KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Intelektikos pagrindai (P176B101)

Antro laboratorinio darbo ataskaita

Atliko:

IFF-1/1 gr. Studentas

Vytenis Kriščiūnas

Priėmė:

lekt. Nečiūnas Audrius

lekt. Budnikas Germanas

KAUNAS 2024

TURINYS

1. Pasirinktas duomenų rinkinys

Naudojau tą patį duomenų rinkinį, kaip ir pirmojo laboratorinio darbo metu su normalizuotais duomenimis.

2. Pasirinkti atributai

Įvesties atributai: „wait“ ir „travel“. Jie šie atributai padeda rasti geriausius prognozuojamus rezultatus.

Išvesties prognozuojamas atributas: „mode“, kurio kardinalumas lygus 4.

3. Turimo duomenų rinkinio suskaidymas

Duomenų rinkinį suskaidžiau į 70-30% dalis, atitinkamai: apmokymo imtis ir testavimo imtis.

	A	B	C	D	E	F	G	H	I	J	K
1	rowname	individual	mode	choice	wait	vcost	travel	gcost	income	size	
2	1	1	air	no	0.69697	0.320225	0.02687	0.167364	0.471429	1	
3	2	1	train	no	0.343434	0.162921	0.224401	0.171548	0.471429	1	
4	3	1	bus	no	0.353535	0.129213	0.257081	0.167364	0.471429	1	
5	4	1	car	yes	0	0.044944	0.084967	0	0.471429	1	
6	5	2	air	no	0.646465	0.314607	0.003631	0.158996	0.4	2	
7	6	2	train	no	0.444444	0.162921	0.211329	0.225941	0.4	2	
8	7	2	bus	no	0.535354	0.129213	0.244009	0.230126	0.4	2	
9	8	2	car	yes	0	0.050562	0.139434	0.083682	0.4	2	
10	9	3	air	no	0.69697	0.634831	0.045025	0.414226	0.542857	1	
11	10	3	train	no	0.343434	0.539326	0.602033	0.690377	0.542857	1	
12	11	3	bus	no	0.353535	0.286517	0.594771	0.497908	0.542857	1	
13	12	3	car	yes	0	0.117978	0.477124	0.297071	0.542857	1	

	A	B	C	D	E	F	G	H	I	J
1	rowname	individual	mode	choice	wait	vcost	travel	gcost	income	size
2	588	147	car	no	0	0.258427	0.567901	0.460251	0.028571	1
3	589	148	air	no	0.646465	0.567416	0.073348	0.410042	0.542857	2
4	590	148	train	yes	0.10101	0.219101	0.484386	0.51046	0.542857	2
5	591	148	bus	no	0.535354	0.230337	0.625999	0.640167	0.542857	2
6	592	148	car	no	0	0.095506	0.606391	0.523013	0.542857	2
7	593	149	air	no	0.646465	0.792135	0.0748	0.577406	0.542857	1
8	594	149	train	yes	0.10101	0.455056	0.553377	0.74477	0.542857	1
9	595	149	bus	no	0.535354	0.382022	0.604938	0.736402	0.542857	1
10	596	149	car	no	0	0.269663	0.615832	0.661088	0.542857	1
11	597	150	air	yes	0.454545	0.516854	0.150327	0.389121	0.828571	1
12	598	150	train	no	0.343434	0.5	0.619463	0.669456	0.828571	1
13	599	150	bus	no	0.353535	0.247191	0.6122	0.476987	0.828571	1

4. Įvestys ir išvestys

Atributo „mode“ galima variantai: *car, air, bus* ir *train.* Juos bandau nuspėti naudodamas tolydinius duomenis iš „wait“ ir „travel“ stulpelių.

5. Sprendimų medžio sudarymas

Naudojau CART algoritmą, nes ID3 algoritmas gali dirbti tik su kategoriniais duomenimis, o man reikia ir tolydinių duomenimų. Šis algoritmas naudoja Gini indeksą, kuris nurodo, kaip dažnai atsitiktinai pasirinkti elementai būtų neteisingai išdėstyti sprendimų medyje.

6. Grafinis sprendinių medžio atvaizdavimas

Naudojau *graphviz* biblioteką sprendimų medžio formavimui.


