МІНІСТЕРСТВО НАУКИ І ОСВІТИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ № 1 ЗА ТЕМОЮ:

**Задачі класифікації в середовищі аналізу даних RStudio**

Група \_\_\_\_\_11\_\_\_\_\_\_

Курс \_\_\_\_\_1\_\_\_\_\_\_\_\_\_

Студент (ка) \_\_\_\_Супруненко М. І. \_\_\_\_

Дата оформлення \_\_\_\_\_\_21.02.2025\_\_\_\_\_\_\_

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МЕТА РОБОТИ: Метою лабораторної роботи є формування професійних вмінь та навичок щодо використання інструментів та методів класифікації даних з використанням методу K-Nearest Neighbor (KNN) в середовищі R, вміння застосовувати отримані знання на практиці в практичних задачах інтелектуального аналізу даних.

ТЕОРІЯ:

K-Nearest Neighbor або KNN — це нелінійний алгоритм класифікації з вчителем. KNN у мові програмування R є непараметричним алгоритмом, тобто він не робить жодних припущень щодо базових даних або їх розподілу.

KNN в R є одним із найпростіших і найбільш широко використовуваних алгоритмів, який залежить від його значення k (сусідів) і знаходить застосування в багатьох галузях, від фінансової індустрії до галузі охорони здоров’я тощо.

В алгоритмі KNN в R K визначає кількість сусідів, а його алгоритм виглядає наступним чином:

1. Виберіть число K сусідів.
2. Візьміть K найближчих сусідів до невідомої точки даних відповідно до визначеної відстані.
3. Серед K-сусідів підрахуйте кількість точок даних у кожній категорії.
4. Призначте нову (розглянуту) точку даних до категорії, з найбільшої кількістю сусідів.

ХІД РОБОТИ:

1. Завантажте навчальний обраний вами невеликий навчальний набір даних та відповідні бібліотеки.
2. Розділіть дані на тренувальний та тестовий набори в співвідношенні (70% на 30%).
3. Нормалізуйте дані (це може бути Z-score).
4. Побудуйте K-model (knn()). Зробіть дослідження та опишіть параметри налаштування функції.
5. Побудуйте матрицю оцінок (Confusiin Matrix). Зробіть висновки.
6. Зробіть дослідження щодо точності прогнозування з різними числами *K (кількості сусідів)*
7. Створіть графік точності прогнозування в залежності від К.
8. Оформіть роботу, підготуйтеся до захисту.

ВХІДНІ ДАНІ:

Набір даних Iris (використовується для демонстрації методів та принципів роботи в ході лабораторної роботи) складається з 50 зразків кожного з 3 видів Iris (Iris setosa, Iris virginica, Iris versicolor) і багатовимірного набору даних, представленого британським статистиком і біологом Рональдом Фішером у його статті 1936 року «Використання кількох вимірювань у таксономічних проблемах». Для кожного зразка було виміряно чотири ознаки, тобто довжину та ширину чашолистків і пелюсток, і на основі комбінації цих чотирьох ознак Фішер розробив лінійну дискримінантну модель, щоб відрізнити види один від одного.

Необхідні бібліотеки:

# Installing Packages

install.packages("e1071")

install.packages("caTools")

install.packages("class")

# Loading package

library(e1071)

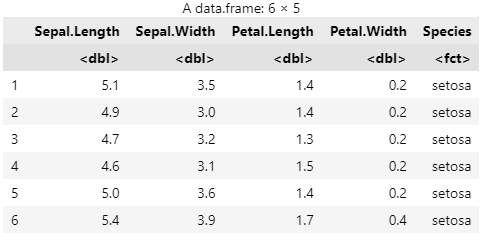
library(caTools)

library(class)

РЕЗУЛЬТАТ ВИКОНАННЯ:

1. Імпортування необхідних бібліотек та завантаження набору даних: Для виконання лабораторної роботи на R у Jupyter Notebook використовуємо пакет class для KNN.

|  |
| --- |
| # Встановлення та завантаження необхідних бібліотек  install.packages("e1071")  # Якщо пакет не встановлений  install.packages("caTools")  # Для розділення на тренувальний і тестовий набір  install.packages("class")  # Для алгоритму KNN  library(e1071)  library(caTools)  library(class)  # Завантаження набору даних Iris  data(iris)  head(iris)  # Переглянути перші кілька рядків |



2. Розділення даних на тренувальний та тестовий набори (70%/30%):

|  |
| --- |
| set.seed(123)  # Фіксуємо генератор випадкових чисел для відтворюваності  split <- sample.split(iris$Species, SplitRatio = 0.7)  # Розподіл даних  train\_set <- subset(iris, split == TRUE)  # Тренувальний набір  test\_set <- subset(iris, split == FALSE)  # Тестовий набір  # Виділення ознак та міток класів  train\_X <- train\_set[, -5]  # Все, крім останньої колонки  train\_Y <- train\_set[, 5]   # Остання колонка (Species)  test\_X <- test\_set[, -5]  test\_Y <- test\_set[, 5] |

3. Нормалізація даних (для покращення роботи алгоритму KNN):

|  |
| --- |
| # Нормалізація даних  normalize <- function(x) {    return ((x - min(x)) / (max(x) - min(x)))  }  train\_X <- as.data.frame(lapply(train\_X, normalize))  test\_X <- as.data.frame(lapply(test\_X, normalize)) |

