**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра інтелектуальних технологій**

**Лабораторна робота №2**

з дисципліни «Інтелектуальний аналіз даних»

Тема роботи: «Регресійний аналіз даних. Логістична регресія»

**Варіант №5**

Виконала:

студентка групи АнД-31

Радоманова С. П.

Перевірив:

Чолишкіна О.Г.

**Київ – 2025**

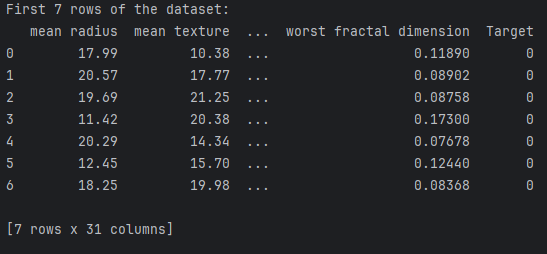
Мета роботи:

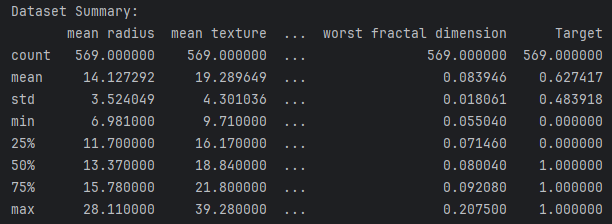
Регресійний аналіз даних. Логістична регресія. Мова Python.

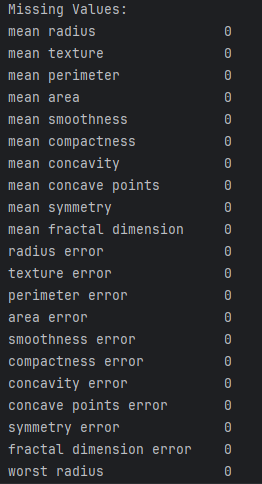
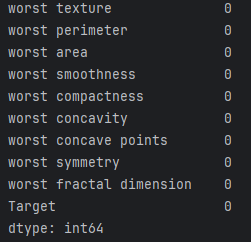
Завдання і порядок виконання роботи:

1. Виконати розвідувальний аналіз даних для власного датасету. (Можна обрати датасет з відкритих джерел).

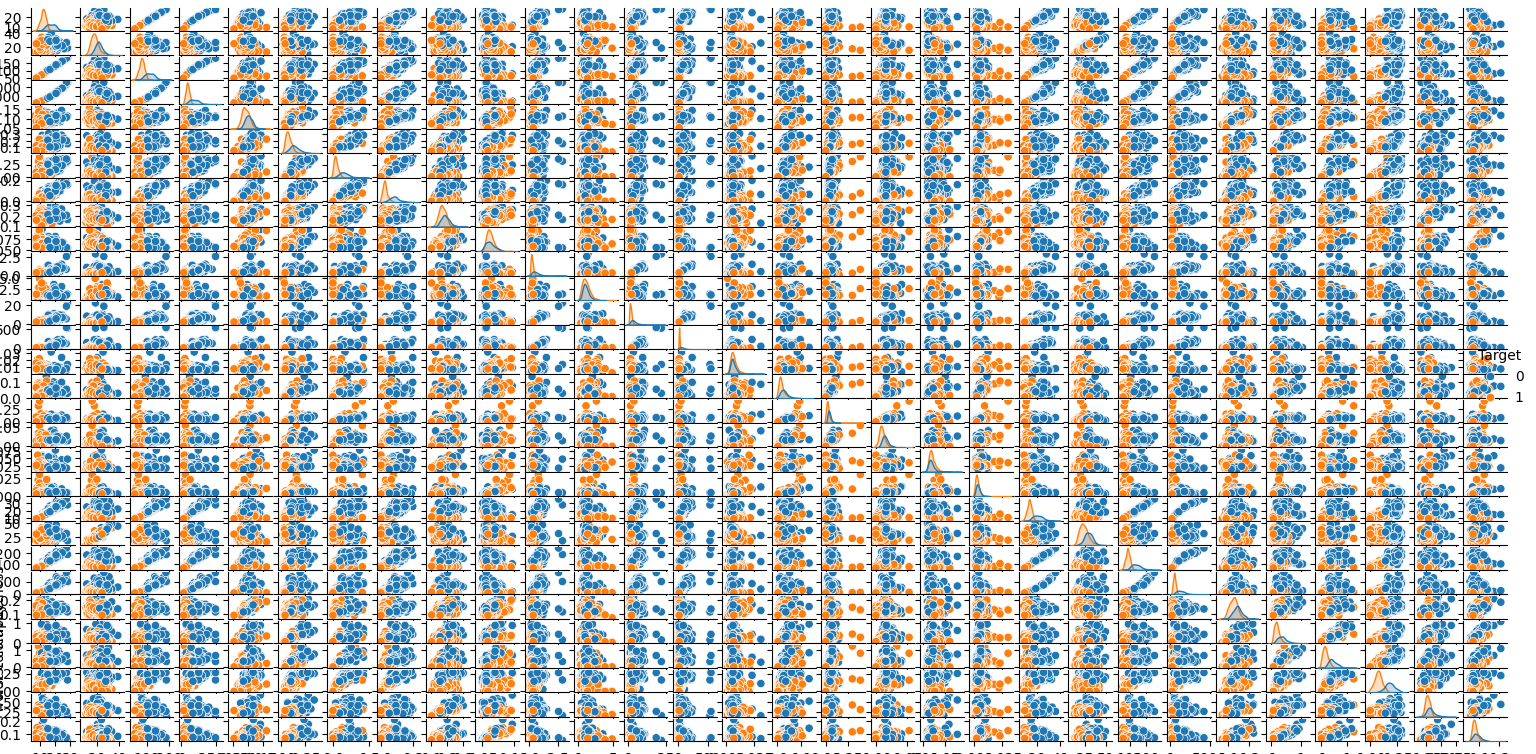
Був обраний датасет load\_breast\_cancer. В датасеті вже наявна таргетна змінна: 0 = злоякісна пухлина, 1 = доброякісна пухлина

