**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

**Лабораторна робота №**3

з дисципліни «Інтелектуальний аналіз та візуалізація даних»

тема «КОРЕЛЯЦІЙНО-РЕГРЕСІЙНИЙ АНАЛІЗ ДАНИХ»

**Варіант №8**

Виконала студентка

групи КН-21

Іванова А.О.

Перевірив(-ла):

МІнаєва Ю.І.

**Київ – 2025**

**Мета роботи:** Кореляційних аналіз даних. Регресійний аналіз даних. Мова Python (R).

# Імпорт бібліотек

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import scipy.stats as stats

from sklearn.datasets import load\_wine

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score, mean\_squared\_error

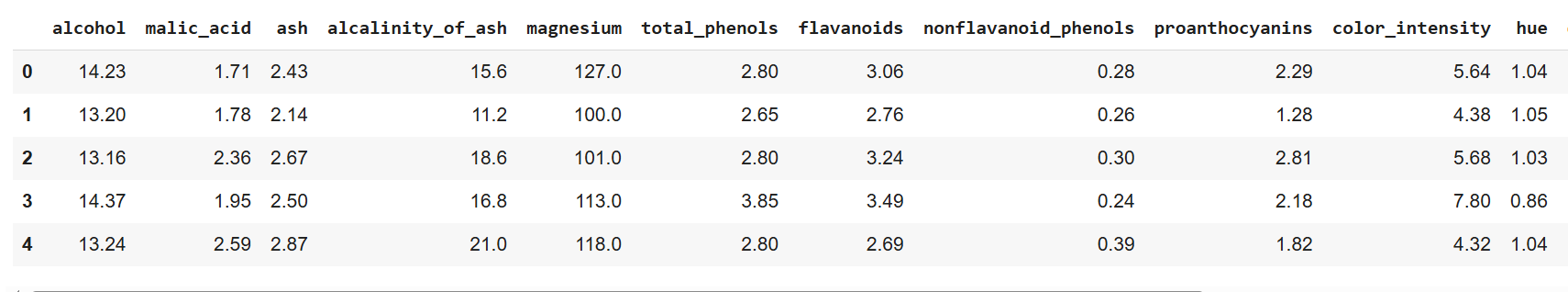
from google.colab import files

wine = load\_wine()

df\_wine = pd.DataFrame(data=wine.data, columns=wine.feature\_names)

df\_wine['target'] = wine.target

df\_wine.head()



# Вибір ознаки

feature = 'alcohol'

# Тест Шапіро-Уілка

stat, p\_value = stats.shapiro(df\_wine[feature])

print(f"Шапіро-Уілк тест для '{feature}':")

print(f"Статистика: {stat:.4f}")

print(f"p-значення: {p\_value:.4f}")

# Інтерпретація

alpha = 0.05

if p\_value > alpha:

print("Дані мають нормальний розподіл")

else:

print("Дані не мають нормального розподілу")

Шапіро-Уілк тест для 'alcohol':

Статистика: 0.9818

p-значення: 0.0200

Дані не мають нормального розподілу

# Вибір ознаки

feature = 'malic\_acid'

# Тест Шапіро-Уілка

stat, p\_value = stats.shapiro(df\_wine[feature])

print(f"Шапіро-Уілк тест для '{feature}':")

print(f"Статистика: {stat:.4f}")

print(f"p-значення: {p\_value:.2e}")

# Інтерпретація

alpha = 0.05

if p\_value > alpha:

print("Дані мають нормальний розподіл")

else:

print("Дані не мають нормального розподілу")

Шапіро-Уілк тест для 'malic\_acid':

Статистика: 0.8888

p-значення: 2.95e-10

Дані не мають нормального розподілу

# Univariate Analysis of Variance (ANOVA) для однієї ознаки по класам

categorical\_variable = 'target'

continuous\_variable = 'alcohol'

groups = [df\_wine[df\_wine[categorical\_variable] == category][continuous\_variable]

for category in df\_wine[categorical\_variable].unique()]

f\_statistic, p\_value = stats.f\_oneway(\*groups)

alpha = 0.05

if p\_value < alpha:

print("Є суттеві відмінності")

else:

print("Відсутні суттеві відмінності між групами")

