

Εισαγωγή στα Δέντρα Αποφάσεων

Πώς τρέχουμε τον κώδικα:

1. Ανοίξτε το RStudio.
2. Δημιουργήστε ένα νέο script (File > New File > R Script).
3. Επικολλήστε τον παραπάνω κώδικα μέσα στο script.
4. Αποθηκεύστε το με ένα ενδεικτικό όνομα, π.χ., r_course_2.R.
5. Εκτελέστε γραμμή προς γραμμή (επιλέξτε και πατήστε Ctrl+Enter ή χρησιμοποιήστε το κουμπί "Run") ώστε να παρακολουθείτε τη διαδικασία. Διαφορετικά, μπορείτε να επιλέξετε και να τρέξετε ολόκληρο τον κώδικα απευθείας.

Αντιγράψτε το παρακάτω κείμενο στο RStudio

Θα το τρέξουμε γραμμή-γραμμή. Επίσης, θα χρειαστεί να εγκαταστήσουμε 2 βιβλιοθήκες. Για να τις εγκαταστήσουμε, τρέχουμε τις παρακάτω εντολές, μία-μία.

```
install.packages("rpart")
```

```
install.packages("rpart.plot")
```

Παράδειγμα εκτέλεσης δέντρων απόφασης σε R (πάνω στο dataset iris)

Example 1: Decision Tree on the Iris Dataset (Classification)

```
# Load the necessary libraries  
library(rpart)  
library(rpart.plot)
```

```
# Load the iris dataset
data(iris)

# Build a classification decision tree
iris_tree <- rpart(Species ~ ., data = iris, method = "class")

# Plot the decision tree
rpart.plot(iris_tree, main = "Decision Tree for Iris Species")

# Make predictions on the dataset
predictions <- predict(iris_tree, iris, type = "class")

# Generate confusion matrix
conf_mat <- table(Predicted = predictions, Actual = iris$Species)

# Display the confusion matrix
print(conf_mat)
```

Example 2: Regression Tree on the mtcars dataset

```
# Load libraries
library(rpart)
library(rpart.plot)

# Build a regression tree to predict mpg
mtcars_tree <- rpart(mpg ~ wt + hp + cyl, data = mtcars, method = "anova")

# Plot the regression tree
rpart.plot(mtcars_tree, main = "Regression Tree for mpg")

# Make predictions
predicted_mpg <- predict(mtcars_tree, mtcars)

# Compare actual vs predicted
head(cbind(Actual = mtcars$mpg, Predicted = predicted_mpg))
```

Example 3: Visualizing the Decision Tree

```
# Basic plot of the tree structure with base R
plot(iris_tree, uniform=TRUE, main="Decision Tree Structure")
text(iris_tree, use.n=TRUE, all=TRUE, cex=0.8)
```

Προαιρετικό μέρος (αλλά πολύ χρήσιμο)

1. Decision Trees in R using tidymodels

Παρόμοια παραδείγματα με τα παραπάνω, αλλά με χρήση μιας διαφορετικής, πολύ δημοφιλούς βιβλιοθήκης. Εδώ θα χρησιμοποιήσουμε και τη βιβλιοθήκη ggplot για να μας βοηθήσει στην οπτικοποίηση.

2. Loading Libraries and Data

```
# Load necessary libraries
library(tidymodels)
library(rpart.plot)    # For tree visualization after fitting
library(tidypredict)   # For plotting decision trees

# Load the iris dataset
data(iris)
```

3. Data Preparation

```
# Check data structure
str(iris)

# Split data into training and testing sets (optional but recommended)
set.seed(123)
iris_split <- initial_split(iris, prop = 0.8)
iris_train <- training(iris_split)
iris_test  <- testing(iris_split)
```

4. Defining the Model Specification

```
# Define the decision tree model (classification)
dt_spec <- decision_tree() %>%
  set_engine("rpart") %>%
  set_mode("classification")
```

5. Creating the Workflow

```
# Set up workflow
iris_wf <- workflow() %>%
  add_model(dt_spec) %>%
  add_formula(Species ~ .)
```

6. Training the Model

```
# Fit the decision tree
iris_fit <- fit(iris_wf, data = iris_train)
```

7. Visualize the Decision Tree

```
# Extract the fitted rpart object
library(rpart.plot)
rpart_obj <- pull_workflow_fit(iris_fit)$fit

# Plot the tree
rpart.plot(rpart_obj, main = "Decision Tree for Iris (tidymodels)")
```

This produces a visual similar to previous rpart examples, showing splits and decision rules.

8. Making Predictions

```
# Predict on the test set
iris_preds <- predict(iris_fit, iris_test) %>%
  bind_cols(iris_test)

# View predictions
head(iris_preds)
```

9. Evaluate Model Performance

```
# Confusion matrix
conf_mat <- conf_mat(iris_preds, truth = Species, estimate = .pred_class)
autoplot(conf_mat)

# Calculate accuracy
accuracy_val <- accuracy(iris_preds, truth = Species, estimate = .pred_class)
print(accuracy_val)
```

10. Visualizing Misclassified Points

```
# Plot Sepal.Length vs Sepal.Width, color by true class, shape by predicted class
library(ggplot2)
```

```
ggplot(iris_preds, aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape
= .pred_class)) +
  geom_point(size=3) +
  labs(title = "Misclassified Points in Iris Dataset",
        shape = "Predicted Class") +
  theme_minimal()
```

Στο opencourse υπάρχει το παραπάνω προαιρετικό script σε αυτοτελή μορφή (lab_3_additional.txt).