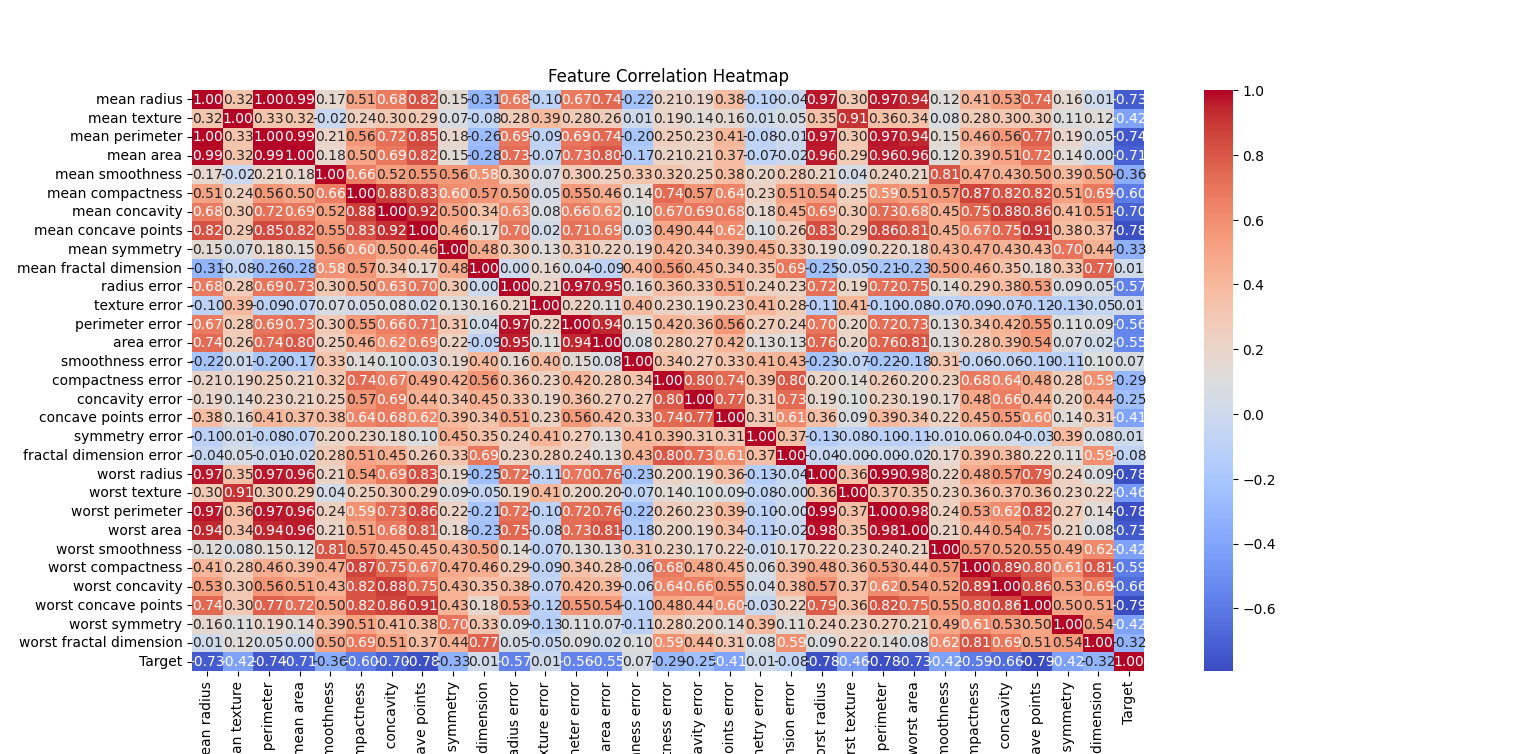




Візуалізація:





1. Виконати регресійний аналіз даних . Побудувати модель логістичної регресії.

accuracy - скільки разів модель вгадала правильний клас відносно всіх прикладів

ROC-AUC показує якість передбачення ймовірностей де 1.0 - ідеальна модель, 0.5 - випадкова.

