Data Warehousing and Data Mining Lab (CSD-421)
National Institute of Technology, Hamirpur

DATA WAREHOUSING AND DATA MINING LAB (CSD-421) LAB ASSIGNMENT 7

AKSHAT RAJ VANSH (185520)

March 21, 2022

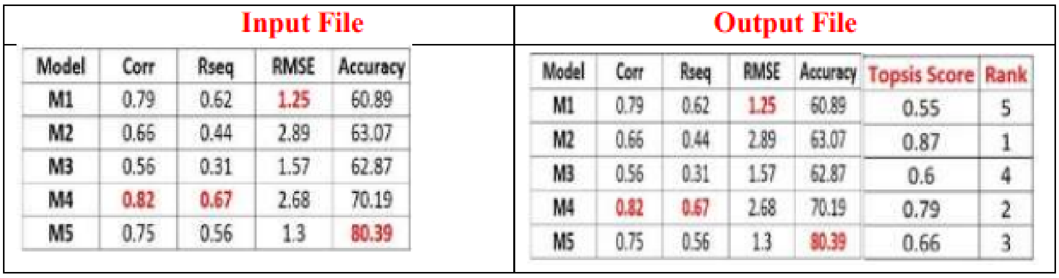
Computer Science Department
National Institute of Technology, Hamirpur

Contents

1 Program 1

1.1 Question

Develop a command line program to implement the TOPSIS.



Input File

Model	Corr	Rseq	RMSE	Accuracy
M1	0.79	0.62	1.25	60.89
M2	0.66	0.44	2.89	63.07
M3	0.56	0.31	1.57	62.87
M4	0.82	0.67	2.68	70.19
M5	0.75	0.56	1.3	80.39

Output File

Model	Corr	Rseq	RMSE	Accuracy	Topsis Score	Rank
M1	0.79	0.62	1.25	60.89	0.55	5
M2	0.66	0.44	2.89	63.07	0.87	1
M3	0.56	0.31	1.57	62.87	0.6	4
M4	0.82	0.67	2.68	70.19	0.79	2
M5	0.75	0.56	1.3	80.39	0.66	3

1.2 Description

Input/Output Files:

- **Input File**
  - Input file contain three or more columns
  - First column is the object/variable name (e.g. M1, M2, M3, M4…..)
  - From 2nd to last columns contain **numeric values only**
- **Output Files**
  - **Result file** contains all the columns of input file and two additional columns having **TOPSIS SCORE and RANK**

Run the program through command line as:

Usages:

```
python topsis.py <InputDataFile> <Weights> <Impacts> <ResultFileName>
```

Example:

```
python topsis.py inputfile.csv  "1,1,1,2"   "+,+,-,+"   result.csv
```

1.3 Check for

- Correct number of parameters (inputFileName, Weights, Impacts, resultFileName).
- Show the appropriate message for wrong inputs.

- Handling of "File not Found" exception
- Input file must contain three or more columns.
- From 2nd to last columns must contain numeric values only (Handling of non-numeric values)
- Number of weights, number of impacts and number of columns (from 2nd to last columns) must be same.
- Impacts must be either +ve or -ve.
- Impacts and weights must be separated by ',' (comma).

