In the name of God

استاد : دکتر ابراهیمی

دانشجو : توحید حقیقی سیس

شماره دانشجویی : 830598021

درس : مباحث ویژه

موضوع : تمرین سوم

تمرین اول :

- **با ذکر دلیل بیان کنید که آیا هر مساله را می توان به روش TOPSISرتبه بندی کرد**

روش تاپسیس یکی از روشهای تصمیم گیری چند شاخصه (MADM) است که به رتبه بندی گزینه ها می پردازد. در این روش از دو مفهوم "حل ایده آل" و "شباهت به حل ایده آل" استفاده شده است. حل ایده آل چنان چه از اسم آن پیداست، آن حلی است که از هر جهت بهترین باشد که عموما در عمل وجود نداشته و سعی بر آن است که به آن نزدیک شویم. به منظور اندازه گیری شباهت یک طرح (یا گزینه) به حل ایده آل و ضد ایده آل، فاصله آن طرح (یا گزینه) از حل ایده آل و ضدایده آل اندازه گیری می شود. سپس گزینه ها بر اساس نسبت فاصله از حل ضد ایده آل به مجموع فاصله از حل ایده آل و ضد ایده آل ارزیابی و رتبه بندی می شوند. واژه TOPSIS از حروف اول عبارت Technique for Order of Preference by Similarity to Ideal Solutionگرفته شده است.

بله تمام مسایلی که میخواهیم از بین گزینه ها ان ها را رتبه بندی کنیم میتوانیم از این روش استفاده کنیم .

- اگر در یک سناریو، گزینه ها با روش AHPبه خوبی رتبه بندی شوند آیا روش TOPSISهم می تواند برای همان سناریو گزینه ها را به خوبی روش AHPرتبه بندی کند؟ پاسخ خود را با ذکر دلیل به صورت کامل شرح دهید

نه لزوما ، در روش TOPSISوزن شاخص ها باید بهصورت قطعی مشخص باشدو در مسائل پیچیده به سختی میتوان وزنها را از پیش تعیین کرد . اما در روش AHPوزن شاخص ها بصورت حدودی بر اساس ترجیحات کاربر محاسبه میشود .به همین دلیل نمیتوان انتظار داشت نتایج این دو روش در یک مسئله یکسانباشد. در دو مقاله بررسی شده ، علت این تفاوت جواب را در ناسازگاری بین دیتا ی نمونه اظهار کرده بودند

- مزیت و معایب روش TOPSISنسنبت به روش AHPرا بررسنی نمایید(. این گزارش می بایسنت شامل نقاط قوت و نقاط ضعف هر دو روش باشد

1. تصمیم گیری در صورت وجود معیارهای مثبت و منفی (حتی توام با هم در یک مساله) امکان پذیر است. معیارهای مثبت معیارهایی هستند گه جنبه سود دارند مثل کیفیت کالا و معیارهای منفی معیارهایی هستند که جنبه ضرر دارند مثل سختی کار.
2. برای تعیین بهترین گزینه می توان تعداد قابل توجهی معیار را مورد بررسی قرار داد در حالی که در _AHPیا روش ANP عملا و ذاتا در این زمینه محدودیت هایی وجود دارد.
3. این روش ساده و دارای سرعت مناسب است و برای تعداد زیادی گزینه و معیار به خوبی پاسخگو است.
4. در روش تاپسیس به راحتی می توان معیارهای کیفی را کمی کرد و تصمیم گیری با وجود معیارهای کیفی و کمی میسر است.
5. خروجی سیستم به صورت کمی است و علاوه بر تعیین گزینه برتر، رتبه سایر گزینه ها به صورت عددی بیان می شود. این مقدار عددی همان نزدیکی نسبی است که پایه قوی این روش را بیان می کند.
6. روش تاپسیس، دارای پایه های ریاضی مناسب است. این روش با فاصله ها سروکار دارد. تاپسیس گزینه ای را که بیشترین فاصله از بدترین گزینه و کمترین فاصله از بهترین گزینه دارد، به عنوان گزینۀ بهینه انتخاب می کند و به همین دلیل و پایۀ ریاضی اش، بر سایر روش های MADM برتری دارد.
7. روش تاپسیس برتری دیگری نسبت به بعضی از روشهای MADM دارد که این روش از روش های جبرانی است. یعنی وزن تمامی گزینه ها و معیارها در تصمیم گیری دخالت داده می شود و هیچ وزنی در این روش نادیده گرفته نمی شود.

