**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

**Лабораторна робота №**5

з дисципліни «Інтелектуальний аналіз та візуалізація даних»

тема «КЛАСИФІКАЦІЯ. МЕТОД K-НАЙБЛИЖЧИХ СУСІДІВ»

**Варіант №8**

Виконала студентка

групи КН-21

Іванова А.О.

Перевірив(-ла):

МІнаєва Ю.І.

**Київ – 2025**

**Мета роботи:** Вивчення алгоритму класифікації методом k-найближчих сусідів. Мова Python.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import load\_wine

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

from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

from sklearn.feature\_selection import SelectKBest, f\_classif

from mlxtend.plotting import plot\_decision\_regions

from google.colab import files

wine = load\_wine()

# Вибір ознак. Обираємо перші 2 з датасету.

X = wine.data[:, :2]

y = wine.target

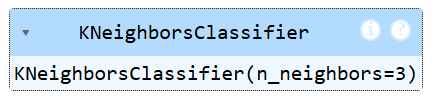
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3,

random\_state=42)

# Застосування алгоритму kNN:

knn = KNeighborsClassifier(n\_neighbors=3)

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



#Візуалізація отриманих результатів

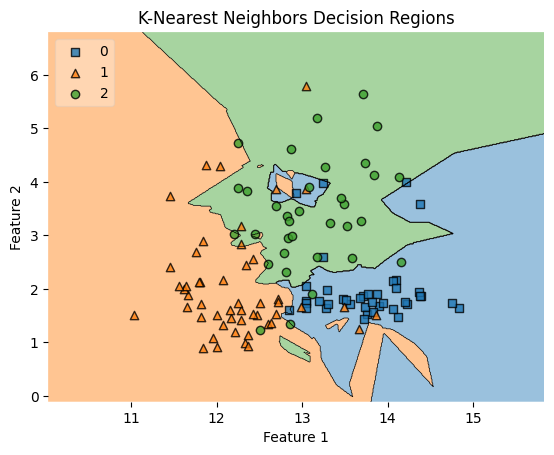
plot\_decision\_regions(X\_train, y\_train, clf=knn, legend=2)

plt.title('K-Nearest Neighbors Decision Regions')

plt.xlabel('Feature 1')

plt.ylabel('Feature 2')

plt.show()



# Обчислення точності класифікації:

accuracy = knn.score(X\_test, y\_test)

print('The accuracy is: {0:0.3f}'. format(accuracy))

The accuracy is: 0.815

uploaded = files.upload()

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

# Вибір ознак. Обираємо перші 2 з датасету.

X = df\_healthcare.iloc[:, 1:3] # беремо 2 колонки, пропускаючи першу

y = df\_healthcare["Outcome"]

feature\_names = X.columns

# Вибір ознак. Обираємо кращі ознаки 2 з датасету за допомогою SelectKBest

selector = SelectKBest(score\_func=f\_classif, k=2)

X\_new = selector.fit\_transform(X, y)

selected\_features = [feature\_names[i] for i in

selector.get\_support(indices=True)]

print(f"Selected features: {selected\_features}")

Selected features: ['Pregnancies', 'Glucose']

# Поділ вибірки на навчальну і тестову.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_new, y, test\_size=0.3,

random\_state=42, stratify=y)

# Стандартизація.

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Застосування алгоритму kNN:

knn = KNeighborsClassifier(n\_neighbors=5)

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

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

# Обчислення точності класифікації:

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

print(f"Accuracy: {accuracy:.2f}")

target\_names = ['Non-Diabetic', 'Diabetic']

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

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

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='crest', xticklabels=target\_names, yticklabels=target\_names)

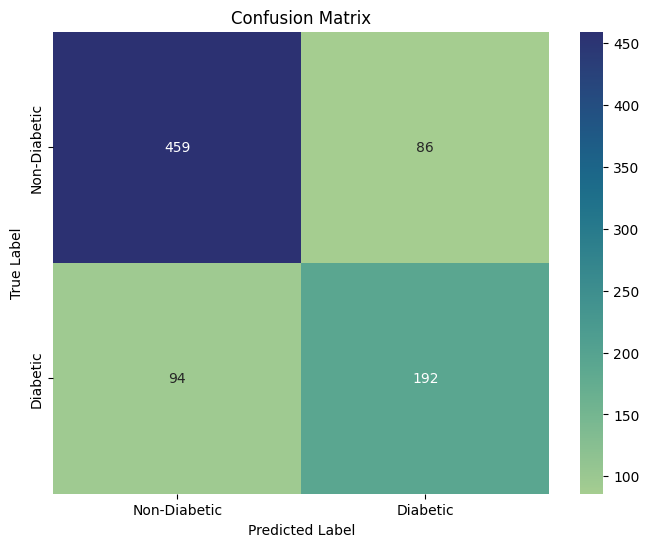
plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix')

plt.show()

Accuracy: 0.78



# Виведення звіту classification report

class\_report = classification\_report(y\_test, y\_pred, target\_names = target\_names)

print("Classification Report:")

print(class\_report)

Classification Report:

precision recall f1-score support

Non-Diabetic 0.83 0.84 0.84 545

Diabetic 0.69 0.67 0.68 286

accuracy 0.78 831

macro avg 0.76 0.76 0.76 831

weighted avg 0.78 0.78 0.78 831

# Візуалізація отриманих результатів

plt.figure(figsize=(10, 6))

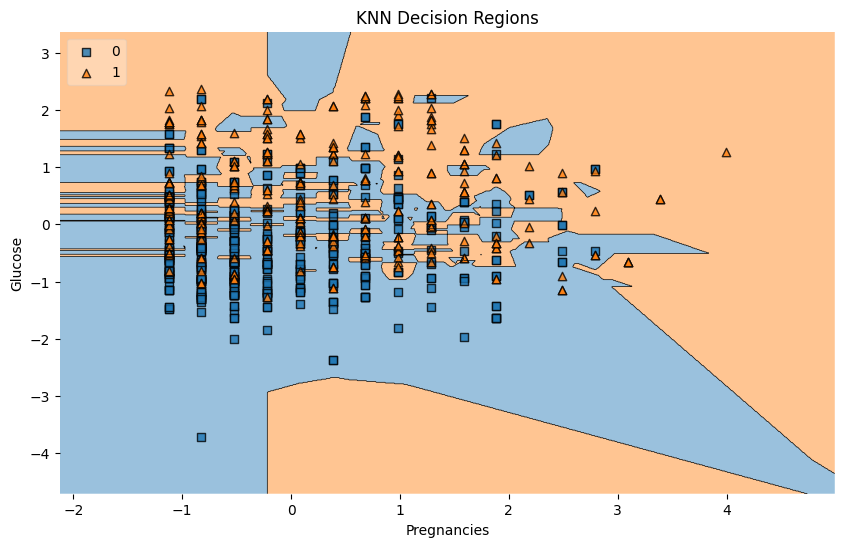
plot\_decision\_regions(X\_test, y\_test.values, clf=knn, legend=2)

plt.title('KNN Decision Regions')

plt.xlabel(selected\_features[0])

plt.ylabel(selected\_features[1])

plt.show()



**Висновок:** лабораторній роботі було вивчено алгоритм методу k-найближчих сусідів, виконано класифікація двух датасетів.