**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №4**

**з навчальної дисципліни «Вступ до технології Data Science»**

**Тема:**

**ДОСЛІДЖЕННЯ БАГАТОКРИТЕРІАЛЬНИХ МЕТОДІВ ПРИЙНЯТТЯ РІШЕНЬ**

**Виконав:**

Студент X курсу кафедри ОТ ФІОТ,

Навчальної групи ІА-73

Петренко П.П.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

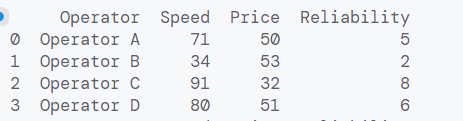
**Київ 2023**

**І. Мета:**

виявити дослідити та узагальнити особливості багатокритеріальних методів прийняття рішень з використанням спеціалізованих пакетів мови програмування Python.

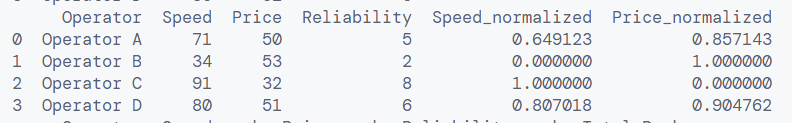
**ІІ. Завдання:**

| **Варіант** (порядковий номер в списку групи) | **Технічні умови завдання** |
| --- | --- |
| 6,21 | Розробити програмний скрипт, що реалізує багатокритеріальне оцінювання ефективності ***телекомунікаційних послуг різних операторів Інтернет послуг***. Формування показників та критеріїв ефективності, синтез багатокритеріальної оптимізаційної моделі здійснити самостійно. |

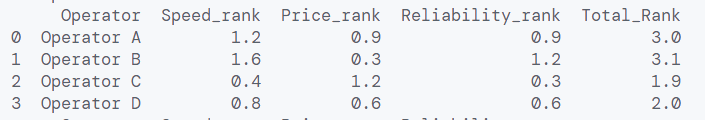
****

**Рисунок 1. Таблиця операторів**

**Метод нормалізації**

****

**Метод ранжування**

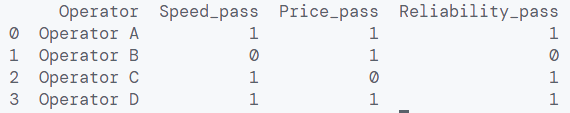
****

Speed\_rank, Price\_rank, Reliability\_rank: Ранжування альтернатив за кожним критерієм (наприклад, чим менше значення, тим краще).

Total\_Rank - Загальний рейтинг кожної альтернативи на основі ваг і рангів критеріїв.

Порогові значення:

**Метод порогових значень**

****

Speed\_pass, Price\_pass, Reliability\_pass: Позначає, чи перевищує кожна альтернатива встановлене порогове значення для швидкості передачі даних, ціни і надійності відповідно. Значення 1 вказує на те, що критерій виконується, а 0 - навпаки.

**ІІІ.** Довести адекватність сформованих моделей та працездатність розробленого скріпта.

Розроблений код повинен бути раціональним та відповідати вимогам до чистого коду.

**import pandas as pd**

**import numpy as np**

**def create\_dataset():**

**np.random.seed(42)**

**data = {**

**'Operator': ['Operator A', 'Operator B', 'Operator C', 'Operator D'],**

**'Speed': np.random.randint(20, 100, size=4),**

**'Price': np.random.randint(30, 100, size=4),**

**'Reliability': np.random.randint(1, 10, size=4)**

**}**

**telecom\_dataset = pd.DataFrame(data)**

**telecom\_dataset.to\_csv('variant\_6.csv', index=False)**

**def normalize(df, columns):**

**result = df.copy()**

**for feature\_name in columns:**

**max\_value = df[feature\_name].max()**

**min\_value = df[feature\_name].min()**

**result[feature\_name + '\_normalized'] = (df[feature\_name] - min\_value) / (max\_value - min\_value)**

**return result**

**def rank(data, weights):**

**result = data.copy()**

**for feature\_name in data.columns[1:]:**

**result[feature\_name + '\_rank'] = data[feature\_name].rank(ascending=False) \* weights[feature\_name]**

**result['Total\_Rank'] = result[[col for col in result.columns if '\_rank' in col]].sum(axis=1)**

**return result**

**def evaluate\_thresholds(data, thresholds):**

**result = data.copy()**

**for feature\_name, threshold\_value in thresholds.items():**

**result[feature\_name + '\_pass'] = (result[feature\_name] >= threshold\_value).astype(int)**

**return result**

**weights = {'Speed': 0.4, 'Price': 0.3, 'Reliability': 0.3}**

**thresholds = {'Speed': 70, 'Price': 40, 'Reliability': 5}**

**create\_dataset()**

**df = pd.read\_csv('variant\_6.csv')**

**print(df)**

**normalized\_data = normalize(df, ['Speed', 'Price'])**

**print(normalized\_data)**

**ranked\_data = rank(df, weights)**

**print(ranked\_data[['Operator', 'Speed\_rank', 'Price\_rank', 'Reliability\_rank', 'Total\_Rank']])**

**threshold\_results = evaluate\_thresholds(df, thresholds)**

**print(threshold\_results[['Operator', 'Speed\_pass', 'Price\_pass', 'Reliability\_pass']])**

**III. Висновок:**

виявив дослідив та узагальнив особливості багатокритеріальних методів прийняття рішень з використанням спеціалізованих пакетів мови програмування Python.

Виконав: студент Петренко П.П.