МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Практикум №6

з курсу «Аналіз даних в інформаційних системах»

на тему: «Класифікація та кластеризація»

Викладач:
Ліхоузова Т.А.

Виконав:
студент 2 курсу
групи ІП-14 ФІОТ
Шляхтун Денис

Київ-2023

Тема: Класифікація та кластеризація.

Мета роботи: ознайомитись з:

- методами класифікації та кластеризації;
- моделями, що використовують дерева прийняття рішень;
- інструментами факторного аналізу методом головних компонент та методом найбільшої подібності.

Основне завдання

Для даних по титаніку titanic.csv побудувати модель, в якій можна визначити, чи виживе пасажир, заповнивши решту параметрів.

Використати декілька методів. Порівняти результати.

Додаткове завдання

Використовуючи файл Data2.csv

1. визначити, який регіон домінує в кластерах по ВВП на душу населення та щільності населення
2. вивести частотні гістограми всіх показників файла Data2.csv, використовуючи цикл
3. створити функцію, яка на вхід отримує два набори даних, перевіряє чи є лінійна залежність та виводить True чи False (будемо розуміти під «є лінійна залежність», якщо коефіцієнт кореляції по модулю більше 0,8)

Виконання основного завдання.

Виконання комп’ютерного практикуму здійснювалося засобами R та RStudio.

Для виконання завдання було імпортовано файл titanic.csv, розглянуто його структуру та досліджено на відсутні значення.

```
> str(data)
'data.frame':   891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
...
 $ Sex        : chr  "male" "female" "female" "female" ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : chr  "" "C85" "" "C123" ...
 $ Embarked   : chr  "S" "C" "S" "S" ...
> summary(data)
  PassengerId       Survived          Pclass          Name               Sex                 Age            SibSp           Parch           Ticket
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891         Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character   Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
 Median :446.0   Median :0.0000   Median :3.000   Mode  :character   Mode  :character   Median :28.00   Median :0.000   Median :0.0000   Mode  :character
 Mean   :446.0   Mean   :0.3838   Mean   :2.309                                         Mean   :29.70   Mean   :0.523   Mean   :0.3816
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000                                         3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
 Max.   :891.0   Max.   :1.0000   Max.   :3.000                                         Max.   :80.00   Max.   :8.000   Max.   :6.0000
                                                                                        NA's   :177

      Fare            Cabin             Embarked
 Min.   :  0.00   Length:891         Length:891
 1st Qu.:  7.91   Class :character   Class :character
 Median : 14.45   Mode  :character   Mode  :character
 Mean   : 32.20
 3rd Qu.: 31.00
 Max.   :512.33

> naPerc
$names
 [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"         "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"        "Cabin"
[12] "Embarked"

$rate
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch      Ticket        Fare       Cabin    Embarked
   0.000000    0.000000    0.000000    0.000000    0.000000   19.865320    0.000000    0.000000    0.000000    0.000000   77.104377    0.224467
```

Рис. 1 – дослідження файлу titanic.csv

Було виявлено, що колонка Age має 20% пропущених значень, Cabin – 77%, Embarked – пропущені значення у 2 рядках.

Для Embarked було відкинуто рядки з пропущеними значеннями.

Для Cabin було відкинуто колонку, адже значень занадто мало для заповнення.

Для Age значення було заповнено середніми значеннями по такому ж класу каюти.

Також були відкинуті колонки Name, Ticket і PassengerID, так як мають не зовсім правильні значення і не впливають на те, чи виживе пасажир.

Для виконання завдання було розроблено три моделі на основі дерев рішень.