- به نظر شما در چه مواردی (در چه سناریوهایی) روش TOPSISبهتر از روش AHPعمل می کند؟ و همچنین در چه مواردی روش AHPبهتر از روش TOPSISعمل می کند

در مسائل کوچک که تعداد شاخص ها اندک و تعیین وزن ها آسان است TOPSISبا توجه به سادگی و کم بود محاسبات بسیار کارآمداست. در مسائل پیچیده تر که شاخص ها همبستگی دارند و تعیین وزن ها بدون ایجاد ناسازگاری مشکل است از AHPاستفاده میکنیم

- آیا برای بهبود عملکرد TOPSISایده ای دارید؟

برای این که معیار های مثبت و منفی را با هم اشتباه نگیریم در همان اول معیار های منفی را از 1 کم کنیم تا مثبت شود و با ان محاسبه را انجام دهیم و همچنین می توانیم برای تعیین وزن ها از Ahp استفاده نکنیم چون نظر شخصی در ان اعمال میشود و ممکن است در تصمیم گیری ما اختلال ایجاد کند .

- اگر بخواهیند برای رتبنه بندی بهتر گزیننه هنای ینک سناریو از تل ی روش هنای AHP و TOPSISاست اده کنید چه ایده ای دارید (بصنورت کلی بیان نکنید در صنورتی که ایده ای دارید با یزییات کامل و بصورت الگوریتم وارشرح دهید.

برای بهبودعملکرد TOPSISمی توان از AHP به عنوان روشی برای تعیین وزن ها ی شاخص ها استفاده کرد و وزن های محاسبه شده را برایرتبه بندی آلترناتیوها به TOPSISداد. به این ترتیب جدی ترین مشکل این روش یعنی تعیین وزن ها مرتفع میگردد

تمرین 2 :

عدد کوک : A,B,C,D

گزینه‌ها	صرفه‌جویی در هزینه	هزینه خرید	امنیت	اطلاعات پایه‌ای	نسبت کوک متداول	تکنیک روبنی
A	۱۰	۴	۵	۹	۳	۱
B	۶	۳	۵	۶	۳	۷
C	۲	۶	۷	۵	۴	۱
D	۷	۸	۹	۶	۵	۷

مرحله اول : تبدیل معیارهای کیفی به کمی .

محاسبه با روش AHP :

گزینه‌ها	C_1 (۰/۲)	C_2 (۰/۱)	C_3 (۰/۲)	C_4 (۰/۳)	C_5 (۰/۱)	C_6 (۰/۱)
A	۱۰/۲۵	۴/۲۱	۵/۲۶	۹/۳۲	۳/۱۵	۱/۱۶
B	۶/۲۵	۳/۲۱	۵/۲۶	۹/۳۲	۳/۱۵	۷/۱۶
C	۱/۲۵	۱/۲۱	۷/۲۶	۵/۳۲	۴/۱۵	۱/۱۶
D	۷/۲۵	۸/۲۱	۹/۲۶	۹/۳۲	۵/۱۵	۷/۱۶
	۲۵	۲۱	۲۶	۳۲	۱۵	۱۶

نرمال سازی ستون‌ها : جمع ستون‌ها و تقسیم هر کدام بر ستون‌ها

گزینه‌ها	C_1 منفی	C_2 منفی	C_3 مثبت	C_5 منفی	C_6 مثبت	C_4 مثبت
A	۰/۰۸	۰/۰۱۹	۰/۰۳۸	۰/۰۲	۰/۰۰۶	۰/۰۸۴
B	۰/۰۴۸	۰/۰۱۴	۰/۰۳۸	۰/۰۲	۰/۰۴۳	۰/۰۸۴
C	۰/۰۱۶	۰/۰۲۸	۰/۰۵۳	۰/۰۱۵	۰/۰۰۶	۰/۰۴۶
D	۰/۰۵۶	۰/۰۳۸	۰/۰۵۶	۰/۰۲	۰/۰۴۳	۰/۰۸۴

مرحله وزن معیارها را در ماتریس نرمال ضرب می‌کنیم

S_- S_+

- یافتن حل ایده آل و ضد ایده آل:

• برای معیارهایی که بار مثبت دارند ایده آل مثبت بزرگترین مقدار آن معیار است.