Є суттеві відмінності

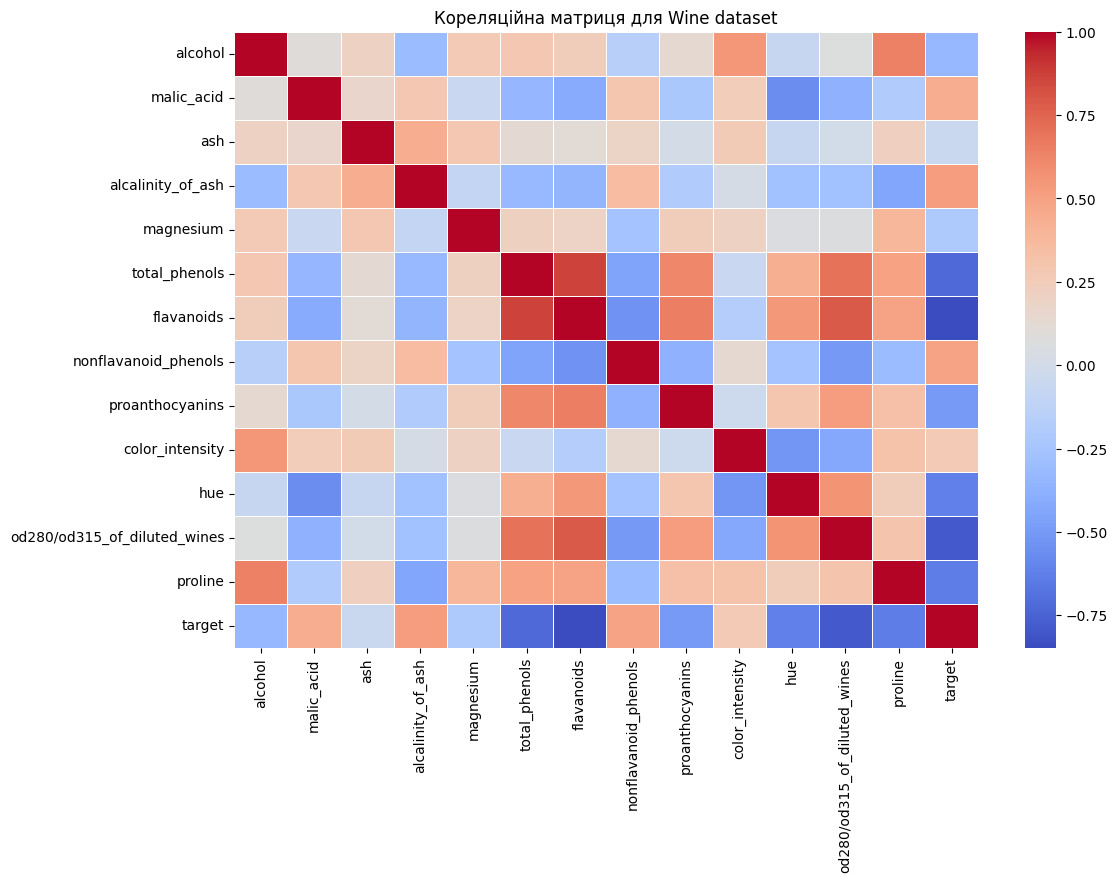
# Кореляційна матриця

plt.figure(figsize=(12, 8))

sns.heatmap(df\_wine.corr(), cmap="coolwarm", annot=False, linewidths=0.5)

plt.title("Кореляційна матриця для Wine dataset")

plt.show()



# Кореляція Спірмана для перших двох ознак

feature\_x = df\_wine.columns[0] # alcohol

feature\_y = df\_wine.columns[1] # malic\_acid

spearman\_corr, \_ = stats.spearmanr(df\_wine[feature\_x], df\_wine[feature\_y])

print(f"Кореляція Спірмана ({feature\_x} vs {feature\_y}): {spearman\_corr:.4f}")

