````
---
title: "beadando"
author: "euv0j3"
date: "`r Sys.Date()`"
output:
  pdf_document: default
  html_document: default
---

# exams_data_writing_score

## Introduction

### Változók leírása

-   Gender
-   Race/ethnicity
-   parental level of education
-   lunch
-   test preparation course
-   math score
-   reading score
-   writing score

### Osztályok

-   writing_score: 75% alatti eredmények és 75% feletti eredmények

### Feladat

a meglévő adatok alapján egy osztályozó modell létrehozása R-ben, a modell futtatása és beküldése a Moodle-ba.

Két fájlt kell feltöltenI:

1.  R program
2.  Az eredmények leírása Word fájlban 1 oldalonThe task was to select 1 out of the 10 exercises and solve the problem.

## Setup

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
library(dplyr)
library(tidyverse)
library(caret)
library(janitor)
library(rpart)
library(rpart.plot)
library(randomForest)
library(gmodels)
library(ggplot2)
library(C50)
```

## Load the Data Set

```{r}
data <- read.csv(file.choose(), sep=",") # exams_writing_score_dataset.csv
data <- janitor::clean_names(data, "snake")
data
```

## Check data

```{r}
````
````

```
print(sum(is.na(data)))
```

There are no missing data values

```{r}
ggplot(data = data, aes(x = math_percentage, y = reading_score_percentage, color = sex))
+
  geom_point() +
  labs(x = "Math Percentage", y = "Reading Percentage", title = "Math vs. Reading
Scores")
```

```{r}
ggplot(data = data, aes(x = parental_level_of_education, y = writing_score_percentage,
fill = sex)) +
  geom_boxplot() +
  labs(x = "Parental Level of Education", y = "Writing Percentage", title = "Writing
Scores by Parental Level of Education") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```{r}
ggplot(data = data, aes(x = race_ethnicity, fill = sex)) +
  geom_bar() +
  labs(x = "Race/Ethnicity", y = "Count", title = "Count of Students by Race/Ethnicity")
+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```{r}
ggplot(data = data, aes(x = math_percentage, fill = sex)) +
  geom_histogram(binwidth = 0.1) +
  labs(x = "Math Percentage", y = "Count", title = "Distribution of Math Scores by Lunch
and Test Preparation Course") +
  facet_grid(lunch ~ test_preparation_course)
```

Binary Variable

Create a new binary variable called writing_class based on the writing_score column (1 for results above 75% and 0 for results below or equal to 75%)

```{r}
data$writing_class <- ifelse(data$writing_score_percentage > 0.75, 1, 0)

# Remove the old column
data$writing_score_percentage <- NULL
```

```{r}
# Convert writing_class to a factor
data$writing_class <- as.factor(data$writing_class)
```

Split the dataset into training and testing sets:

```{r}
set.seed(12345)
# Split the data into training (80%) and testing (20%) sets
sample_size <- floor(0.8 * nrow(data))
train_indices <- sample(seq_len(nrow(data)), size = sample_size)
train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]
```

```{r}
model <- C5.0(train_data[, -which(names(train_data) == "writing_class")],
              train_data$writing_class,
```

```
              rules = FALSE)

# Print the model
print(model)
```

```{r}
summary(model)
```

```{r}
predictions <- predict(model, test_data[, -which(names(test_data) == "writing_class")],
type = "class")
```

```{r}
confusion_matrix <- confusionMatrix(predictions, test_data$writing_class)

# Print the confusion matrix
print(confusion_matrix)
```

```{r}
plot(model)
```

```{r}
if (!requireNamespace("pROC", quietly = TRUE)) {
  install.packages("pROC")
}

# Load the pROC package
library(pROC)

# Predict the probabilities of the positive class (writing_class = 1)
predicted_probabilities <- predict(model, test_data, type = "prob")[, "1"]
roc_obj <- roc(test_data$writing_class, predicted_probabilities)
plot(roc_obj, main = "Decision Tree Model")
```

```{r}
# Calculate the AUC
auc_value <- auc(roc_obj)

# Print the AUC value
print(auc_value)

```
```
```