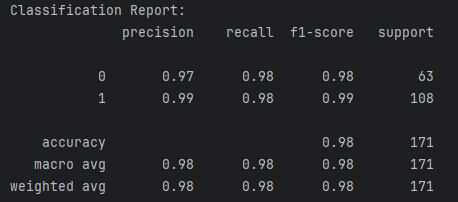


precision – точність: з усіх передбачених як «позитивні», яка частка правильна.

recall – повнота: з усіх справжніх «позитивних», яку частку модель знайшла.

f1-score – гармонічне середнє між precision і recall

support - кількість об’єктів у датасеті, що належать до цього класу

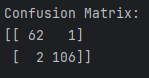


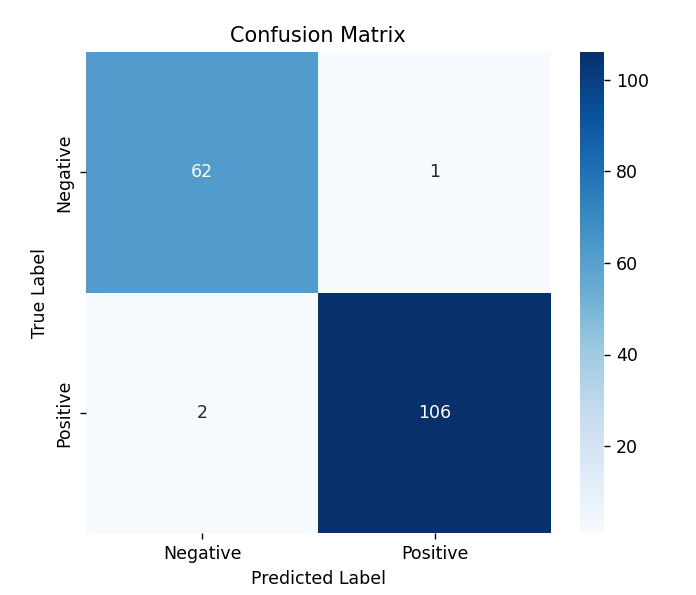
TN -True Negative (правильно передбачено 0)

FP - False Positive (0 передбачено як 1)

FN - False Negative (1 передбачено як 0)

TP - True Positive (правильно передбачено 1)





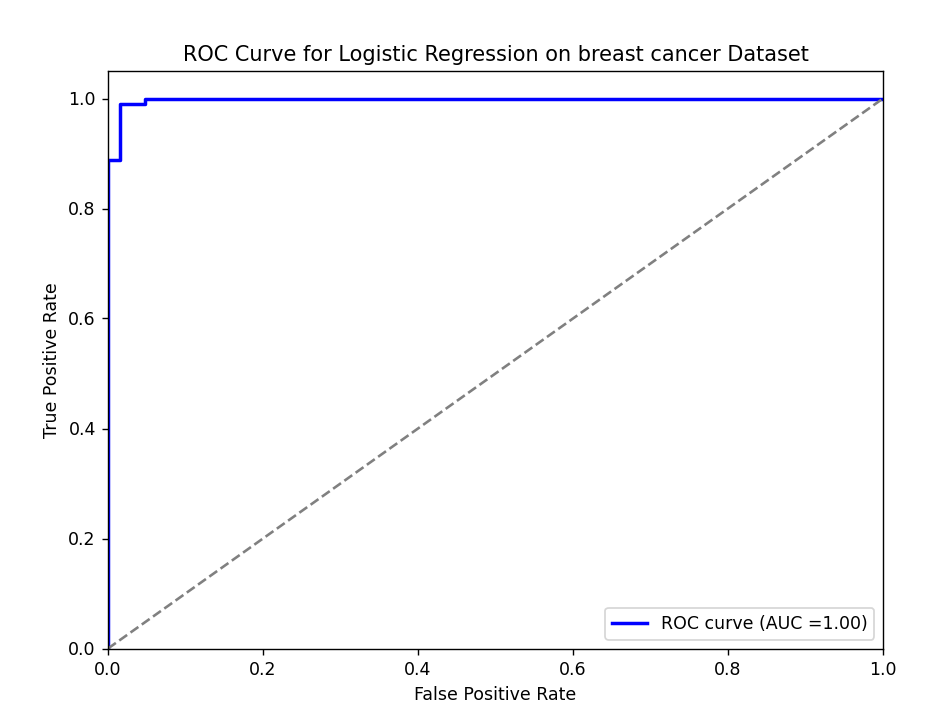
Побудова ROC-кривої

Крива показує, як модель відокремлює класи при різних порогах ймовірності

**FPR** - частка помилкових позитивних передбачень

**TPR** - частка правильних позитивних передбачень

Чим вище крива – тим краща модель



4. Виконати аналіз отриманих результатів, їх змістовну інтерпретацію.

