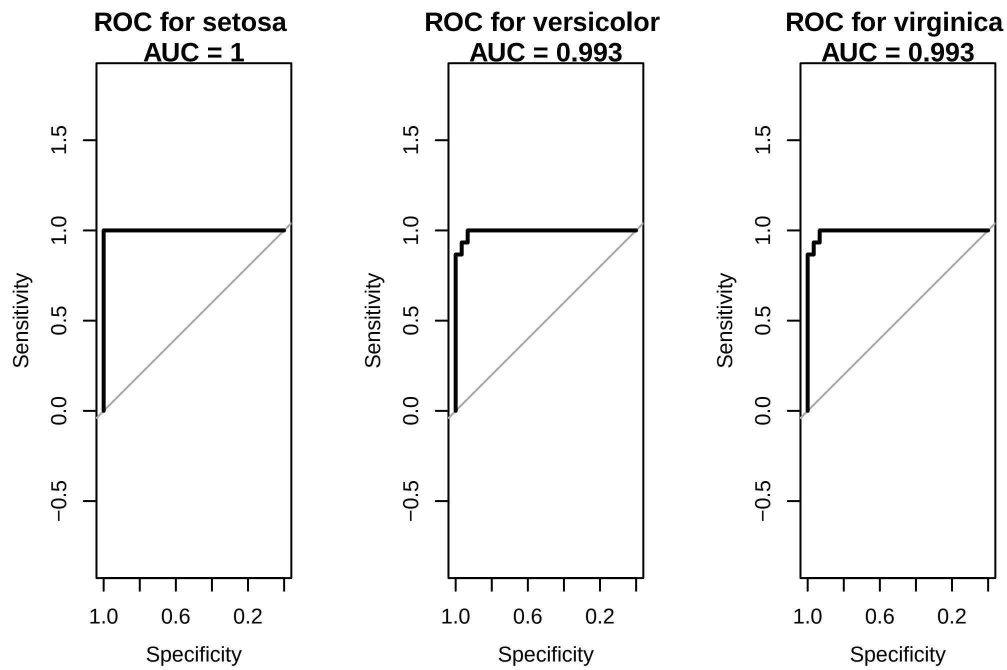
2- Decision Tree

```
tree_probs <- predict(model_tree, test_data, type = "prob")

par(mfrow=c(1,3))

for (cls in levels(test_data$Species)) {
  binary_labels <- ifelse(test_data$Species == cls, 1, 0)
  roc_curve <- roc(binary_labels,
                   tree_probs[, cls],
                   levels = c(0,1),
                   direction = "<")
    auc_value <- auc(roc_curve)
    plot(roc_curve,
       main = paste("ROC for", cls, "\nAUC =", round(auc_value, 3)))
}
```



