**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

**Лабораторна робота №**4

з дисципліни «Інтелектуальний аналіз та візуалізація даних»

тема «РЕГРЕСІЙНИЙ АНАЛІЗ ДАНИХ. ЛОГІСТИЧНА РЕГРЕСІЯ»

**Варіант №8**

Виконала студентка

групи КН-21

Іванова А.О.

Перевірив(-ла):

МІнаєва Ю.І.

**Київ – 2025**

**Мета роботи:** Регресійний аналіз даних. Логістична регресія. Мова Python.

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\_wine(return\_X\_y=True)

feature\_names = datasets.load\_wine().feature\_names

median\_value = np.median(y)

y\_class = (y > median\_value).astype(int)

df = pd.DataFrame(X, columns=feature\_names)

df['Target'] = y\_class

print("First 7 rows of the dataset:")

print(df.head(7))

First 7 rows of the dataset:

alcohol malic\_acid ash alcalinity\_of\_ash magnesium total\_phenols \

0 14.23 1.71 2.43 15.6 127.0 2.80

1 13.20 1.78 2.14 11.2 100.0 2.65

2 13.16 2.36 2.67 18.6 101.0 2.80

3 14.37 1.95 2.50 16.8 113.0 3.85

4 13.24 2.59 2.87 21.0 118.0 2.80

5 14.20 1.76 2.45 15.2 112.0 3.27

6 14.39 1.87 2.45 14.6 96.0 2.50

flavanoids nonflavanoid\_phenols proanthocyanins color\_intensity hue \

0 3.06 0.28 2.29 5.64 1.04

1 2.76 0.26 1.28 4.38 1.05

2 3.24 0.30 2.81 5.68 1.03

3 3.49 0.24 2.18 7.80 0.86

4 2.69 0.39 1.82 4.32 1.04

5 3.39 0.34 1.97 6.75 1.05

6 2.52 0.30 1.98 5.25 1.02

od280/od315\_of\_diluted\_wines proline Target

0 3.92 1065.0 0

1 3.40 1050.0 0

2 3.17 1185.0 0

3 3.45 1480.0 0

4 2.93 735.0 0

5 2.85 1450.0 0

6 3.58 1290.0 0

print("\nDataset Summary:")

print(df.describe())

Dataset Summary:

alcohol malic\_acid ash alcalinity\_of\_ash magnesium \

count 178.000000 178.000000 178.000000 178.000000 178.000000

mean 13.000618 2.336348 2.366517 19.494944 99.741573

std 0.811827 1.117146 0.274344 3.339564 14.282484

min 11.030000 0.740000 1.360000 10.600000 70.000000

25% 12.362500 1.602500 2.210000 17.200000 88.000000

50% 13.050000 1.865000 2.360000 19.500000 98.000000

75% 13.677500 3.082500 2.557500 21.500000 107.000000

max 14.830000 5.800000 3.230000 30.000000 162.000000

total\_phenols flavanoids nonflavanoid\_phenols proanthocyanins \

count 178.000000 178.000000 178.000000 178.000000

mean 2.295112 2.029270 0.361854 1.590899

std 0.625851 0.998859 0.124453 0.572359

min 0.980000 0.340000 0.130000 0.410000

25% 1.742500 1.205000 0.270000 1.250000

50% 2.355000 2.135000 0.340000 1.555000

75% 2.800000 2.875000 0.437500 1.950000

max 3.880000 5.080000 0.660000 3.580000

color\_intensity hue od280/od315\_of\_diluted\_wines proline \

count 178.000000 178.000000 178.000000 178.000000

mean 5.058090 0.957449 2.611685 746.893258

std 2.318286 0.228572 0.709990 314.907474

min 1.280000 0.480000 1.270000 278.000000

25% 3.220000 0.782500 1.937500 500.500000

50% 4.690000 0.965000 2.780000 673.500000

75% 6.200000 1.120000 3.170000 985.000000

max 13.000000 1.710000 4.000000 1680.000000

Target

count 178.000000

mean 0.269663

std 0.445037

min 0.000000

25% 0.000000

50% 0.000000

75% 1.000000

max 1.000000

print("\nMissing Values:")

print(df.isnull().sum())