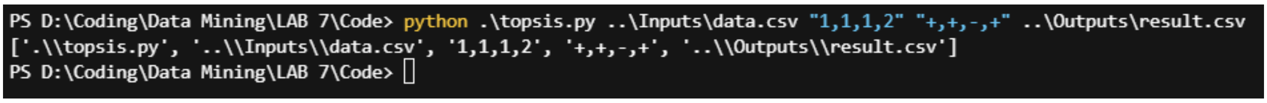
1.4 Code

```
__author__ = 'Akshat Raj Vansh'
__license__ = 'MIT'
__version__ = '0.1.0'

import pandas as pd
import os
import sys
import time

input_string = []

def main():
    if len(input_string) != 5:
        print(input_string)
        print(len(input_string))
        print("ERROR : NUMBER OF PARAMETERS")
        print("USAGE : python topsis.py inputfile.csv '1,1,1,1' '+,+,-,+' result.csv ")
        exit(1)
    elif not os.path.isfile(input_string[1]):
        print(f"ERROR : {input_string[1]} Don't exist!!")
        exit(1)
    elif ".csv" != (os.path.splitext(input_string[1]))[1]:
        print(f"ERROR : {input_string[1]} is not csv!!")
        exit(1)
    else:
        dataset, temp_dataset = pd.read_csv(
            input_string[1]), pd.read_csv(input_string[1])
        noc = len(temp_dataset.columns.values)
        if noc < 3:
            print("ERROR : Input file have less then 3 columns")
            exit(1)
        for i in range(1, noc):
            pd.to_numeric(dataset.iloc[:, i], errors='coerce')
            dataset.iloc[:, i].fillna(
                (dataset.iloc[:, i].mean()), inplace=True)
        try:
            weights = [int(i) for i in input_string[2].split(',')]
        except:
            print("ERROR : In weights array please check again")
            exit(1)
        impact = input_string[3].split(',')
        for i in impact:
            if not (i == '+' or i == '-'):
                print("ERROR : In impact array please check again")
                exit(1)
        if noc != len(weights)+1 or noc != len(impact)+1:
            print(
                "ERROR : Number of weights, number of impacts and number of columns not same")
            exit(1)
        if (".csv" != (os.path.splitext(input_string[4]))[1]):
            print("ERROR : Output file extension is wrong")
            exit(1)
        if os.path.isfile(input_string[4]):
            os.remove(input_string[4])
        topsis_pipy(temp_dataset, dataset, noc, weights, impact)

```

```python

def Normalize(temp_dataset, noc, weights):
    for i in range(1, noc):
        temp = 0
        for j in range(len(temp_dataset)):
            temp = temp + temp_dataset.iloc[j, i]**2
        temp = temp**0.5
        for j in range(len(temp_dataset)):
            temp_dataset.iat[j, i] = (
                temp_dataset.iloc[j, i] / temp)*weights[i-1]
    return temp_dataset


def Calc_Values(temp_dataset, noc, impact):
    p_sln = (temp_dataset.max().values)[1:]
    n_sln = (temp_dataset.min().values)[1:]
    for i in range(1, noc):
        if impact[i-1] == '-':
            p_sln[i-1], n_sln[i-1] = n_sln[i-1], p_sln[i-1]
    return p_sln, n_sln


def topsis_pipy(temp_dataset, dataset, noc, weights, impact):
    temp_dataset = Normalize(temp_dataset, noc, weights)
    p_sln, n_sln = Calc_Values(temp_dataset, noc, impact)
    score = []
    for i in range(len(temp_dataset)):
        temp_p, temp_n = 0, 0
        for j in range(1, noc):
            temp_p = temp_p + (p_sln[j-1] - temp_dataset.iloc[i, j])**2
            temp_n = temp_n + (n_sln[j-1] - temp_dataset.iloc[i, j])**2
        temp_p, temp_n = temp_p**0.5, temp_n**0.5
        score.append(temp_n/(temp_p + temp_n))
    dataset['Topsis Score'] = score
    dataset['Rank'] = (dataset['Topsis Score'].rank(
        method='max', ascending=False))
    dataset = dataset.astype({"Rank": int})
    dataset.to_csv(input_string[4], index=False)


def webService(file, weights,impact):
    global input_string
    string = '.\\topsis.py ..\\Inputs\\'+file+' '+weights+' '+impact+' ..\\Outputs\\result.csv'
    input_string = string.split(' ')
    main()
    output = {'result': str(pd.read_csv(
            input_string[4])),
              'resultname': input_string[4],
              }
    return output

if __name__ == "__main__":
    input_string = sys.argv
    print(input_string)
    main()