3- Random Forest

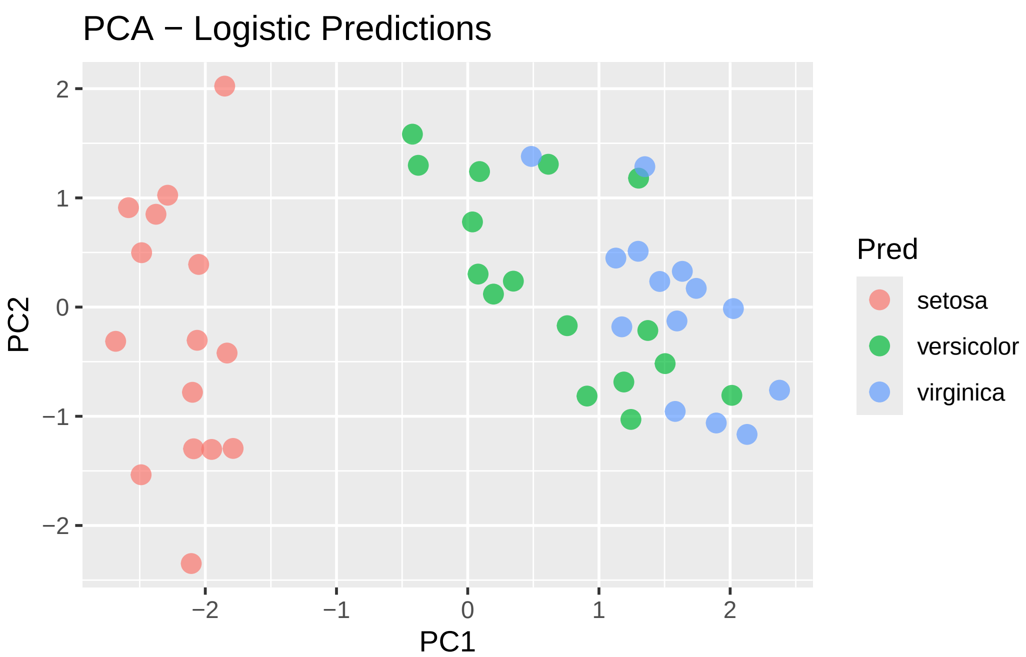
```
rf_probs <- predict(model_rf, test_data, type = "prob")

par(mfrow=c(1,3))

for (cls in levels(test_data$Species)) {
  binary_labels <- ifelse(test_data$Species == cls, 1, 0)
    roc_curve <- roc(binary_labels,
                  rf_probs[, cls],
                  levels = c(0,1),
                  direction = "<")
  auc_value <- auc(roc_curve)
    plot(roc_curve,
       main = paste("ROC for", cls, "\nAUC =", round(auc_value, 3)))
}
```