1. Одне дерево:

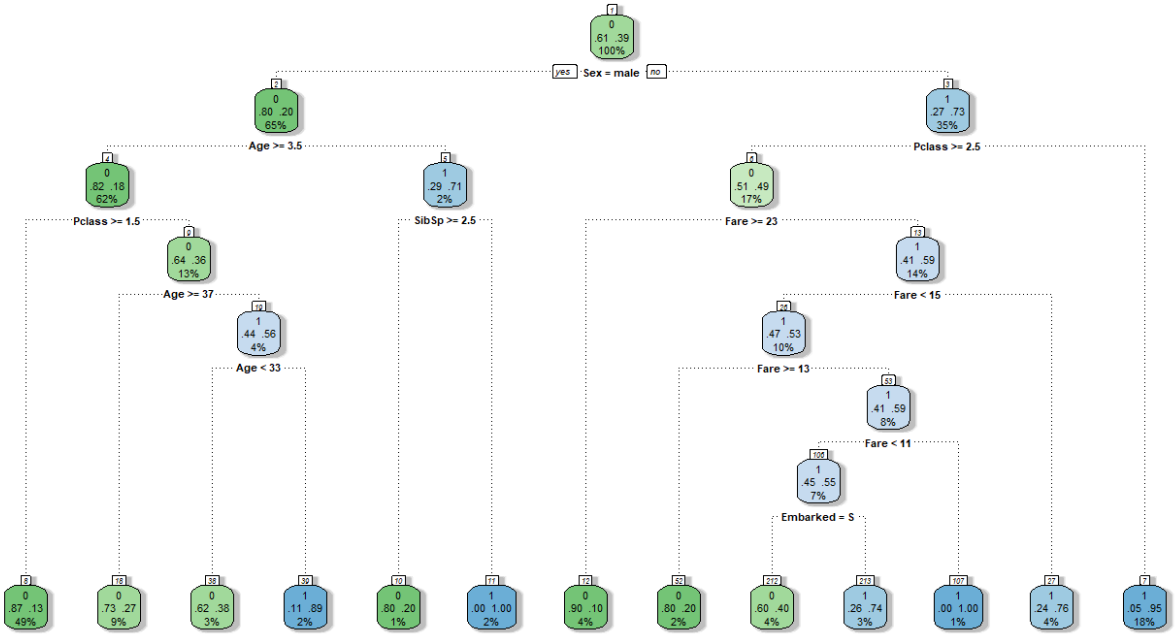

Рис. 2 – одне дерево

Похибка при перевірці на тестових даних: 0.15.

2. Ансамбль дерев.