```

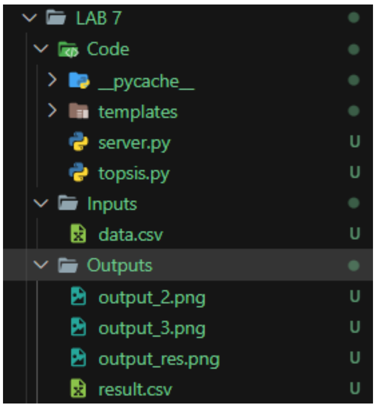
1.5 Output

Running the python script

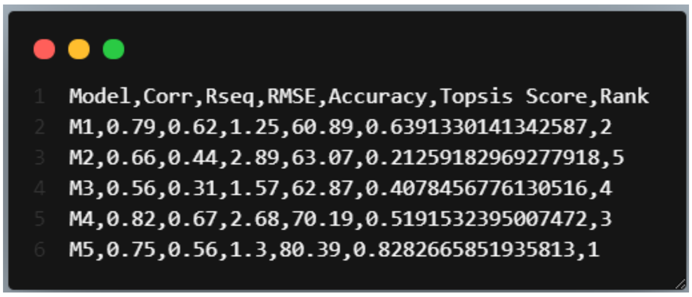


```
PS D:\Coding\Data Mining\LAB 7\Code> python .\topsis.py ..\Inputs\data.csv "1,1,1,2" "+,+,-,+" ..\Outputs\result.csv
['.\\topsis.py', '..\\Inputs\\data.csv', '1,1,1,2', '+,+,-,+', '..\\Outputs\\result.csv']
PS D:\Coding\Data Mining\LAB 7\Code> 
```

Storing Output



Output



```
Model,Corr,Rseq,RMSE,Accuracy,Topsis Score,Rank
M1,0.79,0.62,1.25,60.89,0.6391330141342587,2
M2,0.66,0.44,2.89,63.07,0.21259182969277918,5
M3,0.56,0.31,1.57,62.87,0.4078456776130516,4
M4,0.82,0.67,2.68,70.19,0.5191532395007472,3
M5,0.75,0.56,1.3,80.39,0.8282665851935813,1
```

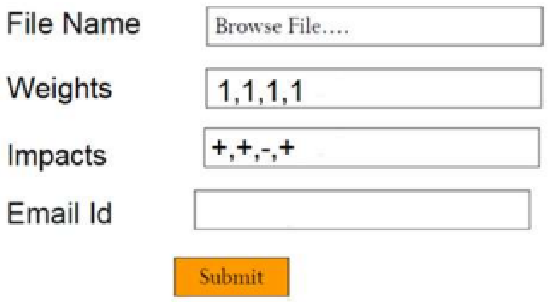
1.6 Result File

```
Model,Corr,Rseq,RMSE,Accuracy,Topsis Score,Rank
M1,0.79,0.62,1.25,60.89,0.6391330141342587,2
M2,0.66,0.44,2.89,63.07,0.21259182969277918,5
M3,0.56,0.31,1.57,62.87,0.4078456776130516,4
M4,0.82,0.67,2.68,70.19,0.5191532395007472,3
M5,0.75,0.56,1.3,80.39,0.8282665851935813,1
```

2 Program 2

2.1 Question

Develop a webservice for program 1



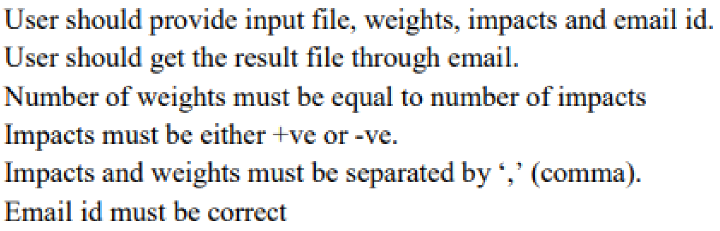
File Name Browse File....

Weights 1,1,1,1

Impacts +,+,-,+

Email Id

2.2 User should get



User should provide input file, weights, impacts and email id.
User should get the result file through email.
Number of weights must be equal to number of impacts
Impacts must be either +ve or -ve.
Impacts and weights must be separated by ',' (comma).
Email id must be correct

2.3 Server Code

```python
import topsis
from bottle import Bottle, template, request

app = Bottle()
output = {}


@app.route('/')
def index():
    """Home Page"""
    return template("./templates/index.tpl", result="", resultname="")


@app.route('/', method="POST")
def formhandler():
    """Handle the form submission"""

    files = request.forms.get('files')
    weights = request.forms.get('weights')
    impact = request.forms.get('impact')

    output = topsis.webService(files,weights,impact)
    return template("./templates/index.tpl",
                    result=output['result'],
                    resultname=output['resultname']
                    )