Missing Values:

alcohol 0

malic\_acid 0

ash 0

alcalinity\_of\_ash 0

magnesium 0

total\_phenols 0

flavanoids 0

nonflavanoid\_phenols 0

proanthocyanins 0

color\_intensity 0

hue 0

od280/od315\_of\_diluted\_wines 0

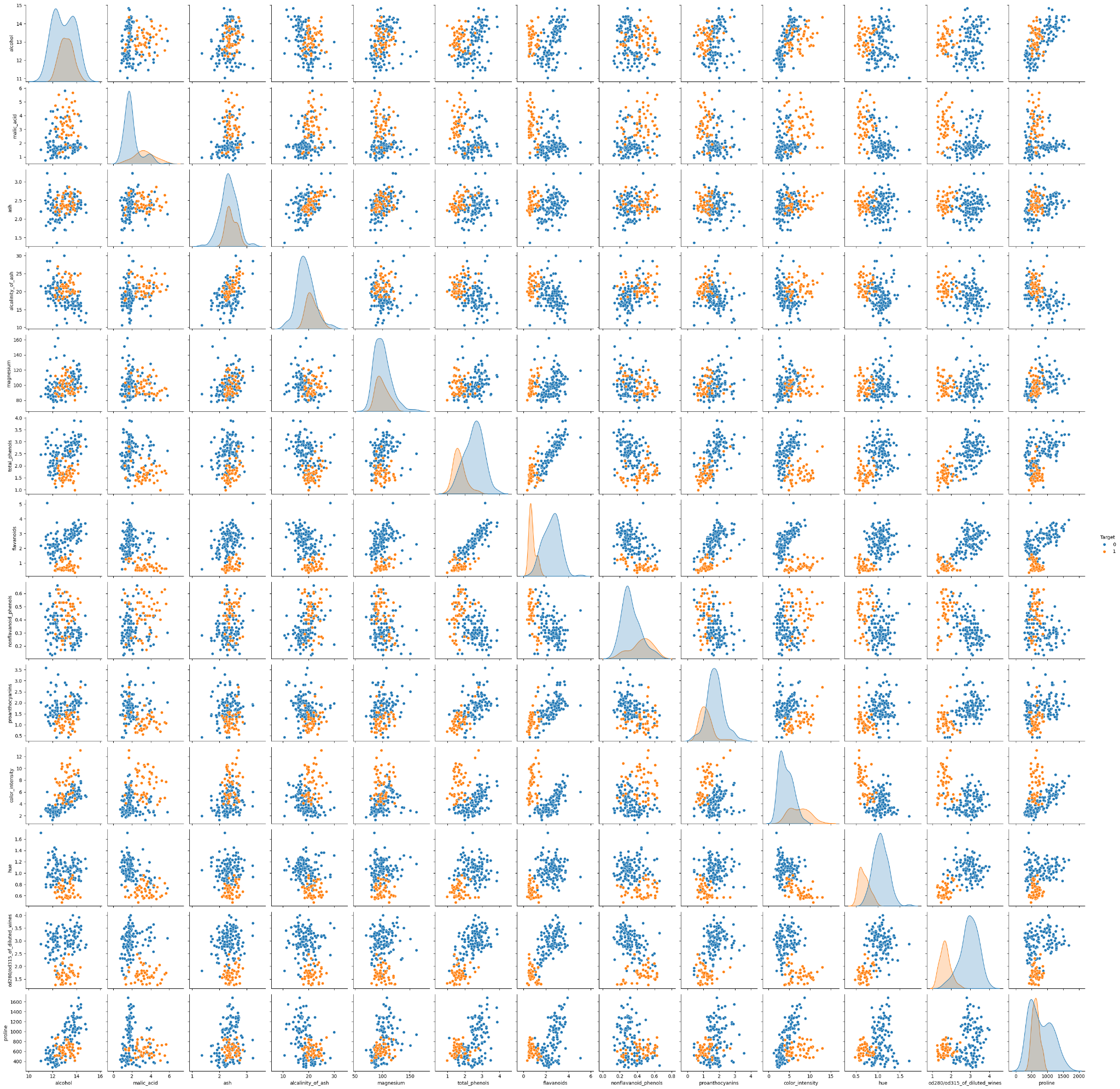
proline 0

Target 0

dtype: int64

sns.pairplot(df, hue='Target', diag\_kind='kde')

plt.show()

