

RUTBALASH ALGORITMLARI - KO'P MEZONLI QARORLARNI HAL QILISH USULLARINI O'RGANISH

Narziyev Nosir Baxshilloevich

TATU ATDT kafedrası katta o'qituvchisi, narziyev@tuit.uz ^[1]

Shonazarov Sarvarbek Maqsud o'g'li

TATU DIF talabasi, sshonazarov938@gmail.com ^[2]

Annotatsiya: *Bu maqola bizda mantiq va parametrlarni almashtirish orqali yaratish mumkin bo'lgan turli xil hal qiluvchi turlari mavjud va ularning deyarli barchasi bizning ehtiyojlarimizdan kelib chiqqan holda turli va qiziqarli natijalar beradi.*

KIRISH

Faraz qilaylik, sizda uy, mashina yoki hatto gitara sotib olish kabi qaror bor. Siz tasodifiy tanlashni yoki kimningdir taklifiga tayanishni xohlamaysiz, balki bilimli qaror qabul qilishni xohlaysiz. Buning uchun siz sotib olmoqchi bo'lgan ob'ekt haqida ma'lumot to'pladingiz (aytaylik, bu mashina). Shunday qilib, sizda narxlari haqidagi ma'lumotlarga ega N avtomobillar ro'yxati mavjud. Odatdagidek, biz ko'proq pul sarflashni xohlamaymiz, shunchaki mashinalarni narxlari bo'yicha (o'sish tartibida) saralashimiz va eng yaxshisini tanlashimiz mumkin (eng kichik narx bilan) va biz tugatdik! Bu yagona mezon asosida qaror qabul qilish edi. Ammo, afsuski, hayot juda oson bo'lsa. Biz ham mashinaning yaxshi yurishini, yaxshi dvigatelini, tezroq tezlashishini (agar siz poyga qilishni istasangiz) va yana bir qancha narsalarni xohlaymiz. Bu erda siz eng kichik narxga ega bo'lgan, lekin eng yuqori kilometr va tezlashuvga ega bo'lgan mashinani tanlamoqchisiz va hokazo. Bu muammoni oddiy tartiblash bilan osonlikcha hal qilib bo'lmaydi. Ko'p mezonli qaror qabul qilish algoritmlarini kiriting!

Ma'lumotlar to'plami

Keling, bitta ma'lumotlar to'plamini tanlaylik, shunda natijani tasavvur qilish, sahna ortida nima sodir bo'layotganini tushunish va nihoyat sezgi yaratish osonroq bo'ladi. Buning uchun men avtomobillar ma'lumotlar to'plamini tanlayapman. Har bir mashina uchun biz atributlar to'plamiga e'tibor qaratamiz va hayotimizni osonlashtirish uchun faqat 10 qatorni (noyob avtomobillar) tanlaymiz. Tanlangan ma'lumotlarga qarang,

	car_name	mpg	displacement	weight	acceleration	price
0	chevrolet chevelle malibu	18.0	307.0	3504	12.0	25561.59078
1	buick skylark 320	15.0	350.0	3693	11.5	24221.42273
2	plymouth satellite	18.0	318.0	3436	11.0	27240.84373
3	amc rebel sst	16.0	304.0	3433	12.0	33684.96888
4	ford torino	17.0	302.0	3449	10.5	20000.00000
5	ford galaxie 500	15.0	429.0	4341	10.0	30000.00000
6	chevrolet impala	14.0	454.0	4354	9.0	35764.33490
7	plymouth fury iii	14.0	440.0	4312	8.5	25899.46557
8	pontiac catalina	14.0	455.0	4425	10.0	32882.53714
9	amc ambassador dpl	15.0	390.0	3850	8.5	32617.05928

Ba'zi atributlarni tushuntirish, mpg- agar siz uning bakiga atigi bir gallon benzin yoki dizel qo'ysangiz, avtomobil qancha masofani bosib o'tishi mumkinligini ko'rsatadigan o'lchov (milli).

displacement-dvigatelning siljishi - bu pistonli dvigatelning barcha pistonlari tomonidan supurilgan silindr hajmining o'lchovidir. Ko'proq joy almashtirish ko'proq quvvatni anglatadi.

acceleration- mashina 0 dan tezlikka yetishi uchun qancha vaqt ketishini ko'rsatadigan o'lchov. Tezlashtirish qanchalik baland bo'lsa, drag poygasi uchun mashina yaxshi bo'ladi.