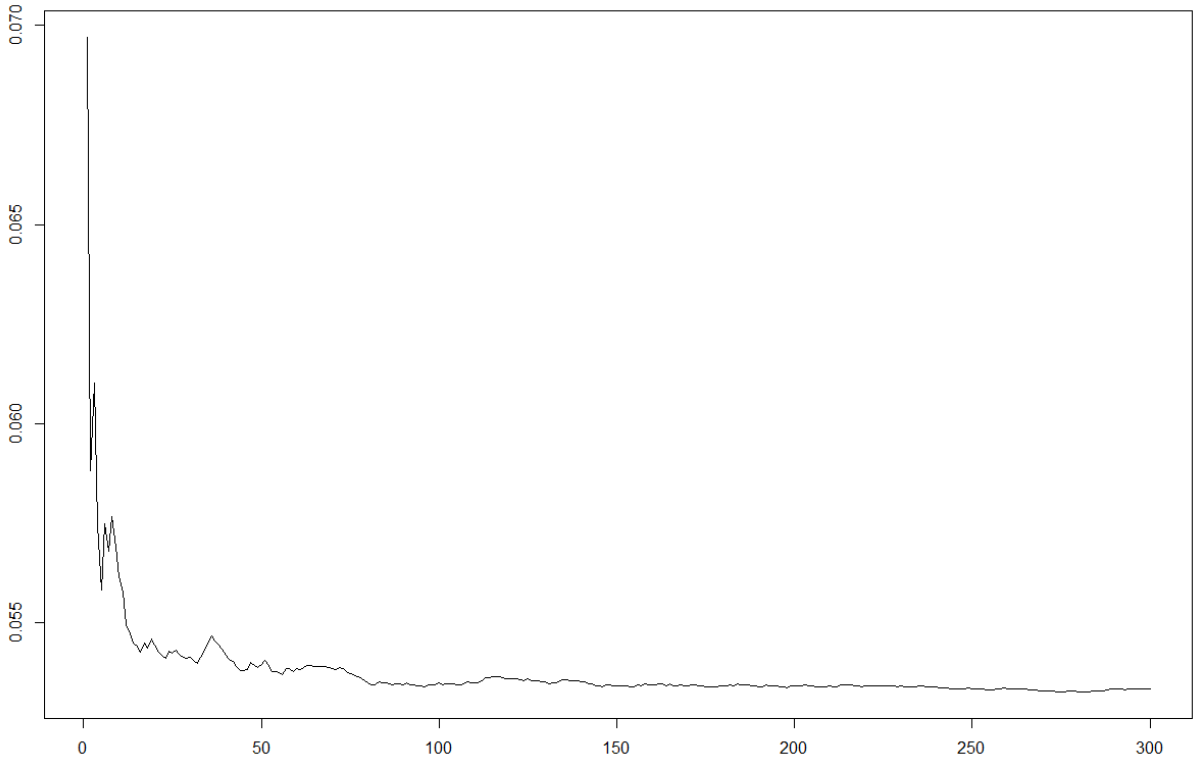

Рис. 3 – похибка залежно від кількості дерев

Похибка при перевірці на тестових даних: 0.053.

3. Випадковий ліс – створений засобами бібліотеки randomForest.

Похибка при перевірці на тестових даних: 0.036.

Можна зробити висновок, що випадковий ліс є найкращим методом, але одночасно найбільш вимогливим до апаратного забезпечення.

Виконання додаткового завдання.

Для виконання завдання було імпортовано файл Data2.csv і виправлено помилки у даних, як це відбувалося у попередніх комп'ютерних практикумах.

1. Визначити, який регіон домінує в кластерах по ВВП на душу населення та щільності населення.

Було оцінено кількість кластерів по tot.withinss

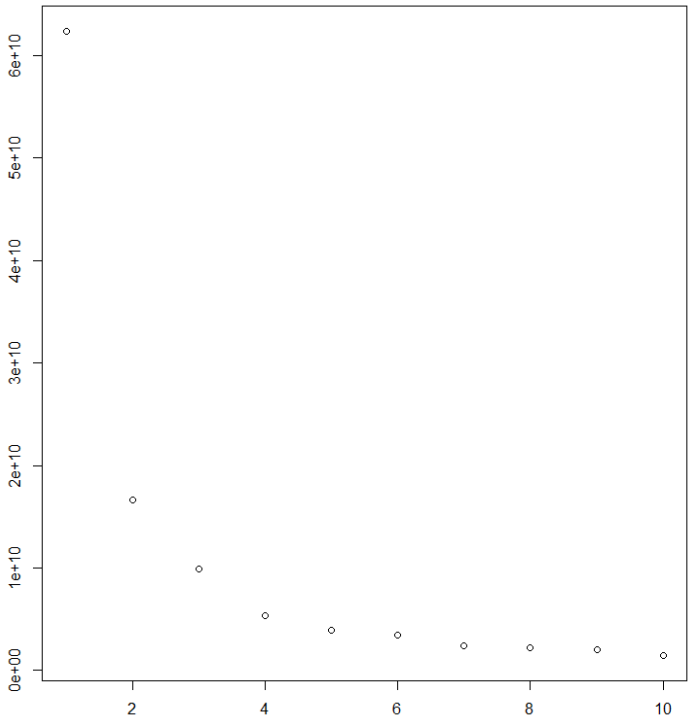

Рис. 4 – визначення кількості кластерів

По графіку видно, що далі 5 значення майже не змінюються, тому було обрано кількість кластерів 5.

За допомогою методу k-середніх було побудовано кластери.

