**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

ЗВІТ

Про виконання лабораторної роботи №1

з дисципліни: «Системний аналіз та теорія прийняття рішень»

Студент групи ПП-31

Вовк Олександр

1. Київ – 2023

**Лабораторна робота №1.**

**Тема:** Прийняття рішення в умовах повної визначеності.

**Мета роботи:** Дослідити методи прийняття рішення в умовах повної визначеності.

**Порядок виконання роботи:**

1. Вирішити задачі засобами MS Excel.
2. Запрограмувати розв’язок задачі.
3. Скласти звіт з лабораторної роботи.
4. Захистити роботу.

Хід роботи:

Скопіював першу таблицю у ексель файл

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерій Альтернатива | **К1** | **К2** | **К3** | **К4** |
| **А1** | 3 | 7 | 2 | 9 |
| **А2** | 8 | 3 | 6 | 7 |
| **А3** | 4 | 8 | 3 | 5 |
| **А4** | 9 | 6 | 5 | 4 |
| **Вес** | 8 | 9 | 6 | 7 |

Далі розрахував оцінки за допомогою формул

|  |  |  |  |
| --- | --- | --- | --- |
| 24 | 63 | 12 | 63 |
| 64 | 27 | 36 | 49 |
| 32 | 72 | 18 | 35 |
| 72 | 54 | 30 | 28 |

Далі підрахував сумарні оцінки та знайшов максимальну

|  |  |  |  |
| --- | --- | --- | --- |
| 162 |  | Макс | 184 |
| 176 |  |  |  |
| 157 |  |  |  |
| 184 |  |  |  |

Скопіював другу таблицю у ексель файл. Також для себе відмітив, які критерії необхідно максимізувати, а які мінімізувати. Мінімізувати необхідно тільки другий критерій.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий Альтернатива | **К1** | **К2** | **К3** | **К4** | **К5** |
| **А1** | 85 | 30 | 22 | 0,65 | 6 |
| **А2** | 60 | 20 | 10 | 0,6 | 7 |
| **А3** | 30 | 12 | 5 | 0,45 | 5 |
| **А4** | 75 | 24 | 13 | 0,7 | 8 |
| **А5** | 40 | 15 | 7 | 0,55 | 7 |
| **Вес** | 7 | 5 | 6 | 8 | 6 |

Нормалізував оцінки за прикладом

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0,8 | 0,333333 |
| 0,545455 | 0,555556 | 0,294118 | 0,6 | 0,666667 |
| 0 | 1 | 0 | 0 | 0 |
| 0,818182 | 0,333333 | 0,470588 | 1 | 1 |
| 0,181818 | 0,833333 | 0,117647 | 0,4 | 0,666667 |

Домножив оцінки на вагу

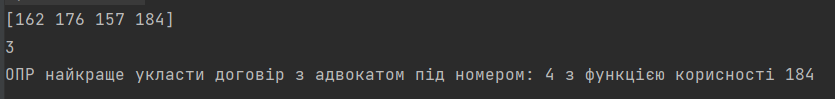
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 0 | 6 | 6,4 | 2 |
| 3,818182 | 2,777778 | 1,764706 | 4,8 | 4 |
| 0 | 5 | 0 | 0 | 0 |
| 5,727273 | 1,666667 | 2,823529 | 8 | 6 |
| 1,272727 | 4,166667 | 0,705882 | 3,2 | 4 |

Далі підрахував сумарні оцінки та знайшов максимальну

|  |  |  |  |
| --- | --- | --- | --- |
| 21,4 |  | макс = 24,21741 | |
| 17,16067 |  |  |  |
| 5 |  |  |  |
| 24,21747 |  |  |  |
| 13,34528 |  |  |  |

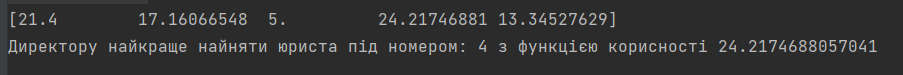
Вирішив перше завдання за допомогою Python. Підраховую сумарні оцінки кожної альтернативи та виводжу найкращу на екран.

import numpy as np  
  
matrix = np.array([  
 [3, 7, 2, 9],  
 [8, 3, 6, 7],  
 [4, 8, 3, 5],  
 [9, 6, 5, 4]  
])  
weights = np.array([8, 9, 6, 7])  
criteria\_values = np.dot(matrix, weights)  
print(criteria\_values)  
position = np.argmax(criteria\_values)  
print(position)  
print(f"ОПР найкраще укласти договір з адвокатом під номером: {position+1} з функцією корисності {criteria\_values[position]}")

****

Реалізував друге завдання. Нормалізую інформацію, перемножую матрицю альтернатив з вектором ваги та виводжу найкраще рішення на екран.

import numpy as np  
  
matrix = np.array([  
 [85, 30, 22, 0.65, 6],  
 [60, 20, 10, 0.6, 7],  
 [30, 12, 5, 0.45, 5],  
 [75, 24, 13, 0.7, 8],  
 [40, 15, 7, 0.55, 7]  
])  
weights = np.array([7, 5, 6, 8, 6])  
def normalize\_matrix(matrix):  
 if not isinstance(matrix, np.ndarray):  
 print("type mismatch")  
 quit()  
  
 if len(matrix) != 5:  
 print("invalid matrix")  
 quit()  
  
 for row in matrix:  
 if not isinstance(row, np.ndarray) or len(row) != 5:  
 print("invalid matrix")  
 quit()  
  
 min\_values = np.min(matrix, axis=0)  
 max\_values = np.max(matrix, axis=0)  
 first\_normalized\_column = (matrix[:, 0] - min\_values[0]) / (max\_values[0] - min\_values[0])  
  
 min\_value = np.min(matrix[:, 1])  
 max\_value = np.max(matrix[:, 1])  
 second\_normalized\_column = (max\_value - matrix[:, 1]) / (max\_value - min\_value)  
  
 min\_value = np.min(matrix[:, 2])  
 max\_value = np.max(matrix[:, 2])  
 third\_normalized\_column = (matrix[:, 2] - min\_value) / (max\_value - min\_value)  
  
 min\_value = np.min(matrix[:, 3])  
 max\_value = np.max(matrix[:, 3])  
 fourth\_normalized\_column= (matrix[:, 3] - min\_value) / (max\_value - min\_value)  
  
 min\_value = np.min(matrix[:, 4])  
 max\_value = np.max(matrix[:, 4])  
 last\_normalized\_column = (matrix[:, 4] - min\_value) / (max\_value - min\_value)  
 return np.column\_stack((first\_normalized\_column,  
 second\_normalized\_column,  
 third\_normalized\_column,  
 fourth\_normalized\_column,  
 last\_normalized\_column))  
result\_matrix = np.dot(normalize\_matrix(matrix), weights)  
print(result\_matrix)  
position = np.argmax(result\_matrix)  
print(f"Директору найкраще найняти юриста під номером: {position+1} з функцією корисності {result\_matrix[position]}")



**Висновки**

В результаті виконання цієї лабораорної роботи, я розглянув прийняття рішення в умовах повної визначеності з використанням Excel та реалізував прийняття рішення програмно.