Bu erda ba'zi fikrlarga e'tibor bering:

1. Atributlarning birligi va taqsimoti bir xil emas. Narx minglab dollarlarda o'ynaydi, tezlashuv o'nlab soniyalarda va hokazo.

	mpg	displacement	weight	acceleration	price
count	10.000000	10.000000	10.000000	10.000000	10.000000
mean	15.600000	374.900000	3879.700000	10.300000	28787.222301
std	1.577621	65.722396	432.151735	1.337494	4997.499733
min	14.000000	302.000000	3433.000000	8.500000	20000.000000
25%	14.250000	309.750000	3462.750000	9.250000	25646.059478
50%	15.000000	370.000000	3771.500000	10.250000	28620.421865
75%	16.750000	437.250000	4333.750000	11.375000	32816.167675
max	18.000000	455.000000	4425.000000	12.000000	35764.334900

2. Har bir atribut uchun eng yaxshi mantig'i ham farq qiladi. Bu erda biz mpg, joy o'zgartirish va tezlashtirishda yuqori qiymatlarga ega bo'lgan avtomobilni topmoqchimiz. Shu bilan birga, og'irlik va narxning past qiymatlari. Ushbu yuqori va past tushunchani mos ravishda atributlarni maksimallashtirish va minimallashtirish deb hisoblash mumkin.

3. Har bir atributni teng deb hisoblamaydigan qo'shimcha talab bo'lishi mumkin. Misol uchun, agar men poyga uchun mashina olishni xohlasam va menga milliarder homiylik qilmoqda desam, mpg va narxga unchalik ahamiyat bermayman. Men tezroq

va eng yengil mashinani xohlayman. Ammo agar men talaba bo'lsam-chi (shuning uchun, ehtimol, qat'iy byudjetda) va ko'p sayohat qilsam, to'satdan mpg va narx eng muhim atributga aylanib qoladi va men joy almashishga ahamiyat bermayman. Atributlarning muhimligi haqidagi bu tushunchalar har bir atributga tayinlangan og'irliklar sifatida qabul qilinishi mumkin. Aytaylik, narx 30% muhim, siljish esa atigi 10% va hokazo.

Talablar aniq bo'lsa, keling, bunday muammolarni qanday hal qilish mumkinligini ko'rib chiqaylik.

Umumiy metodologiya

Ko'p mezonli qarorlarni hal qiluvchi asosiy vositalarning aksariyati umumiy metodologiyaga ega bo'lib, ular quyidagilarga harakat qilishadi:

1. Bir vaqtning o'zida bitta atributni ko'rib chiqing va optimallashtirilgan ball yaratish uchun uni (talab bo'yicha) maksimallashtirish yoki kamaytirishga harakat qiling.

2. Optimallashtirilgan vaznli ballarni olish uchun har bir atributga og'irliklarni kiriting.

3. Tashkilot (bu yerda avtomobil) uchun yakuniy ball yaratish uchun vaznli ballarni (har bir xususiyat uchun) birlashtiring. Shundan so'ng, biz talablarni bitta raqamli atributga (yakuniy ball) aylantirdik va avval qilinganidek, eng yaxshi mashinani olish uchun buni saralashimiz mumkin (bu safar biz eng yuqori ballga ega bo'lgan birini tanlamoqchi bo'lganimiz uchun pasayish bo'yicha saralaymiz). Keling, har bir bosqichni misollar bilan ko'rib chiqaylik.

Maksimallashtirish va kamaytirish

Ma'lumotlar to'plami bo'limining birinchi nuqtasini eslang, atributlar biz hal qilishimiz kerak bo'lgan juda xilma-xil birliklar va taqsimotlarga ega. Mumkin echimlardan biri har bir atributni bir xil diapazonda normallashtirishdir. Va biz ham yaxshilik yo'nalishi o'xshash bo'lishini xohlaymiz (mantiqdan qat'iy nazar). Shunday qilib, normalizatsiya qilingandan so'ng, maksimal diapazonga yaqin qiymatlar (aytaylik, 1) avtomobil ushbu atributda yaxshi ekanligini va pastroq qiymatlar (masalan, 0 ga yaqin) ularning yomonligini bildirishi kerak. Buni quyidagi formula bilan qilamiz, maksimallashtirish uchun birinchi tenglamaga qarang, bitta misol, har bir avtomobilning mpg ni barcha avtomobillarning mpg yig'indisiga bo'lish orqali yangilash (normalizatsiya yig'indisi). Biz mantiqni faqat mpg yoki boshqa formulalarni hisobga olgan holda o'zgartirishimiz mumkin. Maqsad shundan iboratki, buni har bir atributga qo'llaganingizdan so'ng, har bir atributning diapazoni bir xil bo'ladi va biz 1 ga yaqin qiymat yaxshi degan xulosaga kelishimiz mumkin.