• برای معیارهایی که بار مثبت دارند ایده آل منفی کوچکترین مقدار آن معیار است.

• برای " با منفی دارند ایده آل مثبت کوچکترین " "

	C_1	C_2	C_3	C_4	C_5	C_6
حل ایده آل (S+)	۰/۰۱۶	۰/۰۱۴	۰/۰۵۶	۰/۰۸۴	۰/۰۱۵	۰/۰۴۳
حل ضد ایده آل (S-)	۰/۰۰۸	۰/۰۳۸	۰/۰۳۸	۰/۰۴۶	۰/۰۲	۰/۰۰۶

- محاسبه فاصله از حل ایده آل و ضد ایده آل

در این گام میزان نزدیکی نسبی هر گزینه به راه حل ایده آل محاسبه می شود

فاصله اقلیدسی هر گزینه از ایده آل مثبت و منفی با فرمول زیر محاسبه می شود

فاصله گزینه ها از حل ایده آل	$+d_1$	۰/۲۵۳۶	فاصله گزینه ها از حل ضد ایده آل	$-d_1$	۰/۳۸۴۴
	$+d_2$	۰/۵۵۳۲		$-d_2$	۰/۲۷۱۷
	$+d_3$	۰/۳۰۰۴		$-d_3$	۰/۲۰۳۹
	$+d_4$	۰/۴۰۳۱		$-d_4$	۰/۵۳۳۸
	$\cancel{+d_5}$			$\cancel{-d_5}$	
	$\cancel{+d_6}$			$\cancel{-d_6}$	

$$S_i^* = \sqrt{\sum (v_{ij} - v_j^*)^2}$$

$$S_i^- = \sqrt{\sum (v_{ij} - v_j^-)^2}$$

محاسبه شاخص شباهت:

شاخص شباهت (CL) از فرمول زیر بدست می آید و بین صفر و یک است هرچه به ۱ نزدیک تر باشد به جواب ایده آل نزدیک تر است.

$$C_i^* = S_i^- / S_i^* + S_i^-$$

با محاسبه فرمول بالا به اعداد زیر می رسیم

A	B	C	D
۰,۳۹۷۴	۰,۷۷۰۵	۰,۵۹۵۶	۰,۴۲۰۲

پس اعداد هرچه به ۱ نزدیک تر شود جواب مهمتر خواهد بود.

$$\boxed{B > C > D > A}$$

تمرین سوم :

پیاده سازی روش topsis با پایتون

مراحل این روش را با کد ان به ترتیب در زیر توضیح میدهم :

روش تاپسیس از مراحل زیر تشکیل شده است :

- **گرفتن ماتریس شاخص – گزینه ها از کاربر**

```
def make_matrix(self):
    topsis_table=[]
    for i in range(1,self.alternative_count+1):
        alternative_table=[]
        for j in range(1,self.criteria_count+1):
            alternative_table.append(int(input("Alternative {} Criteria {} :".format(i,j))))
        topsis_table.append(alternative_table)

    return topsis_table
```

در کد بالا ابتدا از کاربر تعداد گزینه ها و شاخص ها را میگیریم و بعد تعداد ان را به این تابع ارسال میکنیم تا این تابع در 2 فور تو در تو اطلاعات را از کاربر بگیرد .

- **گرفتن وزن هایی که روش های دیگر به دست می آید از کاربر**

```
def get_ahp_vector(self):
    ahp_list=[]
    for i in range(self.criteria_count):
        ahp_list.append(float(input("Enter {} Criteria Point :".format(i+1))))
    return ahp_list
```

- در مرحله اول ان را نرمال میکنیم و بعد ماتریس اول را در وکتور وزن ها ضرب میکنیم تا در مراحل بعدی از ان ماتریس به دست امده استفاده کنیم .

```python
def make_normalize_matrix(self,topsis_matrix):
    numpy_topsis_matrix=np.array(topsis_matrix)
    x_normed = numpy_topsis_matrix / numpy_topsis_matrix.sum(axis=0)
    return x_normed
```

نرمالایز کردن به روش های مختلفی میتوان انجام داد روش اول نورم 1 ویا نورم 2 است من در این تابع از نرم 1 برای نرمال کردن استفاده کرده ام .