Кореляція Спірмана (alcohol vs malic\_acid): 0.1404

# Лінійна регресія alcohol → malic\_acid

X = df\_wine[feature\_x].values.reshape(-1, 1)

y = df\_wine[feature\_y].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

r2 = r2\_score(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

F\_stat, p\_val = stats.f\_oneway(y\_test, y\_pred)

print("\*Оцінка моделі:")

print(f"Коєфіцієнт детермінації - R²: {r2:.4f}")

print(f"Середньокрадратична похибка - MSE: {mse:.4f}")

print(f"F-критерій Фішера: {F\_stat:.4f}, p-значення: {p\_val:.4f}")

\* Оцінка моделі:

Коєфіцієнт детермінації - R²: -0.0046

Середньокрадратична похибка - MSE: 0.9883

- F-критерій Фішера: 1.3078, p-значення: 0.2567

# Візуалізація регресії

plt.figure(figsize=(8, 6))

plt.scatter(X\_test, y\_test, color='green', label='Фактичні значення')

plt.scatter(X\_test, y\_pred, color='red', marker='s', label='Прогнозовані значення')

x\_line = np.linspace(min(X\_test), max(X\_test), 100).reshape(-1, 1)

y\_line = model.predict(x\_line)

plt.plot(x\_line, y\_line, color='black', label='Лінія регресії')

for i in range(len(y\_test)):

plt.plot([X\_test[i], X\_test[i]], [y\_test[i], y\_pred[i]], color='gray', linestyle='dotted')

plt.xlabel(feature\_x)

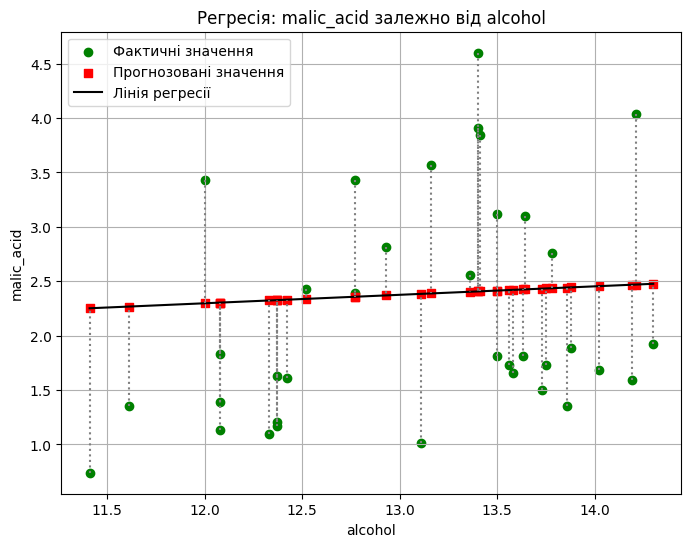
plt.ylabel(feature\_y)

plt.title(f"Регресія: {feature\_y} залежно від {feature\_x}")

plt.legend()

plt.grid(True)

plt.show()

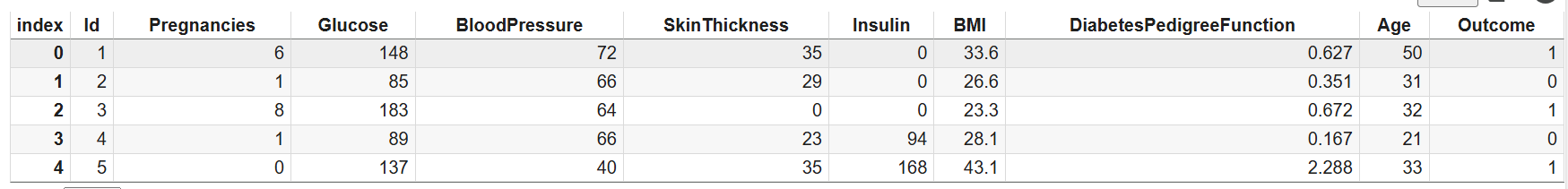


# Завантаження датасету Healthcare Diabetse

uploaded = files.upload()

df\_healthcare = pd.read\_csv("Healthcare-Diabetes.csv")

df\_healthcare.head()



# Вибір ознаки

feature = 'Glucose'

# Тест Шапіро-Уілка

stat, p\_value = stats.shapiro(df\_healthcare[feature])

print(f"Шапіро-Уілк тест для '{feature}':")

print(f"Статистика: {stat:.4f}")

print(f"p-значення: {p\_value:.2e}")