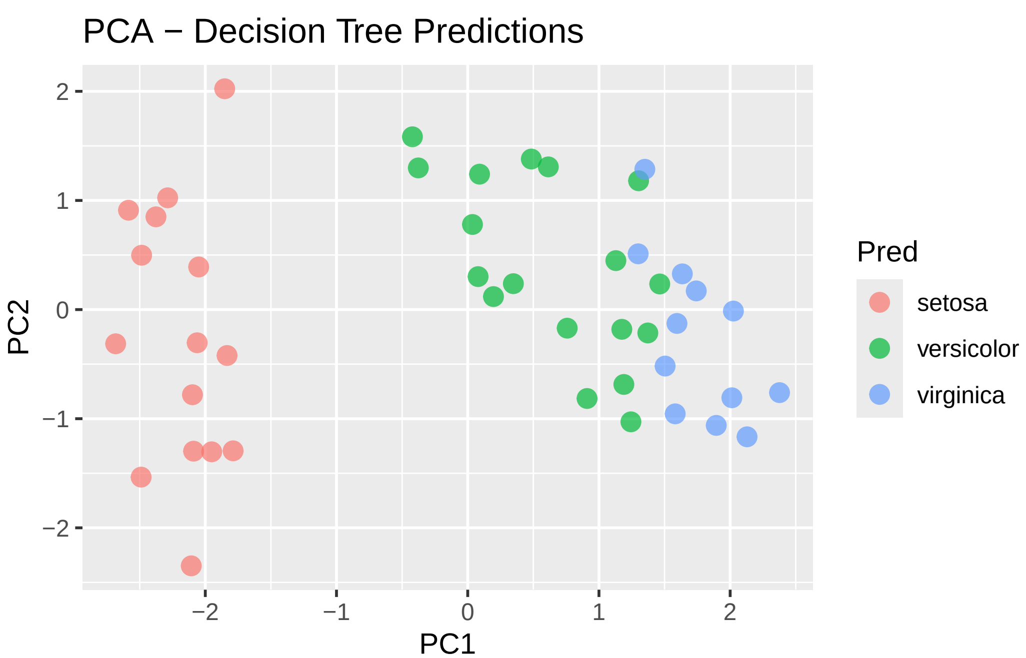


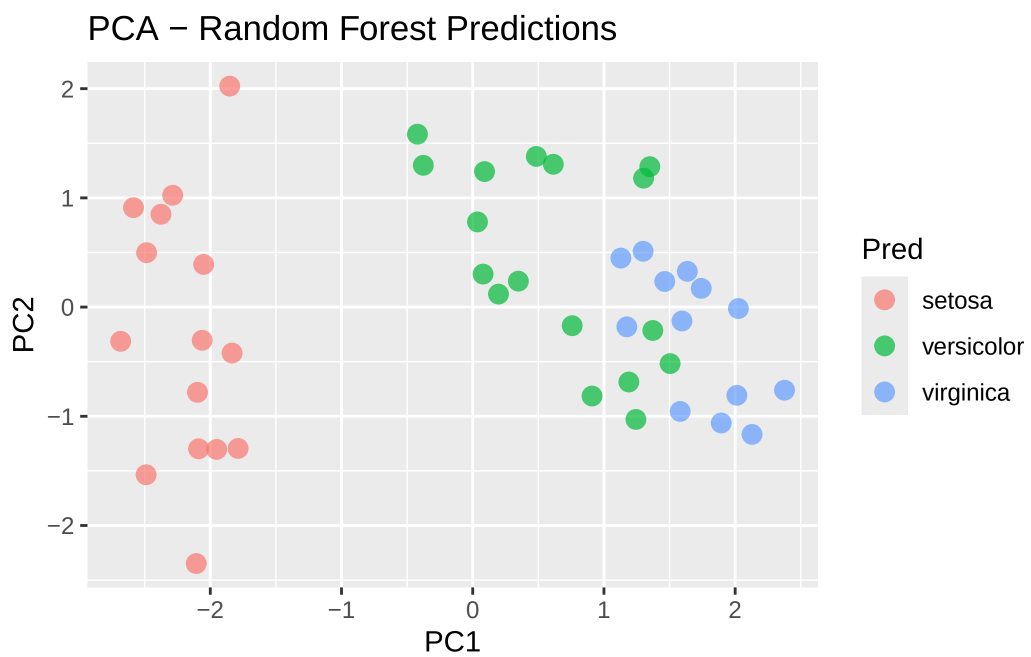
Visualization

PCA Visualization of Iris Features

```
pca_res <- prcomp(test_data[,1:4], scale. = TRUE)

pca_plot <- function(pred, method){
  pca_df <- data.frame(pca_res$x[,1:2], Pred = pred)
  ggplot(pca_df, aes(PC1, PC2, color = Pred)) +
    geom_point(size = 3, alpha = 0.7) +
    labs(title = paste("PCA -", method, "Predictions"))
}

p1 <- pca_plot(log_predict, "Logistic")
p2 <- pca_plot(tree_predict, "Decision Tree")
p3 <- pca_plot(rf_predict, "Random Forest")

print(p1); print(p2); print(p3)
```