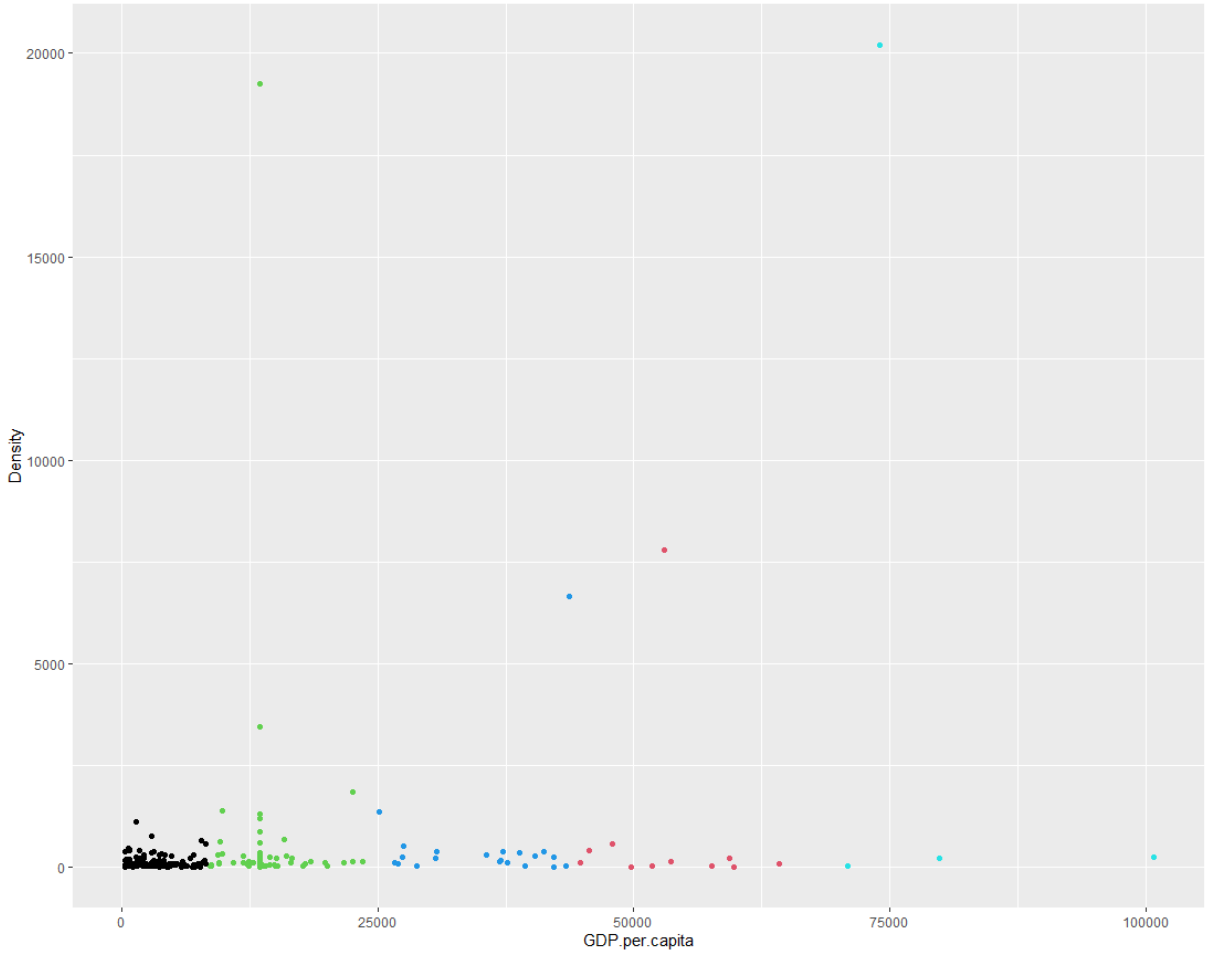

Рис. 5 – побудова кластерів

Далі визначимо за допомогою групування регіони, які домінують в кожному кластері:

cluster	Region	GDP.per.capita	density
<int>	<chr>	<dbl>	<dbl>
1	East Asia & Pacific	6610.	126.
2	Middle East & North Africa	59324.	221.
3	North America	13446.	1307.
4	East Asia & Pacific	36020.	253.
5	East Asia & Pacific	74017.	20204.