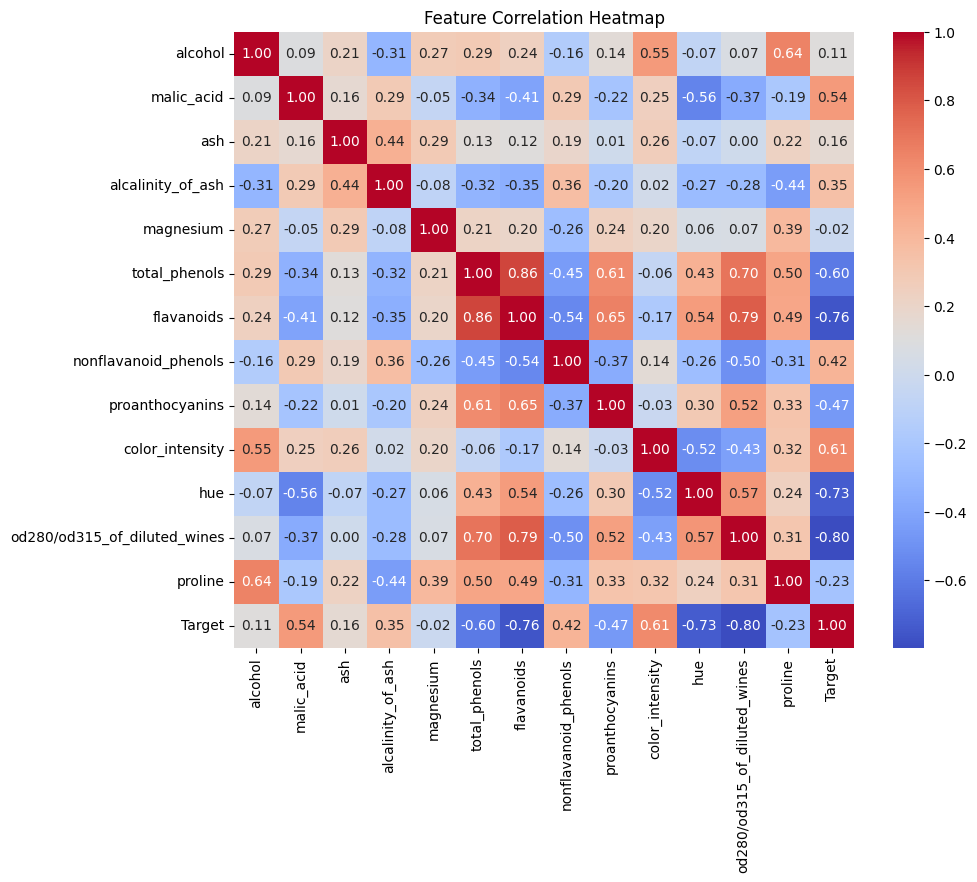


plt.figure(figsize=(10, 8))

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\_class, test\_size=0.3, random\_state=42)

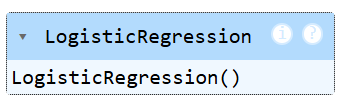
scaler = StandardScaler()

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}')

Accuracy: 1.0000

roc\_auc = roc\_auc\_score(y\_test, y\_probs)

print(f'ROC-AUC Score: {roc\_auc:.4f}')

ROC-AUC Score: 1.0000

print("Classification Report:")

print(classification\_report(y\_test, y\_pred))

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 40

1 1.00 1.00 1.00 14

accuracy 1.00 54

macro avg 1.00 1.00 1.00 54

weighted avg 1.00 1.00 1.00 54

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:")

print(conf\_matrix)

# візуалізація матриці невідповідностей

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')

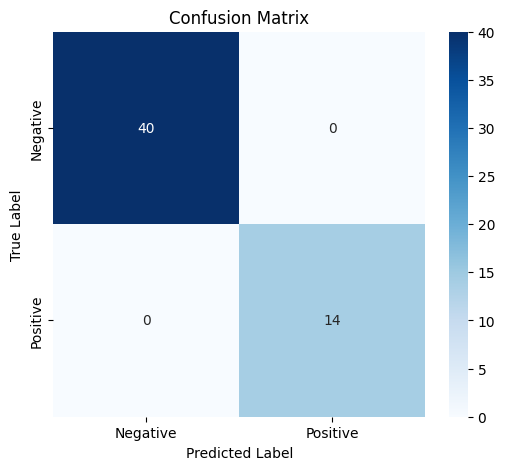
plt.title('Confusion Matrix')

plt.show()

Confusion Matrix:

[[40 0]

[ 0 14]]



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])

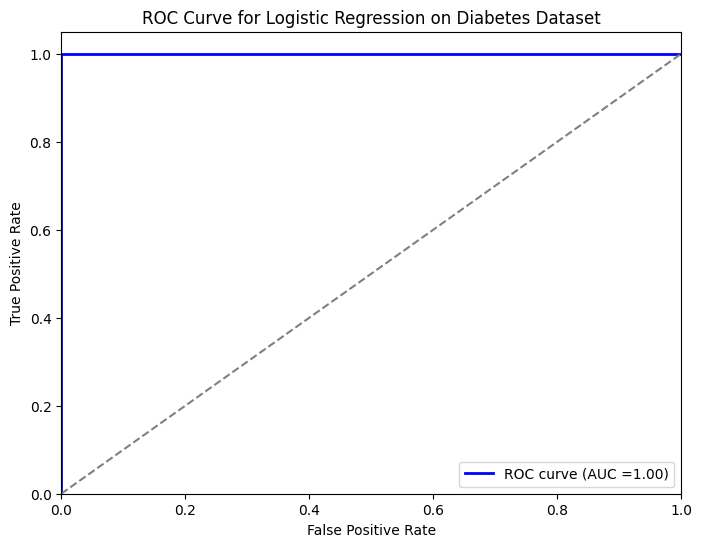
plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve for Logistic Regression on Diabetes Dataset')

plt.legend(loc='lower right')

plt.show()



**Висновок:** лабораторній роботі було виконано розвідувальний та регресійний аналіз, побудовано модель логістичної регресії.