

华东师范大学软件学院课程作业

课程名称：软件质量分析	年级：2023 级本科	姓名：张梓卫
作业主题：软件可信性的度量值计算	学号：10235101526	作业日期：2024/10/29
指导老师：陈仪香	组号：	

目录

一 第一道题	1
1 题目内容	1
2 题目解析	1
3 输出结果	2
二 第二道题	2
1 题目内容	2
2 题目分析	3
3 题目解答	4
三 参考资料	4

一 第一道题

1 题目内容

设软件 S 有五个属性，譬如功能性 (**Functionality**)、安全性 (**Safety**)、可靠性 (**Reliability**)、生存性 (**Survivability**)、可维护性 (**Maintainability**)，分别使用字母 F, SF, R, SV, M 表示。

α_F	α_{SF}	α_R	α_{SV}	α_M
0.25	0.15	0.20	0.23	0.17

F	SF	R	SV	M	T
6.8	8.2	7.8	6.8	7.9	
8.8	8.9	9.2	9.0	9.3	
5.6	5.9	6.0	9.0	5.3	
4.6	7.8	3.8	5.1	4.5	

2 题目解析

可以使用 Python 中的 Numpy 与 Pandas 库进行矩阵运算，因为 Pandas 中的 DataFrame 可以方便地进行遍历计算。

```
1 import pandas as pd
2 import numpy as np
3
4 weights = {
5     'F': 0.25, 'SF': 0.15, 'R': 0.20, 'SV': 0.23, 'M': 0.17
```

```
6 }
7
8 data = [
9     [6.8, 8.2, 7.8, 6.8, 7.9],
10    [8.8, 8.9, 9.2, 9.0, 9.3],
11    [5.6, 5.9, 6.0, 9.0, 5.3],
12    [4.6, 7.8, 3.8, 5.1, 4.5]
13 ]
14
15 def main():
16     dataframe = pd.DataFrame(data, columns=['F', 'SF', 'R', 'SV', 'M'])
17     dataframe['T'] = dataframe.apply(lambda
18 row: np.prod([row[attr] ** weights[attr] for attr in weights]), axis=1)
19     print(dataframe)
20
21 if __name__ == '__main__':
22     main()
```

计算软件可信性度量值

3 输出结果

输出结果如下：

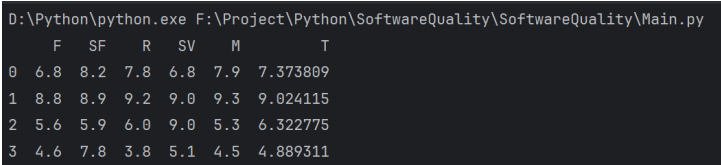


图 1: 软件可信性度量值计算

故最终结果为：

<i>F</i>	<i>SF</i>	<i>R</i>	<i>SV</i>	<i>M</i>	<i>T</i>
6.8	8.2	7.8	6.8	7.9	7.373809
8.8	8.9	9.2	9.0	9.3	9.024115
5.6	5.9	6.0	9.0	5.3	6.322775
4.6	7.8	3.8	5.1	4.5	4.889311

二 第二道题

1 题目内容

有 4 个软件系统，分别编号为 1, 2, 3, 4。它们都有 7 个属性，其中 4 个为关键属性，编号为 y_1, y_2, y_3, y_4 ，其权重分别为 $\alpha_1 = 0.3, \alpha_2 = 0.2, \alpha_3 = 0.35, \alpha_4 = 0.15$ 。3 个为非关键属性，编号为 y_5, y_6, y_7 ，其权重分别为 $\beta_1 = 0.35, \beta_2 = 0.40, \beta_3 = 0.25$ 。关键属性权重 $\alpha = 0.70$ ，非关键属性权重 $\beta = 0.30$ 。根据下表的各属性取值使用模型 T_3 计算各软件的可靠度值。

编号	y_1	y_2	y_3	y_4	y_5	y_6	y_7	ε	ρ
1	8.3	7.1	8.1	6.9	9.1	8.5	8.6	0.1	0.10
	7.8	8.7	8.0	6.2	8.9	8.0	8.8	0.05	0.15
2	5.8	8.1	6.8	5.6	7.9	6.8	8.1	0.1	0.10
	6.8	7.8	8.1	6.9	8.9	7.8	8.8	0.05	0.15
3	3.8	3.7	4.8	4.7	3.9	5.8	6.8	0.1	0.10
	8.4	7.2	8.1	7.7	7.9	8.8	6.8	0.05	0.15
4	9.0	8.7	8.9	8.6	9.1	8.8	8.9	0.1	0.10
	8.0	7.8	6.8	8.7	7.9	8.1	8.3	0.05	0.15