Рис. 6 – регіони, що домінують в кожному кластері

2. Вивести частотні гістограми всіх показників файла Data2.csv, використовуючи цикл

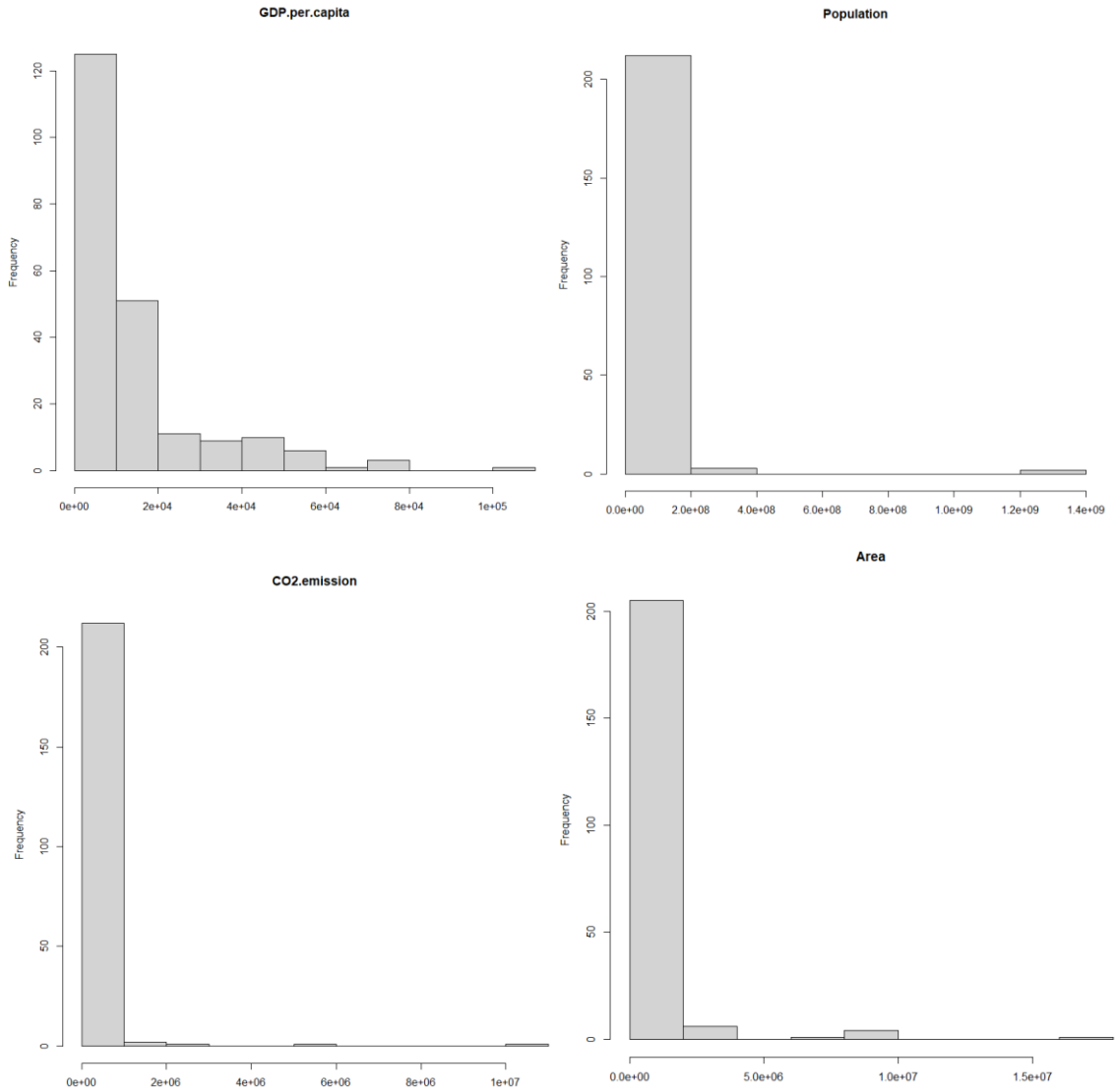

Рис. 7-10 – гістограми частот

3. Створити функцію, яка на вхід отримує два набори даних, перевіряє чи є лінійна залежність та виводить True чи False (будемо розуміти під «є лінійна залежність», якщо коефіцієнт кореляції по модулю більше 0,8)