Maximization

Let A be an attribute which we want to maximize,
and its elements are: $[a_1 \ a_2 \ \dots \ a_n]$; $1 \leq i \leq n$

Then, $maximize(a_i) = maximizeFunc(a_i, A)$

$$where, maximizeFunc(a_i, A) = \begin{cases} \frac{a_i}{sum(A)}, & \text{sum normalization} \\ \frac{a_i}{max(A)}, & \text{max normalization} \\ \frac{a_i - min(A)}{max(A) - min(A)}, & \text{max-min scaling} \\ \dots & \dots \end{cases}$$

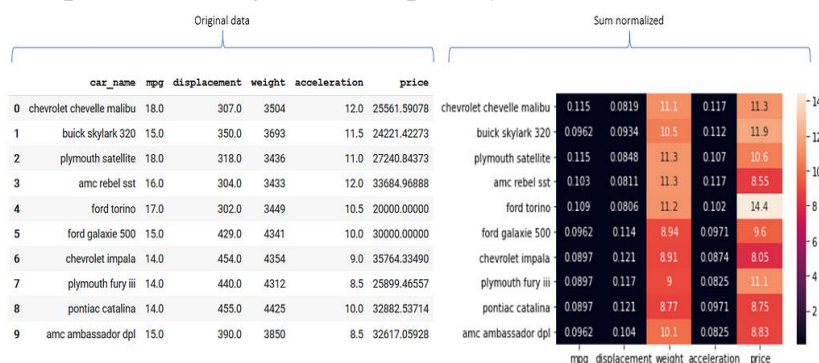
Minimization

Let A be an attribute which we want to minimize,
and its elements are: $[a_1 \ a_2 \ \dots \ a_n]$; $1 \leq i \leq n$

Then, $minimize(a_i) = minimizeFunc(a_i, A)$

$$where, minimizeFunc(a_i, A) = \begin{cases} \frac{1}{maximizeFunc(a_i, A)}, & \text{inverse} \\ 1 - maximizeFunc(a_i, A), & \text{subtract} \\ \dots & \dots \end{cases}$$

Minimallashtirish formulalari maksimallashtirish formulasi bilan deyarli bir xil, biz faqat yaxshilik yo'nalishini teskari qilish uchun (aks holda 1 yomonni, 0 esa yaxshi degan ma'noni anglatadi) faqat teskarisini (1 ga bo'lingan) yoki aks ettiramiz (1 dan ayirish orqali).). Keling, amalda qanday ko'rinishini ko'rib chiqaylik,



Asl ma'lumotlarning yig'indisini normallashtirish issiqlik xaritasiga misol. "Ford torino" ning "mpg" qiymatini tekshiring. Dastlab 17, lekin yig'indisi normallashtirildan so'ng, u $17/156 = 0,109$ bo'lishi kerak. Xuddi shunday, "narx" 20k, teskari bo'lgandan keyin u $1/(20k/287872) = 14,4$ bo'ladi.

Biz vaznni optimallashtirilgan ballar ustiga qo'yishimiz kerak, buni og'irliklarni optimallashtirilgan ballga ko'paytirish orqali osongina bajarish mumkin. Bu erda biz turli xil normalizatsiya turlarini kiritishimiz mumkin,

xuddi shunday: optimallashtirilgan ballga to'g'ridan-to'g'ri og'irliklarni ko'paytirish yig'indi: og'irliklarni yig'indi mantig'i bo'yicha normallashtiring (yuqorida muhokama qilingan), keyin ko'paytiring. maks: maksimal mantiq bilan normallashtiring, keyin ko'paytiring.