Отримані результати моделі логістичної регресії показали високу точність класифікації пухлин – значення accuracy близьке до 0.98. Це означає, що модель правильно класифікує приблизно 98% усіх прикладів у вибірці. Високе значення ROC-AUC (понад 0.97) свідчить, що модель добре розрізняє два класи – злоякісні й доброякісні пухлини – при різних порогах ймовірності, тобто має стійку здатність відокремлювати групи навіть при зміні умов класифікації.

Показники precision та recall для обох класів вказують на те, що модель не лише робить мало помилкових позитивних передбачень (висока точність), але й знаходить більшість справжніх позитивних випадків (висока повнота). Це особливо важливо для медичних даних, оскільки пропуск злоякісної пухлини (False Negative) може мати серйозні наслідки. Значення f1-score, як гармонічного середнього між точністю та повнотою, підтверджує баланс моделі між цими двома характеристиками.

Аналіз матриці помилок (TN, FP, FN, TP) показує, що кількість хибнопозитивних і хибнонегативних передбачень невелика у порівнянні з правильно класифікованими об’єктами. Це означає, що модель є надійною у практичному використанні, а ризик помилкових рішень мінімальний.

Побудована ROC-крива має вигнуту форму, що лежить значно вище діагоналі випадкових передбачень. Це підтверджує, що модель може бути використана для прийняття рішень навіть при зміні порогів відсікання ймовірностей.

Узагальнюючи, отримані результати свідчать про те, що логістична регресія є адекватною й ефективною для даного завдання. Модель демонструє високу точність та стабільність, добре відокремлює класи й може бути використана для підтримки діагностики пухлин молочної залози.

Лістинг:

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
from sklearn import datasets  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import StandardScaler  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import accuracy\_score, roc\_auc\_score, roc\_curve,classification\_report, confusion\_matrix  
  
X, y = datasets.load\_breast\_cancer(return\_X\_y=True)  
feature\_names = datasets.load\_breast\_cancer().feature\_names  
  
  
df = pd.DataFrame(X, columns=feature\_names)  
df['Target'] = y  
  
print("First 7 rows of the dataset:")  
print(df.head(7))  
  
print("\nDataset Summary:")  
print(df.describe())  
  
print("\nMissing Values:")  
print(df.isnull().sum())  
  
# sns.pairplot(df, hue='Target', diag\_kind='kde')  
# plt.show()  
  
  
# plt.figure(figsize=(10, 8))  
# sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f')  
# plt.title('Feature Correlation Heatmap')  
# plt.show()  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,  
test\_size=0.3, random\_state=42)  
  
scaler = StandardScaler()  
X\_train = scaler.fit\_transform(X\_train)  
X\_test = scaler.transform(X\_test)  
  
model = LogisticRegression()  
model.fit(X\_train, y\_train)  
  
y\_probs = model.predict\_proba(X\_test)[:, 1]  
  
y\_pred = model.predict(X\_test)  
accuracy = accuracy\_score(y\_test, y\_pred)  
print(f'Accuracy: {accuracy:.4f}')  
  
roc\_auc = roc\_auc\_score(y\_test, y\_probs)  
print(f'ROC-AUC Score: {roc\_auc:.4f}')  
  
print("Classification Report:")  
print(classification\_report(y\_test, y\_pred))  
  
conf\_matrix = confusion\_matrix(y\_test, y\_pred)  
print("Confusion Matrix:")  
print(conf\_matrix)  
  
  
plt.figure(figsize=(6, 5))  
sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues',  
xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])  
plt.xlabel('Predicted Label')  
plt.ylabel('True Label')  
plt.title('Confusion Matrix')  
plt.show()  
  
fpr, tpr, \_ = roc\_curve(y\_test, y\_probs)  
plt.figure(figsize=(8, 6))  
plt.plot(fpr, tpr, color='blue', lw=2, label=f'ROC curve (AUC ={roc\_auc:.2f})')  
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')  
plt.xlim([0.0, 1.0])  
plt.ylim([0.0, 1.05])  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('ROC Curve for Logistic Regression on breast cancer Dataset')  
plt.legend(loc='lower right')  
plt.show()