Функція:

```
isLinearDependent<-function(x,y){
  return(abs(cor(x,y))>0.8)
}
```

Висновок.

При виконанні лабораторної роботи було розроблено моделі на основі дерев прийняття рішень. Було досліджено 3 методи: одне дерево, ансамбль

дерев та випадковий ліс. Методом з найменшою похибкою на тестових даних виявився випадковий ліс. Для виконання додаткового завдання було розглянуто кластеризацію даних. Також було розглянуто інструменти R для створення функцій і циклів.

Додаток А. Код мовою програмування R

```
# основне завдання
data<-read.csv("titanic.csv",sep=",",dec = ".")
str(data)
summary(data)
naPerc
# відсоток відсутніх значень
naPerc<-NULL
naPerc$names<-colnames(data)
naPerc$rate<-colSums(is.na(data))/nrow(data)*100
naPerc$rate[4]<-nrow(data[data[4]=="",])/nrow(data)*100
naPerc$rate[5]<-nrow(data[data[5]=="",])/nrow(data)*100
naPerc$rate[11]<-nrow(data[data[11]=="",])/nrow(data)*100
naPerc$rate[12]<-nrow(data[data[12]=="",])/nrow(data)*100
naPerc
# відсутні значення у колонках Age (20%), Cabin (77%) і Embarked (0.22%)

# для Embarked відкинемо ці рядки (лише 2 шт.)
workData <- data[data[12] != "",]

# для Age заповнимо середнім значенням по такому ж класу каюти
Mean<-data%>%select(Pclass,Age)%>%group_by(Pclass)%>%summarise(mean(Age,na.rm = TRUE))
workData$Age[is.na(workData$Age)&workData$Pclass==1]<-Mean[[2]][1]
workData$Age[is.na(workData$Age)&workData$Pclass==2]<-Mean[[2]][2]
workData$Age[is.na(workData$Age)&workData$Pclass==3]<-Mean[[2]][3]

# для Cabin відкинемо колонку
workData$Cabin<-NULL

# також відкинемо ім'я, квиток, ID
workData$Name <- NULL
workData$Ticket <- NULL
workData$PassengerId <- NULL
```

```
# визначення навчальної і тестової вибірки
trainSize = nrow(workData)/3*2 # дві третини навчальна вибірка
set.seed(1)
index = sample( seq_len(nrow(workData)), size = trainSize ) # відібрати випадкові індекси рядків
# поділ вибірки на навчальну та тестову
train = workData[index , ]
test = workData[-index , ]

# побудова 3 різних моделей

# одне дерево
tree<-rpart(Survived ~ .,data = train, method = "class", control=rpart.control(minbucket = 2))
tree
# намалювати дерево
fancyRpartPlot(tree)
# перевірка моделі
testPred<-predict(tree, newdata = test, type="vector")
sqrt(sum((testPred - test$Survived)^2))/length(testPred)/mean(data$Survived)  # похибка

# ансамбль дерев
numtrees <- 300
res <- numeric(numtrees)   # масив відповідає за похибку по вказаній кількості дерев
prd <- numeric(nrow(test)) # середні результати, по яким і дивимось похибки
for(i in 1:numtrees){
  # випадково виберемо 80% рядків та стовпчиків з множини даних
  x <- runif(nrow(train))>0.2;
  y <- runif(ncol(train))>0.2;
  # обовязково включимо Survived , бо для нього будуємо модель
  y[1] <- TRUE
  traindata <- train[x,y]
  # генеруємо повне дерево
  atree <- rpart(Survived ~ ., traindata, control=rpart.control(cp=.0))
  # усереднюємо передбачення з усіма попередніми деревами
  prd <- prd + predict(atree, test)
```

```
  predictions <- prd / i