2 题目分析

根据 T_3 的公式，我们可以逐步写出代码：

先定义各种常数和变量：

```

1 import pandas as pd
2 import numpy as np
3
4 alpha_weights = [0.3, 0.2, 0.35, 0.15]
5 beta_weights = [0.35, 0.40, 0.25]
6 alpha = 0.70
7 beta = 0.30
8
9 data = [
10     [8.3, 7.1, 8.1, 6.9, 9.1, 8.5, 8.6, 0.10, 0.10],
11     [7.8, 8.7, 8.0, 6.2, 8.9, 8.0, 8.8, 0.05, 0.15],
12     [5.8, 8.1, 6.8, 5.6, 7.9, 6.8, 8.1, 0.10, 0.10],
13     [6.8, 7.8, 8.1, 6.9, 8.9, 7.8, 8.8, 0.05, 0.15],
14     [3.8, 3.7, 4.8, 4.7, 3.9, 5.8, 6.8, 0.10, 0.10],
15     [8.4, 7.2, 8.1, 7.7, 7.9, 8.8, 6.8, 0.05, 0.15],
16     [9.0, 8.7, 8.9, 8.6, 9.1, 8.8, 8.9, 0.10, 0.10],
17     [8.0, 7.8, 6.8, 8.7, 7.9, 8.1, 8.3, 0.05, 0.15]
18 ]
19
20 df = pd.DataFrame(data, columns=['y1', 'y2', 'y3', 'y4', 'y5', 'y6', 'y7', 'epsilon', 'rho'])

```

定义常数和变量

```

1 def calculate_T3(row):
2     y_crucial = row[['y1', 'y2', 'y3', 'y4']].values
3     y_non_crucial = row[['y5', 'y6', 'y7']].values
4     epsilon = row['epsilon']
5     rho = row['rho']

```

获取数据

在刚刚的代码的基础上，按照运算顺序计算 T_3 的值：

```

1 min = np.min((y_crucial / 10) ** epsilon) # 使用 numpy 自带最小值函数求最小值
2 crucial_prod = np.prod(y_crucial ** alpha_weights) # 使用 prod 计算乘积
3 crucial_prod_rho = (min * crucial_prod) ** (-rho) # 按照步骤，是 -p 次方
4 crucial_value = alpha * crucial_prod_rho # 最后再乘以全局关键属性的权重
5
6 # 有了以上代码，非关键属性的可靠度值也应运而生了。
7 non_crucial_prod = np.prod(y_non_crucial ** beta_weights)
8 non_crucial_prod_rho = (non_crucial_prod) ** (-rho)
9 non_crucial_value = beta * non_crucial_prod_rho
10
11 # 注意，最后的T3值需要用倒数计算

```

```
12 | T_3 = (crucial_value + non_crucial_value) ** (-1/rho)
13 | return T_3
```

计算 T3

```
1 | def main():
2 |     df['T3'] = df.apply(calculate_T3, axis=1)
3 |     print(df)
4 |
5 | if __name__ == '__main__':
6 |     main()
```

Main 函数

3 题目解答

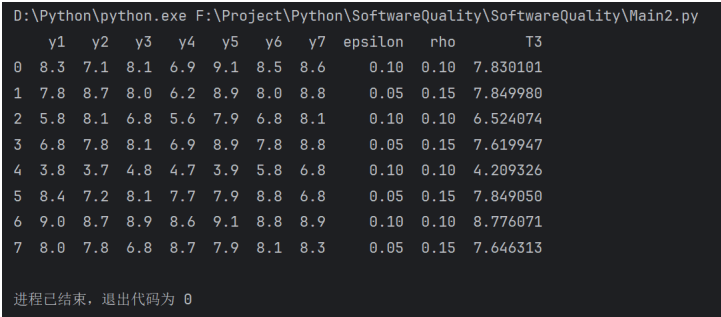


图 2: 软件可靠度值计算

编号	y_1	y_2	y_3	y_4	y_5	y_6	y_7	ε	ρ	T_3
1	8.3	7.1	8.1	6.9	9.1	8.5	8.6	0.1	0.10	7.830101
	7.8	8.7	8.0	6.2	8.9	8.0	8.8	0.05	0.15	7.849980
2	5.8	8.1	6.8	5.6	7.9	6.8	8.1	0.1	0.10	6.524074
	6.8	7.8	8.1	6.9	8.9	7.8	8.8	0.05	0.15	7.619947
3	3.8	3.7	4.8	4.7	3.9	5.8	6.8	0.1	0.10	4.209326
	8.4	7.2	8.1	7.7	7.9	8.8	6.8	0.05	0.15	7.849050
4	9.0	8.7	8.9	8.6	9.1	8.8	8.9	0.1	0.10	8.776071
	8.0	7.8	6.8	8.7	7.9	8.1	8.3	0.05	0.15	7.646313

三 参考资料

- Pandas 官方文档: https://pandas.pydata.org/docs/user_guide/10min.html