**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

**Лабораторна робота №**7

з дисципліни «Інтелектуальний аналіз та візуалізація даних»

тема «КЛАСИФІКАЦІЯ, МЕТОД ОПОРНИХ ВЕКТОРІВ»

**Варіант №8**

Виконала студентка

групи КН-21

Іванова А.О.

Перевірив(-ла):

МІнаєва Ю.І.

**Київ – 2025**

**Мета роботи:**

Вивчення алгоритму класифікації методом опорних векторів.

import numpy as np

from sklearn import datasets

import matplotlib.pyplot as plt

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

from sklearn.svm import SVC

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

import pandas as pd

from google.colab import files

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

from mlxtend.plotting import plot\_decision\_regions

# Завантаження датасету Wine

wine = datasets.load\_wine()

X = wine.data

y = wine.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

svm\_model = SVC(kernel='linear')

svm\_model.fit(X\_train, y\_train)

y\_pred = svm\_model.predict(X\_test)

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

print("Classification Accuracy:", accuracy)

Classification Accuracy: 0.9814814814814815

print("Classification Report:")

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

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 19

1 1.00 0.95 0.98 21

2 0.93 1.00 0.97 14

accuracy 0.98 54

macro avg 0.98 0.98 0.98 54

weighted avg 0.98 0.98 0.98 54

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

print("Confusion Matrix:")

print(conf\_matrix)

Confusion Matrix:

[[19 0 0]

[ 0 20 1]

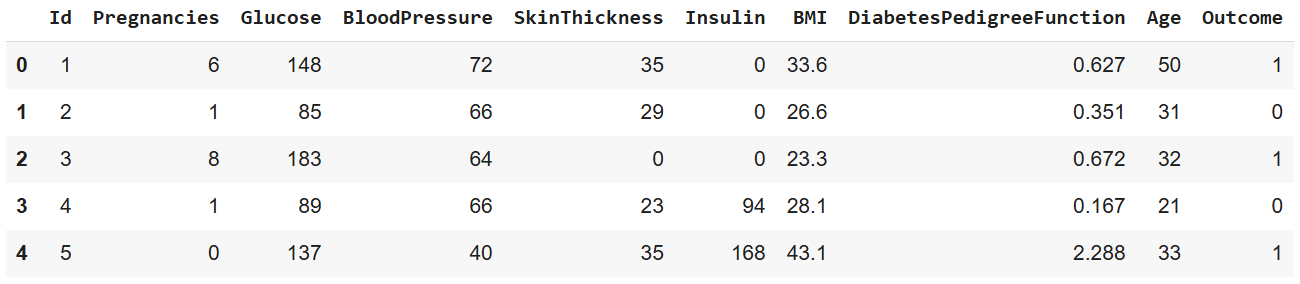
[ 0 0 14]]

# Завантаження датасету Healthcare Diabetse

uploaded = files.upload()

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

df\_healthcare.head()



X = df\_healthcare.drop(columns=["Outcome"])

y = df\_healthcare["Outcome"]

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

svm\_model = SVC(kernel='rbf')

svm\_model.fit(X\_train, y\_train)

y\_pred = svm\_model.predict(X\_test)

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

print("Classification Accuracy:", accuracy)

Classification Accuracy: 0.6762936221419976

print("Classification Report:")

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

Classification Report:

precision recall f1-score support

0 0.68 1.00 0.81 562

1 0.00 0.00 0.00 269

accuracy 0.68 831

macro avg 0.34 0.50 0.40 831

weighted avg 0.46 0.68 0.55 831

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

print("Confusion Matrix:")

print(conf\_matrix)

Confusion Matrix:

[[562 0]

[269 0]]

# 1. Вибір двох конкретних ознак

X\_selected = X\_train[["Glucose", "Age"]]

y\_selected = y\_train

# 2. Навчання SVM-моделі тільки на цих ознаках

from sklearn.svm import SVC

svm\_selected = SVC(kernel="linear") # або 'rbf', 'poly', тощо

svm\_selected.fit(X\_selected, y\_selected)

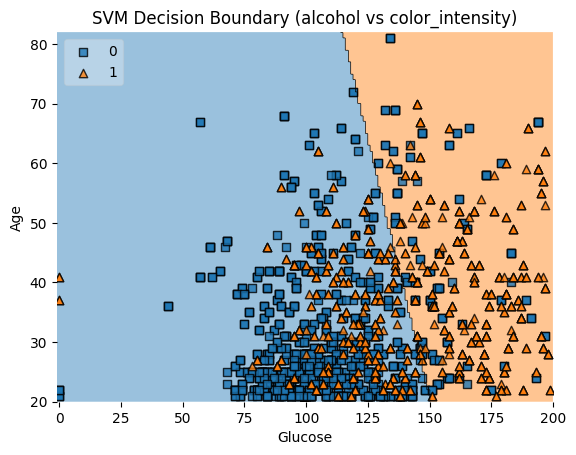
plot\_decision\_regions(X=X\_selected.values, y=y\_selected.values.ravel(), clf=svm\_selected, legend=2)

plt.xlabel("Glucose")

plt.ylabel("Age")

plt.title("SVM Decision Boundary (alcohol vs color\_intensity)")

plt.show()



**Висновок:** лабораторній роботі було вивчено алгоритму класифікації методом опорних векторів.