  # оцінюємо похибку
  res[i] <- sqrt(sum((predictions - test$Survived)^2))/length(predictions)/mean(data$Survived)
}
plot(res,type="l")
res[numtrees]

# випадковий ліс
library(randomForest)
randForest <- randomForest(Survived ~ ., train)
predictions <- predict(randForest, test)
print(sqrt(sum((as.integer(predictions) - as.integer(test$Survived))^2))/length(predictions))

# одне дерево      0.151
# ансамбль дерев   0.053
# випадковий ліс   0.036



# додаткове завдання

# імпорт даних з файлу Data2.csv і виправлення даних (з КП3/КП4)
data2 <- read.csv("Data2.csv", sep=";", header = TRUE, dec = ',')
str(data2)
# перейменувати колонку
names(data2)[names(data2) == "Populatiion"] <- "Population"
# від'ємні значення взяти по модулю
data2$GDP.per.capita <- abs(data2$GDP.per.capita)
data2$Area <- abs(data2$Area)
# замінити пропущені значення на середні
data2$GDP.per.capita[is.na(data2$GDP.per.capita)] <- mean(data2$GDP.per.capita, na.rm = TRUE)
data2$Population[is.na(data2$Population)] <- mean(data2$Population, na.rm = TRUE)
data2$CO2.emission[is.na(data2$CO2.emission)] <- mean(data2$CO2.emission, na.rm = TRUE)
str(data2)
```

```
summary(data2)

# 1. визначити, який регіон домінує в кластерах по ВВП на душу населення та щільності населення

data2$Density<-data2$Population/data2$Area  # створимо нову колонку зі значенням щільності населення
workData<-select(data2,2,3,7)    # вибрано лише регіон, ВВП на душу населення і щільність населення

# Метод k-середніх

# оцінюємо моделі з різною кількістю кластерів по tot.withinss
kbest<-c(1:10)
for (i in 1:10) {
  kres <- kmeans(workData[,2:3],i,nstart=20)   # кількість кластерів перебираємо від 1 до 10
  kbest[i]<-kres$tot.withinss
}
plot(kbest)

# найкраща модель з 5 класами
kres <- kmeans(workData[,2:3],5,nstart=20)
workData$cluster<-kres$cluster

ggplot(data=workData,aes(x=GDP.per.capita, y=Density))+
  geom_point(col=workData$cluster)

workData$Population<-data2$Population
workData$Area<-data2$Area

# Групування даних за регіоном та кластером
regionClusters <- workData %>%
  group_by(Region, cluster) %>%
  summarize(GDP.per.capita = sum(GDP.per.capita*Population)/sum(Population), density = sum(Population)/sum(Area)) %>%
  ungroup()
```

```
# Знаходження регіону з найбільшою середньою вартістю ВВП на душу населення та щільністю населення в кластерах
topRegion <- regionClusters %>%
  group_by(cluster) %>%
  slice(which.max(GDP.per.capita * density)) %>%
  select(-cluster)

# Виведення результатів
cat("Регіон, що домінує в кластерах:\n")
print(topRegion)



# 2. вивести частотні гістограми всіх показників файла Data2.csv, використовуючи цикл
for (i in 3:6){
  hist(data2[,i],main=colnames(data2)[i])
}

# 3. створити функцію, яка на вхід отримує два набори даних, перевіряє чи є лінійна залежність та виводить True чи False
# будемо розуміти під «є лінійна залежність», якщо коефіцієнт кореляції по модулю більше 0,8
isLinearDependent<-function(x,y){
  return(abs(cor(x,y))>0.8)
}
cor(data2$GDP.per.capita,data2$CO2.emission)
isLinearDependent(data2$GDP.per.capita,data2$CO2.emission)
```