wait ≤ 0.005
gini = 0.75
samples = 587
value = [147.0, 147.0, 146.0, 147.0]
class = air
True
False
gini = 0.0
samples = 146
value = [0, 0, 146, 0]
class = car
travel ≤ 0.144
gini = 0.667
samples = 441
value = [147, 147, 0, 147]
class = air
travel ≤ 0.132
gini = 0.027
samples = 148
value = [146, 1, 0, 1]
class = air
wait ≤ 0.52
gini = 0.503
samples = 293
value = [1, 146, 0, 146]
class = bus
gini = 0.0
samples = 143
value = [143, 0, 0, 0]
class = air
wait ≤ 0.551
gini = 0.56
samples = 5
value = [3, 1, 0, 1]
class = air
wait ≤ 0.379
gini = 0.457
samples = 218
value = [0, 77, 0, 141]
class = train
wait ≤ 0.545
gini = 0.149
samples = 75
value = [1, 69, 0, 5]
class = bus
wait ≤ 0.278
gini = 0.5
samples = 2
value = [0, 1, 0, 1]
class = bus
gini = 0.0
samples = 3
value = [3, 0, 0, 0]
class = air
wait ≤ 0.348
gini = 0.499
samples = 154
value = [0.0, 74.0, 0.0, 80.0]
class = train
travel ≤ 0.758
gini = 0.089
samples = 64
value = [0, 3, 0, 61]
class = train
gini = 0.0
samples = 68
value = [0, 68, 0, 0]
class = bus
travel ≤ 0.648
gini = 0.449
samples = 7
value = [1, 1, 0, 5]
class = train
gini = 0.0
samples = 1
value = [0, 0, 0, 1]
class = train
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
wait ≤ 0.323
gini = 0.227
samples = 92
value = [0, 12, 0, 80]
class = train
gini = 0.0
samples = 62
value = [0, 62, 0, 0]
class = bus
wait ≤ 0.449
gini = 0.061
samples = 63
value = [0, 2, 0, 61]
class = train
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
travel ≤ 0.206
gini = 0.278
samples = 6
value = [1, 0, 0, 5]
class = train
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
travel ≤ 0.208
gini = 0.375
samples = 48
value = [0, 12, 0, 36]
class = train
gini = 0.0
samples = 44
value = [0, 0, 0, 44]
class = train
gini = 0.0
samples = 52
value = [0, 0, 0, 52]
class = train
travel ≤ 0.526
gini = 0.298
samples = 11
value = [0, 2, 0, 9]
class = train
wait ≤ 0.626
gini = 0.5
samples = 2
value = [1, 0, 0, 1]
class = air
gini = 0.0
samples = 4
value = [0, 0, 0, 4]
class = train
wait ≤ 0.076
gini = 0.091
samples = 21
value = [0, 1, 0, 20]
class = train
travel ≤ 0.532
gini = 0.483
samples = 27
value = [0, 11, 0, 16]
class = train
gini = 0.0
samples = 6
value = [0, 0, 0, 6]
class = train
travel ≤ 0.542
gini = 0.48
samples = 5
value = [0, 2, 0, 3]
class = train
gini = 0.0
samples = 1
value = [0, 0, 0, 1]
class = train
gini = 0.0
samples = 1
value = [1, 0, 0, 0]
class = air
wait ≤ 0.035
gini = 0.375
samples = 4
value = [0, 1, 0, 3]
class = train
gini = 0.0
samples = 17
value = [0, 0, 0, 17]
class = train
travel ≤ 0.51
gini = 0.49
samples = 14
value = [0, 8, 0, 6]
class = bus
wait ≤ 0.227
gini = 0.355
samples = 13
value = [0, 3, 0, 10]
class = train
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
travel ≤ 0.568
gini = 0.375
samples = 4
value = [0, 1, 0, 3]
class = train
gini = 0.0
samples = 2
value = [0, 0, 0, 2]
class = train
gini = 0.5
samples = 2
value = [0, 1, 0, 1]
class = bus
travel ≤ 0.493
gini = 0.469
samples = 8
value = [0, 3, 0, 5]
class = train
travel ≤ 0.523
gini = 0.278
samples = 6
value = [0, 5, 0, 1]
class = bus
travel ≤ 0.542
gini = 0.5
samples = 6
value = [0, 3, 0, 3]
class = bus
gini = 0.0
samples = 7
value = [0, 0, 0, 7]
class = train
gini = 0.0
samples = 2
value = [0, 0, 0, 2]
class = train
travel ≤ 0.586
gini = 0.5
samples = 2
value = [0, 1, 0, 1]
class = bus
travel ≤ 0.337
gini = 0.48
samples = 5
value = [0, 3, 0, 2]
class = bus
gini = 0.0
samples = 3
value = [0, 0, 0, 3]
class = train
travel ≤ 0.517
gini = 0.444
samples = 3
value = [0, 2, 0, 1]
class = bus
gini = 0.0
samples = 3
value = [0, 3, 0, 0]
class = bus
gini = 0.0
samples = 2
value = [0, 0, 0, 2]
class = train
travel ≤ 0.564
gini = 0.375
samples = 4
value = [0, 3, 0, 1]
class = bus
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
gini = 0.0
samples = 1
value = [0, 0, 0, 1]
class = train
travel ≤ 0.218
gini = 0.444
samples = 3
value = [0, 1, 0, 2]
class = train
gini = 0.0
samples = 2
value = [0, 2, 0, 0]
class = bus
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
gini = 0.5
samples = 2
value = [0, 1, 0, 1]
class = bus
travel ≤ 0.553
gini = 0.5
samples = 2
value = [0, 1, 0, 1]
class = bus
gini = 0.0
samples = 2
value = [0, 2, 0, 0]
class = bus
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
gini = 0.0
samples = 2
value = [0, 0, 0, 2]
class = train
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = bus
gini = 0.0
samples = 1
value = [0, 0, 0, 1]
class = train