# Інтерпретація

alpha = 0.05

if p\_value > alpha:

print("Дані мають нормальний розподіл (не відхиляємо H₀)")

else:

print("Дані не мають нормального розподілу (відхиляємо H₀)")

Шапіро-Уілк тест для 'Glucose':

Статистика: 0.9702

p-значення: 1.54e-23

Дані не мають нормального розподілу (відхиляємо H₀)

# Вибір ознаки

feature = 'DiabetesPedigreeFunction'

# Тест Шапіро-Уілка

stat, p\_value = stats.shapiro(df\_healthcare[feature])

print(f"Шапіро-Уілк тест для '{feature}':")

print(f"Статистика: {stat:.4f}")

print(f"p-значення: {p\_value:.4f}")

# Інтерпретація

alpha = 0.05

if p\_value > alpha:

print("Дані мають нормальний розподіл")

else:

print("Дані не мають нормального розподілу")

Шапіро-Уілк тест для 'DiabetesPedigreeFunction':

Статистика: 0.8446

p-значення: 0.0000

Дані не мають нормального розподілу

# Univariate Analysis of Variance (ANOVA) для однієї ознаки по класам

categorical\_variable = 'Outcome'

continuous\_variable = 'Glucose'

groups = [df\_healthcare[df\_healthcare[categorical\_variable] == category][continuous\_variable]

for category in df\_healthcare[categorical\_variable].unique()]

f\_statistic, p\_value = stats.f\_oneway(\*groups)

alpha = 0.05

if p\_value < alpha:

print("Є суттеві відмінності")

else:

print("Відсутні суттеві відмінності між групами")

Є суттеві відмінності

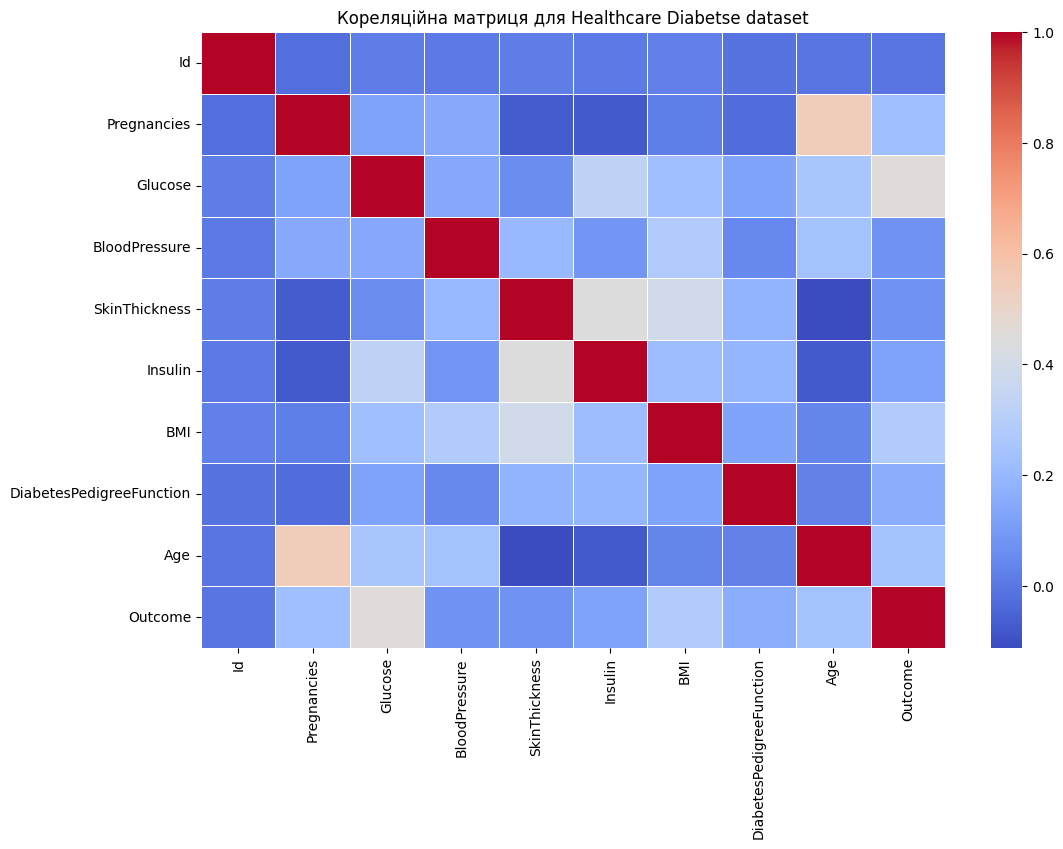
# Кореляційна матриця

plt.figure(figsize=(12, 8))