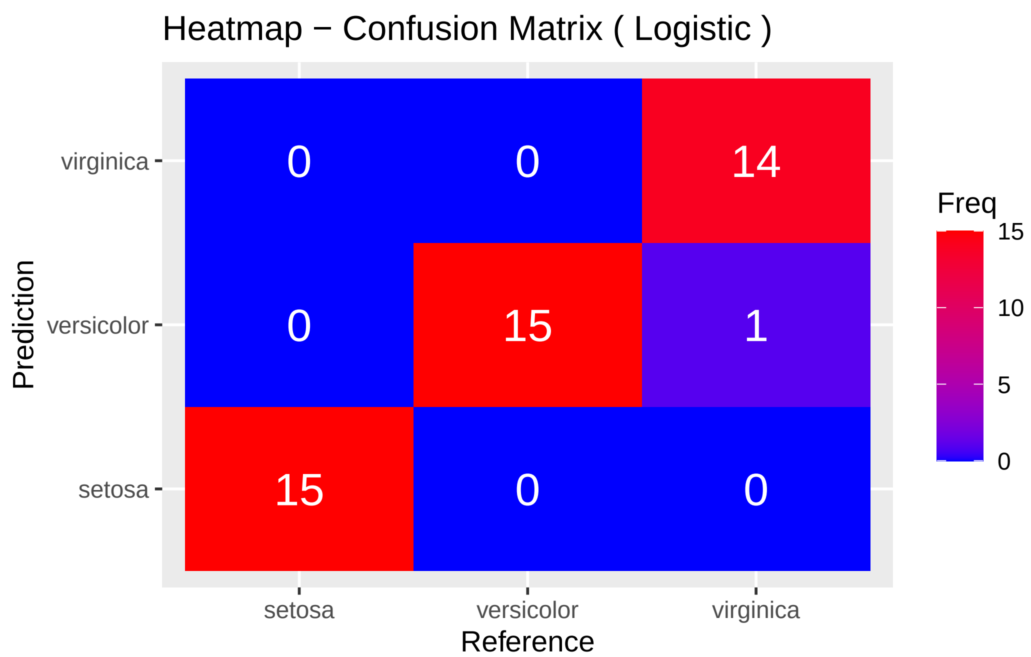

PCA − Logistic Predictions
PC2
2
1
0
−1
−2
PC1
−2
−1
0
1
2
Pred
setosa
versicolor
virginica

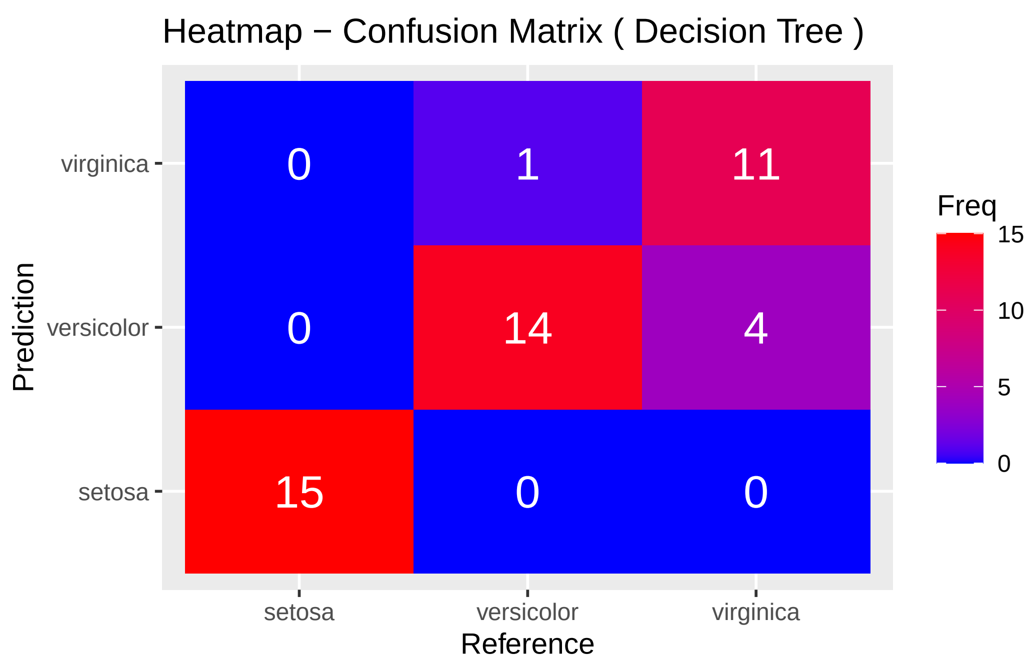

PCA − Decision Tree Predictions
PC2
2
1
0
−1
−2
PC1
−2
−1
0
1
2
Pred
setosa
versicolor
virginica