Weight modification

Let W denote a vector with weights for each attribute, and its elements are: $[w_1 \ w_2 \ \dots \ w_m]$; $1 \leq i \leq m$

$$modifyWeight(w_i) = \begin{cases} w_i, & \text{as it is} \\ \frac{w_i}{sum(W)}, & \text{sum normalization} \\ \frac{w_i}{max(W)}, & \text{max normalization} \\ \dots & \end{cases}$$

vaznni o'zgartirish mantig'i

Ballarni birlashtiring

Nihoyat, biz hisobni bitta qilish uchun birlashtiramiz. Bu ikki xil mantiq bilan amalga oshirilishi mumkin,

summa: barcha individual ballarni birga qo'shing

mahsulot: barcha individual ballarni birgalikda ko'paytiring. Darhaqiqat, ko'plab ilovalar mahsulotlarni olish o'rniga qiymatning logarifmini qo'shadi, bu kichik qiymatlarni ko'paytirishda juda kichikroq natijani qayta ishlash uchun amalga oshiriladi. Ko'p mezonli qaror qabul qilish muammosi uchun ko'plab algoritmlarni taqdim etadigan skriteria deb nomlangan juda yaxshi python paketi mavjud. Aslida skriteria.madm.simple modulidagi ikkita algoritm:

WeightedSum -individual ball birlashtiruvchi mantiq yig'indisidir

WeightedProduct - individual ball mantiqni birlashtirgan mahsulot (loglarning yig'indisi)

Va bu usullarning ikkalasi ham kirish sifatida ikkita parametrni oladi,

mnorm - qiymatni maksimallashtirish normalizatsiya mantiqini aniqlang (minimallashtirish har doim bir xil maksimallashtirish mantiqiga teskari).

wnorm - vaznni normallashtirish mantiqini aniqlang

Ma'lumotlarimiz bo'yicha reytingni amalga oshirish uchun, avvalo, biz ularni skriteria.Data ob'ekti sifatida yuklashimiz kerak,

```

criteria_data = Data(
    cars_data, # the pandas dataframe
    [MAX, MAX, MIN, MAX, MIN], # direction of goodness for each column
    anames = cars_name, # each entity's name, here car name
    cnames = cars_data.columns, # attribute/column name
    # weights=[1,1,1,1,1] # weights for each attribute (optional)
)
criteria_data

```

	ALT./CRIT.	mpg (max)	displacement (max)	weight (min)	acceleration (max)	price (min)
chevrolet chevelle malibu	18	307	3504	12	25561.6	
buick skylark 320	15	350	3693	11.5	24221.4	
plymouth satellite	18	318	3436	11	27240.8	
amc rebel sst	16	304	3433	12	33685	
ford torino	17	302	3449	10.5	20000	
ford galaxie 500	15	429	4341	10	30000	
chevrolet impala	14	454	4354	9	35764.3	
plymouth fury iii	14	440	4312	8.5	25899.5	
pontiac catalina	14	455	4425	10	32882.5	
amc ambassador dpl	15	390	3850	8.5	32617.1	

ma'lumotlarni Data ob'ektiga yuklash

Ma'lumotlar yuklangan bo'lsa, biz qilishimiz kerak bo'lgan narsa ma'lumotlar ob'ekti va parametr sozlamalari bilan tegishli qaror qabul qiluvchi funksiyasini chaqirishdir. Chiqarishda ko'rsatilgan barcha mezonlarni hisobga olgan holda yakuniy reytingni ko'rsatish uchun bitta qo'shimcha daraja ustuni mavjud.

```
from skcriteria.madm import simple
# weighted sum
dm = simple.WeightedSum(mnorm="sum")
dec = dm.decide(criteria_data)
dec
```

WeightedSum (mnorm=sum, wnorm=sum) - Solution:

ALT./CRIT.	mpg (max)	displacement (max)	weight (min)	acceleration (max)	price (min)	Rank
chevrolet chevelle malibu	18	307	3504	12	25561.6	3
buick skylark 320	15	350	3693	11.5	24221.4	2
plymouth satellite	18	318	3436	11	27240.8	4
amc rebel sst	16	304	3433	12	33685	6
ford torino	17	302	3449	10.5	20000	1
ford galaxie 500	15	429	4341	10	30000	8
chevrolet impala	14	454	4354	9	35764.3	10
plymouth fury iii	14	440	4312	8.5	25899.5	5
pontiac catalina	14	455	4425	10	32882.5	9
amc ambassador dpl	15	390	3850	8.5	32617.1	7

qiymatlarni yig'indisi normalizatsiyasi bilan og'irlikdagi summa mantiqiga misol

Biz hatto yakuniy ballni `dec.e_.punktlari` va o'rinlarni `dec.rank_` bo'yicha eksport qilishimiz mumkin.

