Table 1:

	(C1) (%)	(C2)	(C3)	(C4)
0.57381126	0.226415	0.233333	0.337079	0.347826
0.23882897	0.339623	0.300000	0.224719	0.260870
0.13101431	0.283019	0.266667	0.269663	0.217391
0.05634546	0.150943	0.200000	0.168539	0.173913

1.1

- : 0.2494
- : 0.3107
- : 0.2737
- : 0.1663

```
import numpy as np
import pandas as pd

# CSV
file_path = ’D:\\Code\\Python\\mm\\\\judgment_matrix.csv’
data = pd.read_csv(file_path, header=None)
matrix = data.values

#
eigenvalues, eigenvectors = np.linalg.eig(matrix)

#
max_eigenvalue = np.max(eigenvalues)

#  (CI)
n = matrix.shape[0]  #
CI = (max_eigenvalue - n) / (n - 1)

#  (CR) RI  0.9
RI = 0.9
CR = CI / RI

#
if CR < 0.1:
    print("")
else:
    print("CR:", CR)

#
#
max_eigenvalue_index = np.argmax(eigenvalues)
max_eigenvector = eigenvectors[:, max_eigenvalue_index]
```

```
#
weights = np.abs(max_eigenvector) / np.sum(np.abs(max_eigenvector))

#
print(":", max_eigenvalue)
print(" (CI):", CI)
print(" (CR):", CR)
print(":", weights)




import pandas as pd
import numpy as np

#
data = {
    ’’: [’’, ’’, ’’, ’’],
    ’ (C1) (%)’: [12, 18, 15, 8],
    ’ (C2)’: [7, 9, 8, 6],
    ’ (C3)’: [4, 6, 5, 8],
    ’ (C4)’: [8, 6, 5, 4]
}

# DataFrame
df = pd.DataFrame(data)

#
float_columns = df.columns[1:]
df[float_columns] = df[float_columns].apply(pd.to_numeric, errors=’coerce’).astype(float)

#
#
#
for col in df.columns[1:]:
    if col == ’ (C3)’:  #
        df[col] = (1 / df[col]) / (1 / df[col]).sum()  #
    else:  #
        df[col] = df[col] / df[col].sum()  #

#
print("\n")
print(df.to_string(index=False))  # to_string

#
weights = np.array([0.57381126, 0.23882897, 0.13101431, 0.05634546])

#
df[’’] = np.dot(df.iloc[:, 1:], weights)

#
print("\n")
for index, row in df.iterrows():
    print(f"{row[’’]} : {row[’’]:.4f}")

# CSV
output_path = r"D:\Code\Python\mm\\output.csv"
df.to_csv(output_path, index=False)
```