**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра інтелектуальних технологій**

**Лабораторна робота №3.1**

з дисципліни «Інтелектуальний аналіз даних»

Тема роботи: «Класифікація. Метод к-найближчих сусідів»

**Варіант №5**

Виконала:

студентка групи АнД-31

Радоманова С. П.

Перевірив:

Чолишкіна О.Г.

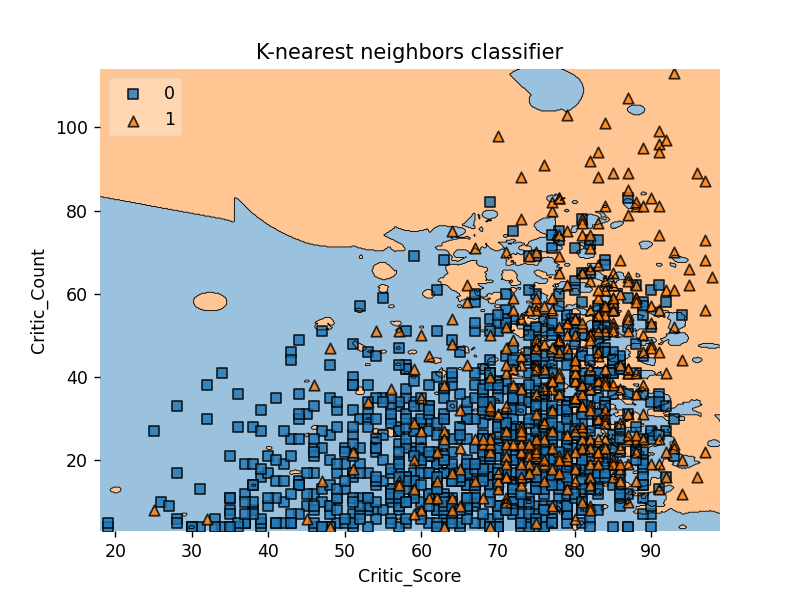
**Київ – 2025**

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

Для лабораторної роботи був обраний датасет Video\_Games\_Sales\_as\_at\_22\_Dec\_2016 з інформацією про продажі випущених в 2016 відеоігор, на основі цих даних будемо класифікувати ігри як успішні та неуспішні базуючись на основі показника Global\_sales та інших.

Для початку очищаємо датасет від порожніх значень та дублювань, а також строкові значення переводимо в числові за допомогою LabelEncoder. Як було сказано раніше, за таргетну змінну обираємо загальну кількість продажів, Х формуємо з інших змінних (з них виключаю інші показники продажів, такі як продажі в Японії, Америці, Європі і тд, оскільки ці показники очевидно найбільше корелюють з загальною кількістю продажів і будуть обиратись алгоритмом SelectKbest). За допомогою селектора обираємо дві ознаки, за якими і буде працювати КНН, в моєму випадку це 

Важливо обрати саме дві ознаки, для подальшої побудови двомірної області прийняття рішеннь. Після цього ділимо вибірку на тренувальні і тестові дані, навчаємо модель і візуалізуємо результати:

****

Можна зробити висновок, що «найхітовішими» стають ігри, які оцінює найбільша кількість критиків, і в яких при цьому найвищі оцінки.

**Висновки:** Оцінюючи складність алгоритму, можна сказати, що KNN є простим у реалізації та не потребує складного навчання, але має відносно високу обчислювальну складність на етапі прогнозування, оскільки кожного разу розраховує відстані до всіх сусідів. Для невеликих датасетів, як у нашому випадку, це не створює значних проблем, однак для великих обсягів даних може бути проблематично.

Точність класифікації виявилась задовільною: модель правильно відокремлює «хітові» ігри від менш успішних, спираючись на ключові ознаки (оцінки критиків та кількість рецензій). Це підтверджує придатність методу k-найближчих сусідів для розв’язання задач класифікації.

Лістинг програми:

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.model\_selection import train\_test\_split  
from sklearn.neighbors import KNeighborsClassifier  
from mlxtend.plotting import plot\_decision\_regions  
from sklearn.feature\_selection import SelectKBest, f\_classif  
from sklearn.preprocessing import LabelEncoder  
  
data = pd.read\_csv("Video\_Games\_Sales\_as\_at\_22\_Dec\_2016.csv/Video\_Games\_Sales\_as\_at\_22\_Dec\_2016.csv")  
  
print(data.head())  
data = data.dropna()  
data = data.drop\_duplicates()  
  
# Перетворюємо текстові колонки у числа  
df = data.copy()  
for col in df.select\_dtypes(include=['object']).columns:  
 df[col] = LabelEncoder().fit\_transform(df[col])  
  
  
X = df.drop(columns=['Global\_Sales', 'NA\_Sales', 'EU\_Sales', 'JP\_Sales', 'Other\_Sales',])  
# Знаходимо поріг для топ-25% ігор  
threshold = df['Global\_Sales'].quantile(0.75)  
y = (df['Global\_Sales'] >= threshold).astype(int)  
  
  
selector = SelectKBest(f\_classif, k=2)  
X\_new = selector.fit\_transform(X, y)  
selected\_features = [X.columns[i] for i in selector.get\_support(indices=True)]  
print(f"Selected features: {selected\_features}")  
  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_new, y, test\_size=0.2, random\_state=42)  
knn = KNeighborsClassifier(n\_neighbors=5, weights='distance', algorithm='auto')  
knn.fit(X\_train, y\_train)  
  
plot\_decision\_regions(X\_test, y\_test.values, knn, legend=2)  
plt.title("K-nearest neighbors classifier")  
plt.xlabel(selected\_features[0])  
plt.ylabel(selected\_features[1])  
  
plt.show()