Taqqoslash

Keling, ma'lumotlar to'plamimizdagi turli xil qaror qabul qilish algoritmlari (turli parametrlarga ega) natijalarini taqqoslaylik. Buni amalga oshirish uchun men `weightedSum` va `weightedProduct` ilovalaridan foydalanaman (bir marta maksimal, keyin esa summa qiymatini normallashtirish bilan). Bundan tashqari, sukut bo'yicha minmaxni bajaradigan va normalizatsiyani olib tashlaydigan `normalize_data` funksiyasini amalga oshirdim. Keyin men chiqishda yig'indisi kombinatsiyasini qo'llayman.

```
# import
from skcriteria.madm import simple

# make a copy of original dataset
cars_data_copy = cars_data.copy()

# weighted sum, sumNorm
dm = simple.WeightedSum(mnorm="sum")
dec = dm.decide(criteria_data)
cars_data_copy.loc[:, 'rank_weightedSum_sumNorm_inverse'] = dec.rank_

# weighted sum, maxNorm
dm = simple.WeightedSum(mnorm="max")
dec = dm.decide(criteria_data)
cars_data_copy.loc[:, 'rank_weightedSum_maxNorm_inverse'] = dec.rank_

# weighted product, sumNorm
dm = simple.WeightedProduct(mnorm="sum")
dec = dm.decide(criteria_data)
cars_data_copy.loc[:, 'rank_weightedProduct_sumNorm_inverse'] = dec.rank_

# weighted product, maxNorm
dm = simple.WeightedProduct(mnorm="max")
dec = dm.decide(criteria_data)
cars_data_copy.loc[:, 'rank_weightedProduct_maxNorm_inverse'] = dec.rank_

# min max scale + mirror
cars_data_copy.loc[:, 'rank_weightedSum_minmaxScale_subtract'] = \
    pd.Series(normalize_data().sum(axis=1)).rank(ascending=False).astype(int)
```

5 xil ko'p mezonlarni hal qiluvchilar

Ba'zi fikrlar,

Ford Torino 4/5 hal qiluvchilar bo'yicha 1-o'rinni egallaydi (eng yuqori ballga ega avtomobil). Minmax Chevrolet Malibu-ni afzal ko'radi.

Impala - universal past darajali :(WeightedProduct-ning ikkala ilovasi ham barcha avtomobillarga bir xil reytingni beradi. Bu erda qiziq narsa yo'q.Ikkala og'irlikdagi so'mni qo'llash reytingidagi yuqori tafovutlar.

MinMax eng yaxshi 4 yigit uchun eng xilma-xil reytinglarni beradi.

Normalizatsiyani o'zgartirganda (summdan maksimalgacha) natijaning farqlanishining asosiy sababi asl ma'lumotlarga tarjima qilinganligi bilan bog'liq. Ushbu tarjima ma'lumotlar oralig'ini o'zgartiradi (masalan, x va y o'rtasidagi hamma narsani o'lchaydi) va teskari bo'lsa, chiziqilikni ham o'zgartiradi (aytaylik, dastlabki ma'lumotlardagi 1 ga teng qadamlar o'zgartirilgan ma'lumotlarga mos kelmaydi). Bu quyidagi natija bilan aniqroq bo'ladi,

```
# import
import numpy as np
from sklearn.preprocessing import minmax_scale

# create sample data
x = np.array(range(1, 10))

print("Original data: ", x)
print("\nApproach 1:")
print("MinMax: ", minmax_scale(x))
print("MinMax_subtract: ", 1 - minmax_scale(x))
print("\nApproach 2:")
print("sumNorm: ", np.round(x/sum(x), 2))
print(f"sumNorm_inverse: ", np.round(1/(x/sum(x)), 2))
print("\nApproach 3:")
print("maxNorm: ", np.round(x/max(x), 2))
print("maxNorm_inverse: ", np.round(1/(x/max(x)), 2))
```

```
Original data: [1 2 3 4 5 6 7 8 9]

Approach 1:
MinMax: [0.    0.125 0.25  0.375 0.5   0.625 0.75  0.875 1.    ]
MinMax_subtract: [1.    0.875 0.75  0.625 0.5   0.375 0.25  0.125 0.    ]

Approach 2:
sumNorm: [0.02 0.04 0.07 0.09 0.11 0.13 0.16 0.18 0.2 ]
sumNorm_inverse: [45.  22.5 15.  11.25 9.  7.5  6.43 5.62 5.  ]

Approach 3:
maxNorm: [0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.  ]
maxNorm_inverse: [9.  4.5 3.  2.25 1.8 1.5 1.29 1.12 1.  ]
```