- گرفتن شاخص های مثبت و منفی از کاربر :

```python
if __name__ == "__main__":
    alternative_count=int(input("Count of Alternative : "))
    Criteria_count=int(input("Count of Criteria : "))


    # get negative criteria column
    negative_column=[]
    for negative in range(0,Criteria_count):
        ans=input("is {} Criteria column negative :".format(negative+1))
        if(ans=="y"):
            negative_column.append(-1)
        else:
            negative_column.append(1)

    print(negative_column)

    #end negative column

    #get matrix criteria and alternative
    topsis = Topsis(alternative_count,Criteria_count)
    topsis_tab=topsis.make_matrix()

    #end get alternative
```

در این قسمت شاخص هایی که تاثیر منفی دارد را از کاربر گرفته و با یک روش ابتکاری ان را در یک وکتور با اعداد 1 و -1 قرار میدهیم .

- یافتن حل ایده ال مثبت و منفی :

برای این روش یک حرکت ابتکاری درست زدم که جواب یکسان میدهد ماتریس مثبت و منفی های مرحله قبل را در ماتریس اصلی ضرب نقطه ای کردم تا ستون هایی که منفی هستند مقادیر منفی بگیرند تا در این مرحله max و min گرفتن از ان راحت تر شود

```
def Get_Ideal_Postive(self,topsis_matrix):
    return topsis_matrix.max(axis=0)

def get_Ideal_negative(self,topsis_matrix):
    return topsis_matrix.min(axis=0)
```

- یافتن فاصله از حالت های مثبت و منفی با ماتریس اقلیدسی اصلی :

```
def Find_Distance_Postive_Matrix(self,postive_ideal,matrix):
    distance_postive=[]
    for i,criteria in enumerate(matrix):
        sum_negative=0
        for j in criteria:
            a=np.sqrt(j**2+postive_ideal[i-1]**2)
            sum_negative+=a
        distance_postive.append(sum_negative)
    return distance_postive

def Find_Distance_negative_Matrix(self,negative_ideal,matrix):
    distance_negative=[]
    for i,criteria in enumerate(matrix):
        sum_negative=0
        for j in criteria:
            a=np.sqrt(j**2+negative_ideal[i-1]**2)
            sum_negative+=a
        distance_negative.append(sum_negative)
    return distance_negative
```

$$d_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^+\right)^2}$$

Sanaye20.ir

$$d_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^-\right)^2}$$

برای هر وکتور مثبت و منفی از فرمول بالا استفاده میکند .

- در مرحله اخر شاخص شباهت را بدست می آوریم

$$cl_i^* = \frac{d_i^-}{d_i^- + d_i^+}$$

Sanaye20.ir

```
def Closness_Coeficent(self,negative_distance,postive_distance):
    array_negative=np.array(negative_distance)
    array_postive=np.array(postive_distance)
    cc=array_negative/(array_postive+array_negative)
    return cc
```

Closeness coefisent برای رتبه بندی گزینه ها استفاده میشود و خروجی ان همیشه اعداد بین صفر ویک است .

فراخوانی توابع در اخر کد ها در main قرار دارد .

```
#normalize matrix for sum of column and devide to sum
normalize_matrix=topsis.make_normalize_matrix(topsis_tab)
# get weight vector
ahp_vector=topsis.get_ahp_vector()
#end

multi_ahp_topsis=topsis.multiply_ahp_into_topsis(ahp_vector,normalize_matrix)

print(multi_ahp_topsis)
add_negative_totopsis = multi_ahp_topsis*negative_column

print(add_negative_totopsis)
#get postive ideal answer
postive_ideal=topsis.Get_Ideal_Postive(add_negative_totopsis)
negative_ideal=topsis.get_Ideal_negative(add_negative_totopsis)



print("----------------------------------------------")

print(postive_ideal)

print("----------------------------------------------")

print(negative_ideal)

# find distance
```