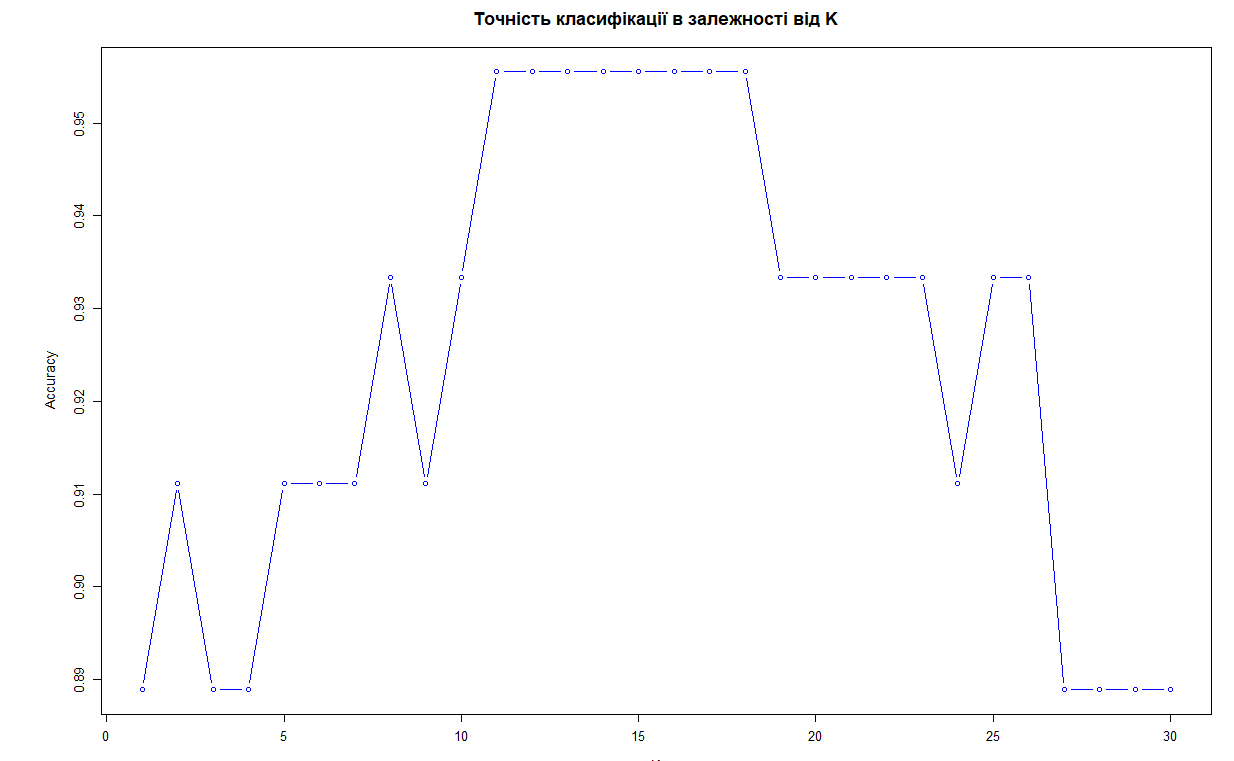
4. Побудова та оцінка моделі KNN:

|  |
| --- |
| # Вибір кількості сусідів (k)  k <- 15  predictions <- knn(train = train\_X, test = test\_X, cl = train\_Y, k = k)  # Створення матриці сплутаності (Confusion Matrix)  conf\_matrix <- table(Predicted = predictions, Actual = test\_Y)  print(conf\_matrix)  # Обчислення точності  accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)  print(paste("Accuracy:", round(accuracy \* 100, 2), "%")) |

5. Дослідження точності з різними значеннями K:

|  |
| --- |
| accuracy\_values <- c()  for (k in 1:30) {    predictions <- knn(train = train\_X, test = test\_X, cl = train\_Y, k = k)    conf\_matrix <- table(Predicted = predictions, Actual = test\_Y)    accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)    accuracy\_values <- c(accuracy\_values, accuracy)  }  # Побудова графіка точності залежно від K  # Зберігаємо графік у файл PNG з більшими розмірами  png("accuracy\_plot.png", width = 1200, height = 800)  plot(1:30, accuracy\_values, type = "b", col = "blue", xlab = "K", ylab = "Accuracy",       main = "Точність класифікації в залежності від K", cex.main = 1.5, cex.lab = 1.2, cex.axis = 1.1)  dev.off()  # Закриваємо файл |

СХЕМА (МАЛЮНОК) РОБОТИ (У РАЗІ НАЯВНОСТІ): Наведіть схему (малюнок) або код програми для реалізації дослідження.



ВИСНОВКИ:

· Вибір оптимального значення K є важливим, оскільки різні значення можуть дати різні результати точності.

· Коли K занадто мале, модель може бути дуже чутливою до шуму, а коли K занадто велике, модель може бути занадто гладкою і менш точними.

· Рекомендується проводити тестування на різних значеннях K, щоб знайти оптимальний баланс.