normallashtirish va o'zgartirilgan ma'lumotlar uchun turli yondashuv

Bu erda kiritilgan ma'lumotlar 1 dan 9 gacha raqamlardan iborat (e'tibor bering, har qanday ketma-ket ikkita raqam orasidagi farq 1 ga teng, ya'ni qadam bir xil). Birinchi yondashuv (minmax) 0 dan 1 gacha bo'lgan ma'lumotlarni tarjima qiladi va qadam hali ham bir xil. Endi 2 va 3 yondashuvning minimallashtirish mantiqiga (_teskari) qarang. Bu yerda boshida (past asl qiymatlar) qadam oxirgi elementning deyarli yarmini tashkil qiladi, lekin oxiriga yaqin (yuqori asl qiymat) qadam juda kichik, hattoki. Garchi asl ma'lumotlarda biz bir xil 1 qadam bilan harakat qilyapmiz.Shu sababli, minimallashtirilgan taqdirda, "yaxshi" avtomobillarga (past qiymatlarga ega) va hatto kichik nopoklik moddasiga (minimallashtirilganda, yuqori qiymat = past ball) juda yuqori ball beriladi va natijaning keskin pasayishiga olib keladi. Biz juda sinchkovlik qilayotganga o'xshaymiz, yo siz eng yaxshisiz yoki siz yarim

ball olasiz :) Boshqa tomondan, yuqori qiymatlar uchun kichik iflosliklar muhim emas. Agar mashina bu atribut bilan allaqachon yomon bo'lsa, unda uning qiymati 7 yoki 8 yoki 9 bo'ladimi va ballning kamayishi juda kam bo'ladimi, bizni qiziqtirmaydi! Biz ushbu tushunchadan o'z ehtiyojimizga ko'ra to'g'ri parametr bilan to'g'ri qaror echuvchini tanlash uchun foydalanishimiz mumkin.

Xulosa

Ushbu maqola ko'p mezonli qaror qabul qilish sohasi yuzasiga tegishli. Hatto skriteria to'plamida ham TOPSIS va MOORA kabi ko'plab algoritmlar mavjud bo'lib, ular bu muammolarni hal qilish uchun mutlaqo boshqacha sezgiga ega. Ammo shunga qaramay, ularning ko'pchiligida yaxshilik tushunchasi va ularning barchasini birlashtirish uchun individual xususiyatlarni boshqarish g'oyasi qo'llaniladi. Shunday qilib, ehtimol biz boshqa maqolada ko'proq algoritmlarni o'rganamiz. Ammo ushbu maqoladan asosiy xulosalar qaror qabul qiluvchilarning nima uchun va nima ekanligini tushunish bo'lishi kerak. Har bir bunday qaror bir nechta mezonlar bilan boshqarilishi mumkin. Shuningdek, biz har bir mezon uchun ajratilgan yaxshilik va ahamiyat haqida turli xil tushunchaga ega bo'lishimiz mumkin. Va nihoyat, bizda mantiq va parametrlarni almashtirish orqali yaratish mumkin bo'lgan turli xil hal qiluvchi turlari mavjud va ularning deyarli barchasi bizning ehtiyojlarimizdan kelib chiqqan holda turli va qiziqarli natijalar beradi.

FOYDALANILGAN ADABIYOTLAR:

1. Aruldoss, M.; Lakshmi, M.T.; Venkatesan, V.P. A survey on multi criteria decision making methods and its applications. *Am. J. Inf. Syst.* 2013, 1, 31–43. [Google Scholar]
2. Velasquez, M.; Hester, P.T. An analysis of multi-criteria decision making methods. *Int. J. Oper.* 2013, 10, 56–66. [Google Scholar]
3. Hajduk, S. Multi-Criteria Analysis in the Decision-Making Approach for the Linear Ordering of Urban Transport Based on TOPSIS Technique. *Energies* 2021, 15, 274. [Google Scholar] [CrossRef]