```

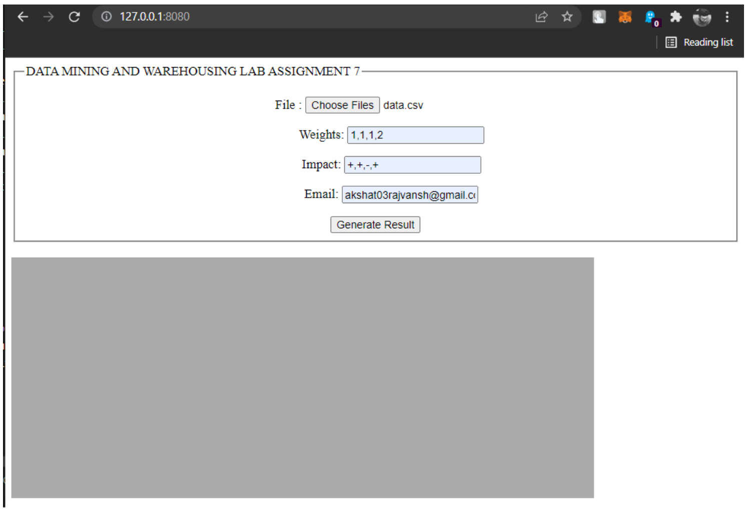
```

if __name__ == '__main__':
    app.run(debug=True)
```

2.4 HTML Page

```
<!DOCTYPE html>
<html>

<head>
    <title>LAB ASSIGNMENT 7</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
        * {
            box-sizing: border-box;
        }

        .column {
            float: none;
            width: 80%;
            padding: 10px;
            height: 300px;
            overflow-x: hidden;
            overflow-y: auto;
            text-align: justify;
        }

        .row:after {
            content: "";
            display: table;
            clear: both;
        }
    </style>
</head>

<body>
    <form method="post" action="/">
        <fieldset>
            <legend>DATA MINING AND WAREHOUSING LAB ASSIGNMENT 7</legend>
            <center>
                <ul>
                    <label for="files">File : </label>
                    <input type="file" id="files" name="files" multiple />
                </ul>
                <ul>Weights:
                    <input name="weights" /></ul>
                <ul>Impact:
                    <input name="impact" /></ul>
                <ul>Email:
                    <input name="email" /></ul>
                <input type="submit" value="Generate Result" />
            </center>
        </fieldset>
    </form>
    <br />

    </div>
    <div class="row">
        <div class="column" style="background-color:#aaa;">
            <h2>{{resultname}}</h2>
            <p>{{result}}</p>
        </div>
    </div>

</body>

</html>
```

2.5 Output

Before Selecting the file and values



DATA MINING AND WAREHOUSING LAB ASSIGNMENT 7

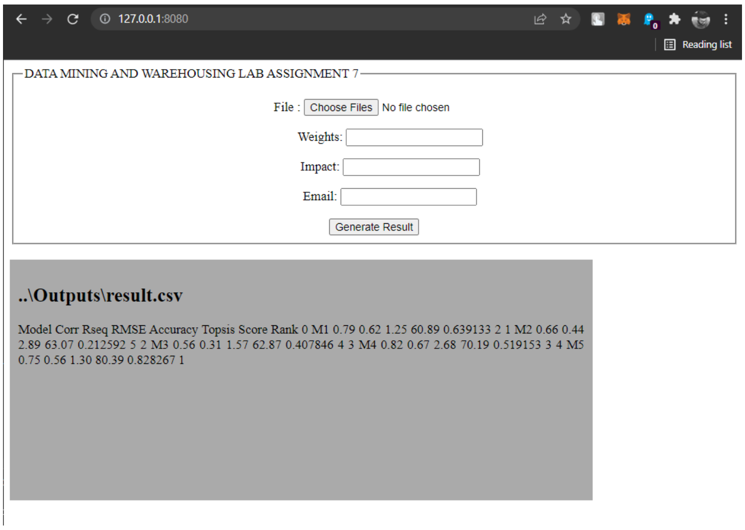
File : data.csv

Weights: 1,1,1,2

Impact: +,+,-,+

Email: akshat03rajvansh@gmail.c

After submitting the selected file



DATA MINING AND WAREHOUSING LAB ASSIGNMENT 7

File : No file chosen

Weights:

Impact:

Email:

..\Outputs\result.csv

Model Corr Rseq RMSE Accuracy Topsis Score Rank 0 M1 0.79 0.62 1.25 60.89 0.639133 2 1 M2 0.66 0.44 2.89 63.07 0.212592 5 2 M3 0.56 0.31 1.57 62.87 0.407846 4 3 M4 0.82 0.67 2.68 70.19 0.519153 3 4 M5 0.75 0.56 1.30 80.39 0.828267 1