7. Sprendinių medžio testavimas

Prognozavimo tikslumui nustatyti naudojau tikslumo metiką:

$$tikslumas = \frac{teisingų\ spėjimų\ skaičius}{visas\ spėjimų\ skaičius}$$

Prognozavimo tikslumas:

```
Prognozavimo tikslumas: 0.9565217391304348
```

Sumaišymo matrica (stulpelis nurodo tikrąją atributo reikšmę, o eilutė spėjamas reikšmes):

```
Sumaišymo matrica:
[[61  0  0  2]
 [ 3 55  0  5]
 [ 0  0 64  0]
 [ 0  1  0 62]]
```

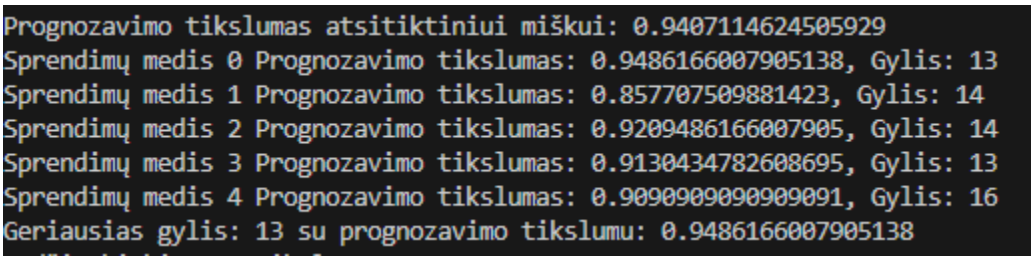
8. Sprendimų medžio gylio keitimas

Galima pastebėti, kad gyliui augant auga ir jo formavimo trukmė. Tikslumas pagerėja pasiekus pakanamai dedidelį gylį ir praktiškai nusistovi, taigi labai didelis medžio gylis nesuteikia jokio pranašumo prognozavimo tikslumo prasme.

```
Rezultatai:
Gylis   Formavimo trukmė (s)\Tikslumas
2       0.004044532775878906            0.7272727272727273
4       0.003522634506225586            0.8458498023715415
8       0.0035212039947509766           0.9604743083003953
16      0.003018617630004883            0.9525691699604744
32      0.003017425537109375            0.9604743083003953
64      0.005030393600463867            0.9565217391304348
128     0.007094860076904297            0.9565217391304348
256     0.009632110595703125            0.9525691699604744
```

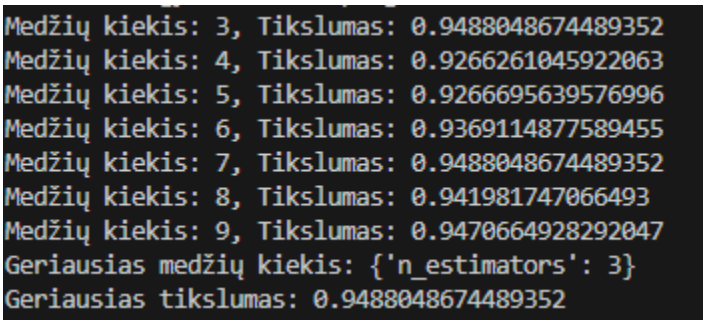
9. Atsitiktinio miško sudarymas

Galima pastebėti, kad atsitiktinio miško prognozės tikslumas yra šiek tiek mažesnis nei vieno iš sprendimų medžio, taip yra todėl, kad medžių kiekis bei mažesnio tikslumo medžiai gali nulemti neteisingų rezultatų priėmimą (vote). Vis dėlto gautas tikslumas panaudojus atsitiktinį mišką dažnai yra didesnis už pavienius medžius.



10. Atsitiktinį mišką sudarančių medžių kiekio keitimas

Mažiausiai medžių turintis atsitiktinis miškas – 3, gavo geriausią prognozės rezultatą. Galima daryti išvadą, kad atsitiktinai suformuotų medžių kiekis ir jų rūšis lemia atsitiktinio miško prognozės tikslumą.



11. Sprendinių medžio ir atsitiktinio miško palyginimas

Sprendinių medžio tikslumas šiuo atvėju gavosi geresnis nei atsitiktinio miško, tai galėjo nulemti medžių skirtumai ir atsitiktiniame miške naudotų sprendimų medžių duomenų pasikartojimas – kuo yra daugiau vienodų duomenų tuo sunkiau priimti korektiškus sprendimus.

Suformavus naują atsitiktinį mišką yra gaunamas geresnis tikslumas už pavienį sprendimų medį:

```
Prognozavimo tikslumas: 0.9565217391304348
```

```
Prognozavimo tikslumas atsitiktiniui miškui: 0.9644268774703557
```