## **Державний університет інформаційно-комунікаційних технологій**

**Практична робота №5**

**з дисципліни: Технології штучного інтелекту**

**Тема: Машинні моделі в задачах прогнозування за методами лінійної та логістичної регресії**

Виконав:  
студент ДУІКТ  
Тертишний В.Ю.

група: ШІДМ-51

м.Київ

**Мета:** Ознайомитися з основними поняттями лінійної регресії та методами її застосування для прогнозування. Навчитися будувати та аналізувати модель лінійної регресії, визначати залежність між змінними, проводити оцінку якості моделі та робити прогнозування на основі отриманих результатів. Практично закріпити навички використання інструментів програмування Python та бібліотеки scikit-learn для реалізації моделі лінійної регресії на реальних даних.

**Код:**

import pandas as pd

import numpy as np

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler

import os

# Вивести поточну робочу директорію

print("Поточна робоча директорія:", os.getcwd())

# Завантаження даних з CSV

file\_path = r' #шлях на файл ПР5 '

try:

# Завантажуємо дані з CSV

data = pd.read\_csv(file\_path)

print("Дані успішно завантажено!")

except FileNotFoundError:

print(f"Файл за шляхом {file\_path} не знайдено. Перевірте шлях до файлу.")

exit()

# Попередній перегляд даних

print("Перші рядки даних:")

print(data.head())

# Перевірка на пропущені значення

print("\nКількість пропущених значень у кожному стовпці:")

print(data.isnull().sum())

# Заповнення пропущених значень тільки для числових стовпців

numerical\_columns = data.select\_dtypes(include=['float64', 'int64']).columns

data[numerical\_columns] = data[numerical\_columns].fillna(data[numerical\_columns].mean())

# Перетворення стовпця 'month\_year' у формат дати

data['month\_year'] = pd.to\_datetime(data['month\_year'], errors='coerce') # Якщо формат різний, неперетворені значення стануть NaT

data['Month'] = data['month\_year'].dt.month # Виділяємо місяць

data['Year'] = data['month\_year'].dt.year # Виділяємо рік

# Закодування категоріальних змінних (наприклад, місяців як сезонних змінних)

data = pd.get\_dummies(data, columns=['Month'], drop\_first=True)

# Нормалізація числових даних

numerical\_features = data.select\_dtypes(include=['float64', 'int64']).columns

scaler = StandardScaler()

data[numerical\_features] = scaler.fit\_transform(data[numerical\_features])

# Обчислення кореляційної матриці тільки для числових стовпців

numerical\_data = data.select\_dtypes(include=['float64', 'int64']) # Використовуємо тільки числові стовпці

plt.figure(figsize=(12, 10))

sns.heatmap(numerical\_data.corr(), annot=True, cmap='coolwarm') # Тепер застосовуємо corr() тільки до числових стовпців

plt.title('Кореляційна матриця')

plt.show()

# Вибір цільової змінної та ознак

# Замініть 'Sales' на назву вашої цільової змінної

X = data.drop(['Sales', 'month\_year'], axis=1) # Видаляємо цільову змінну та зайві стовпці

y = data['Sales']

# Розподіл на навчальну та тестову вибірки

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Ініціалізація та тренування моделі лінійної регресії

model = LinearRegression()

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

# Прогнозування на тестовій вибірці

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

# Оцінка точності моделі

mse = mean\_squared\_error(y\_test, y\_pred)

mae = mean\_absolute\_error(y\_test, y\_pred)

print(f'\nОцінка моделі:')

print(f'MSE (Середньоквадратична помилка): {mse}')

print(f'MAE (Середня абсолютна помилка): {mae}')

# Візуалізація реальних і передбачених значень

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

plt.scatter(y\_test, y\_pred, alpha=0.7)

plt.xlabel('Реальні значення')

plt.ylabel('Передбачені значення')

plt.title('Реальні проти передбачених значень')

plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--') # Лінія ідеальної відповідності

plt.show()

# Збереження результатів оцінки моделі у файл

results = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

results.to\_csv('sales\_predictions.csv', index=False)

print("\nРезультати збережено у файл 'sales\_predictions.csv'")

**Робота коду:**  
(**>Скріни сюди<**)

**Висновки**

В рамках виконання цієї мети було ознайомлено з основними поняттями лінійної регресії, яка є важливим інструментом для аналізу залежностей між змінними та прогнозування результатів. Було вивчено основні етапи побудови моделі лінійної регресії, починаючи від підготовки даних і закінчуючи оцінкою її ефективності. Зокрема, було засвоєно, як визначати залежність між незалежними та залежними змінними за допомогою моделі, а також як оцінювати її точність і придатність для прогнозування на основі метрик, таких як середньоквадратична та середня абсолютна помилка. Крім того, було набуте практичне вміння працювати з інструментами програмування Python та бібліотекою scikit-learn, що дозволяє ефективно будувати, аналізувати та оцінювати моделі лінійної регресії на реальних даних. Важливою частиною процесу стало розуміння якості моделей, їх здатності робити прогнози на основі реальних наборів даних, а також можливості застосування таких моделей для вирішення практичних задач у різних галузях.