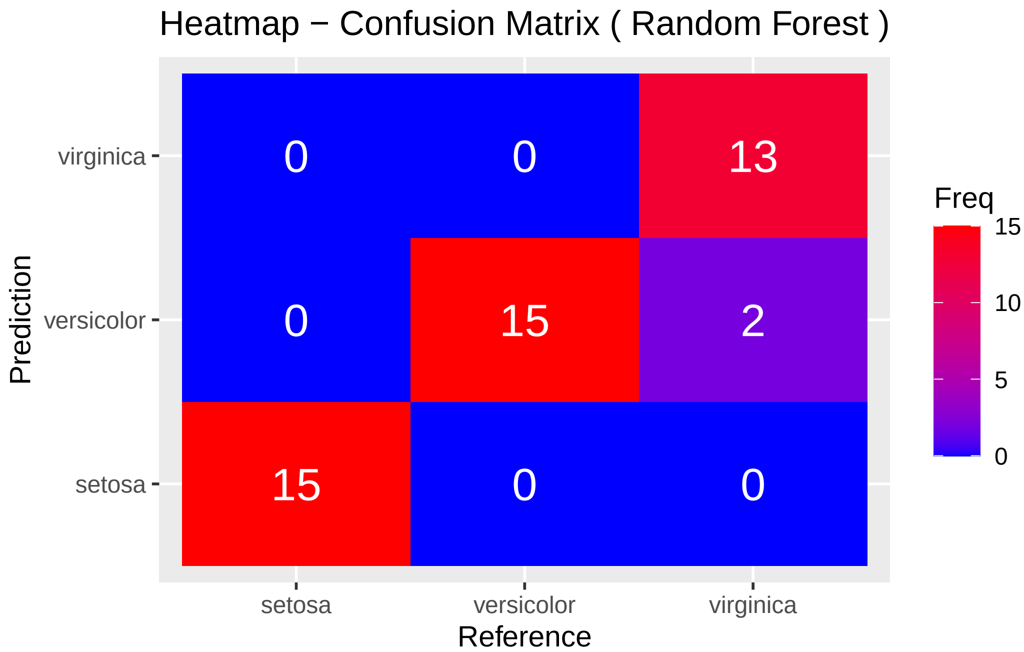


Confusion Matrix Heatmap

```
heatmap_cm <- function(pred, method){
  cm <- confusionMatrix(pred, test_data$Species)
  cm_table <- as.data.frame(cm$table)

  ggplot(cm_table, aes(Reference, Prediction, fill = Freq)) +
    geom_tile() +
    geom_text(aes(label = Freq), color = "white", size = 6) +
    scale_fill_gradient(low = "blue", high = "red") +
    labs(title = paste("Heatmap - Confusion Matrix (", method, ")"))
}

h1 <- heatmap_cm(log_predict, "Logistic")
h2 <- heatmap_cm(tree_predict, "Decision Tree")
h3 <- heatmap_cm(rf_predict, "Random Forest")

print(h1); print(h2); print(h3)
```


Heatmap − Confusion Matrix (Logistic)
Prediction
virginica
versicolor
setosa
0
0
14
0
15
1
15
0
0
setosa
versicolor
virginica
Reference
Freq
15
10
5
0


Heatmap − Confusion Matrix (Decision Tree)
Prediction
virginica
versicolor
setosa
0
1
11
0
14
4
15
0
0
setosa
versicolor
virginica
Reference
Freq
15
10
5
0



Conclusion

The experiments on the Iris dataset demonstrated that machine learning techniques are capable of classifying the three species with high accuracy and efficiency, with variations in performance depending on the nature of each algorithm and its approach to handling the data. The results showed that **Logistic Regression** achieved the highest accuracy in classification, followed by **Random Forest**, which delivered strong performance close to the first model, while **Decision Tree** ranked last with relatively lower performance. This study highlights the importance of comparing multiple models when addressing classification problems and emphasizes that selecting the most suitable algorithm largely depends on the characteristics of the data and the intended application context.