sns.heatmap(df\_healthcare.corr(), cmap="coolwarm", annot=False, linewidths=0.5)

plt.title("Кореляційна матриця для Healthcare Diabetse dataset")

plt.show()



# Кореляція Спірмана для перших двох ознак

feature\_x = df\_healthcare.columns[2] # Glucose

feature\_y = df\_healthcare.columns[7] # DiabetesPedigreeFunction

spearman\_corr, \_ = stats.spearmanr(df\_healthcare[feature\_x], df\_healthcare[feature\_y])

print(f"Кореляція Спірмана ({feature\_x} vs {feature\_y}): {spearman\_corr:.4f}")

Кореляція Спірмана (Glucose vs DiabetesPedigreeFunction): 0.0773

# Лінійна регресія Glucose → DiabetesPedigreeFunction

X = df\_healthcare[feature\_x].values.reshape(-1, 1)

y = df\_healthcare[feature\_y].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

r2 = r2\_score(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

F\_stat, p\_val = stats.f\_oneway(y\_test, y\_pred)

print("\*Оцінка моделі:")

print(f"Коєфіцієнт детермінації - R²: {r2:.4f}")

print(f"Середньокрадратична похибка - MSE: {mse:.4f}")

print(f"F-критерій Фішера: {F\_stat:.4f}, p-значення: {p\_val:.4f}")

\*Оцінка моделі:

Коєфіцієнт детермінації - R²: 0.0117

Середньокрадратична похибка - MSE: 0.0964

F-критерій Фішера: 0.8876, p-значення: 0.3463

# Візуалізація регресії

plt.figure(figsize=(8, 6))

plt.scatter(X\_test, y\_test, color='green', label='Фактичні значення')

plt.scatter(X\_test, y\_pred, color='red', marker='s', label='Прогнозовані значення')

x\_line = np.linspace(min(X\_test), max(X\_test), 100).reshape(-1, 1)

y\_line = model.predict(x\_line)

plt.plot(x\_line, y\_line, color='black', label='Лінія регресії')

for i in range(len(y\_test)):

plt.plot([X\_test[i], X\_test[i]], [y\_test[i], y\_pred[i]], color='gray', linestyle='dotted')

plt.xlabel(feature\_x)

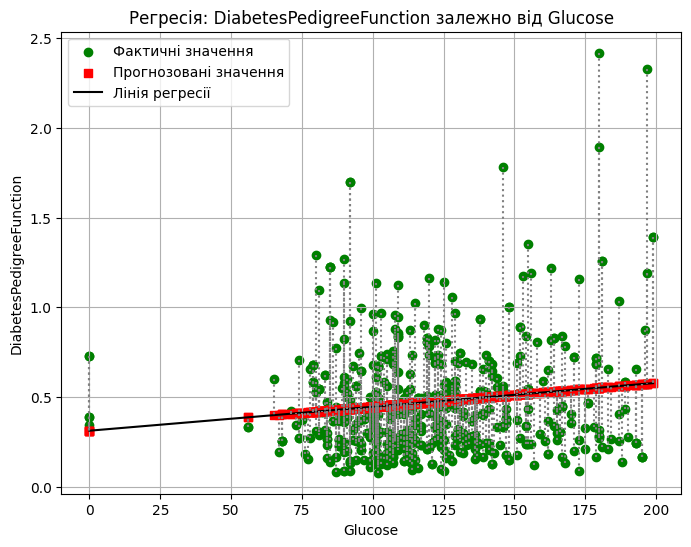
plt.ylabel(feature\_y)

plt.title(f"Регресія: {feature\_y} залежно від {feature\_x}")

plt.legend()

plt.grid(True)

plt.show()



**Висновок:** лабораторній роботі було виконано кореляційний та регресійний аналіз.