1. domača naloga: Rudarjenje podatkov

Rudarjenje podatkov je sistematično iskanje informacij v veliki količini podatkov. Pri tem želimo z uporabo polavtomatskih ali avtomatskih postopkov zgraditi modele, ki kar najbolje povezujejo podatke med seboj. Z zgrajenimi modeli lahko iz znanih podatkov ocenimo vrednosti neznanih podatkov, podatke uvrščamo v razrede, napovedujemo prihodnje obnašanje in podobno. Mnogi modeli podatkovnega rudarjenja se pri iskanju povezav med podatki pomagajo z entropijo. Model podatkov bomo z uporabo entropije zgradili tudi sami. V praksi model gradimo iz podatkov v učni množici, njegovo uspešnost pa preverjamo na podatkih iz testne množice, ki jih nismo uporabili med gradnjo drevesa. Da postopek poenostavimo, bomo v tej nalogi uspešnost modela preverjali kar na učni množici.

Podatki

Podatki v učni množici so predstavljeni z množico zapisov. Vsak zapis vključuje vrednosti N vhodnih diskretnih značilnic ali atributov, $A_1, A_2, \ldots, A_N$, iz katerih želimo kar najbolj enostavno določiti razred $R \in \{r_1, \ldots, r_n\}$ zapisa. Pri tem značilnica A_z zavzame eno od n_z vrednosti, $A_z \in \{a_{z1}, \ldots, a_{zn_z}\}$. Podatki v tabeli 1 prikazujejo vpliv vremenskih dejavnikov (značilnice) na odločitev za kolesarjenje (razred). Vsaka značilnica lahko zavzame eno od dveh vrednosti: Nebo $\in \{\text{j(asno)}, \text{o(blačno)}\}$, Temp(eratura) $\in \{\text{t(oplo)}, \text{h(ladno)}\}$, Veter $\in \{\text{v(etrovno)}, \text{m(irno)}\}$, zapise želimo uvrščati v dva razreda Kolo $\in \{\text{d(a)}, \text{n(e)}\}$.

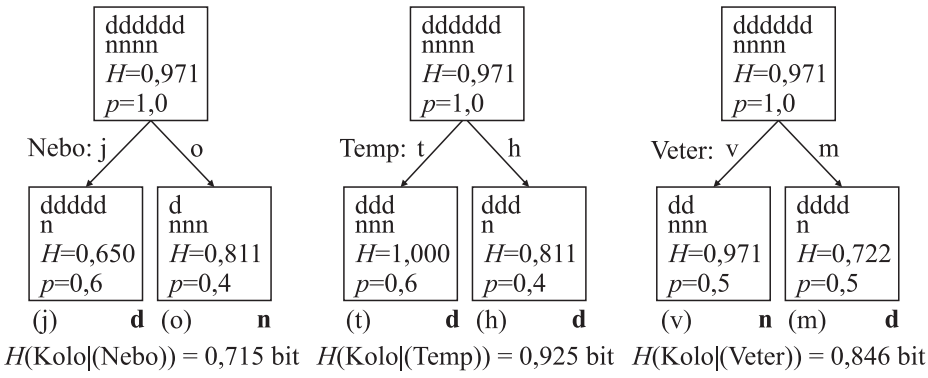
Tabela 1: Vpliv vremenskih podatkov na kolesarjenje.

Nebo	j	j	j	j	j	j	o	o	o	o
Temp	h	h	t	h	h	t	t	t	t	t
Veter	v	v	v	m	m	m	v	v	m	m
Kolo	d	n	d	d	d	d	n	n	d	n

Gradnja drevesa

V nalogi bomo zgradili poenostavljeno odločitveno drevo [1], namenjeno uvrščanju podatkov v naprej določene razrede. Odločitveno drevo bomo gradili po korakih. Najprej bomo med vsemi vhodnimi značilnicami izbrali tisto, ki najbolje loči razrede med seboj. Zapise ločimo na podlagi vrednosti izbrane značilnice, kot prikazuje slika 1. Najbolj všečna je leva razdelitev, saj so v listih večinoma prisotni zapisi, ki sodijo v isti razred. Ta razdelitev je najbolj določena in ima zato najmanjšo entropijo. Postopek lahko v naslednjem koraku ponovimo na listih.

Postopek izbiranja formalizirajmo. Vzemimo, da smo v prvem koraku izbrali značilnico A_1^* s pripadajočim številom različnih vrednosti n_1^*. Če z delitvijo še nismo zadovoljni, v drugem koraku že izbrani značilnici dodamo še eno. Drugo značilnico A_2^* spet izberemo tako, da imajo listi, ki jih oblikujemo s parom značilnic $A_1^* A_2^*$, čim več zapisov, ki pripadajo



Slika 1: Izbiranje značilnic, prvi korak.

istemu razredu. Po $k-1$ korakih ima odločitveno drevo $\Pi_{i=1}^{k-1} n_i^*$ listov, ki jih označimo z vrednostmi izbranih značilnic $(A_1^*, A_2^*, \ldots, A_{k-1}^*)$.

Vzemimo, da je razred R naključna spremenljivka in da je $p(r_i)$ verjetnost, da zapis pripada razredu r_i. Potem lahko nedoločenost razreda zapišemo kot

$$H(R) = -\sum_{i=1}^{n} p(r_i) \log_2 p(r_i) \quad . \tag{1}$$

V vsakem koraku gradnje drevesa želimo z delitvijo po izbranih značilnicah nedoločenost zmanjšati. V k-tem koraku značilnico izberemo iz nabora še neuporabljenih značilnic $A_z \in \{A_1, \ldots, A_N\}$, $A_z \notin \{A_1^*, \ldots, A_{k-1}^*\}$. Za vsako značilnico A_z v drevesu iz prejšnjega koraka liste spremenimo v notranja vozlišča s po n_z novimi listi. Dobimo drevo z $L_z = n_z \Pi_{i=1}^{k-1} n_i^*$ listi.

Glede na vrednosti značilnic $(A_1^*, \ldots, A_{k-1}^*, A_z)$ vse zapise razporedimo v liste $l_{zj}^k \in \{l_{z1}^k, \ldots, l_{zL_z^k}^k\}$. Za značilnico A_z ločeno izračunamo nedoločenost razreda za vsakega od listov,

$$H(R|l_{zj}^k) = -\sum_{i=1}^{n} p(r_i|l_{zj}^k) \log_2 p(r_i|l_{zj}^k) \quad , \tag{2}$$

pri čemer je $p(r_i|l_{zj}^k)$ delež zapisov v listu l_{zj}^k, ki pripada razredu r_i. Nedoločenosti se po listih razlikujejo, zato za vrednotenje značilnic uporabimo njihovo uteženo povprečje,

$$H(R|(A_1^*, \ldots, A_{k-1}^*, A_z)) = \sum_{j=1}^{L_z^k} p(l_{zj}) H(R|l_{zj}^k) \quad , \tag{3}$$

kjer je $p(l_{zj})$ delež vseh zapisov, ki so razvrščeni v list l_{zj}. Nazadnje med vsemi značilnicami izberemo tisto, ki ima najmanjšo vrednost uteženega povprečja,

$$A_k^* = \arg\min_{A_z} H(R|(A_1^*, \ldots, A_{k-1}^*, A_z)) \quad . \tag{4}$$

Ko v koraku K zaključimo z gradnjo drevesa, vsakemu listu priredimo razred R, v katerega sodi največ zapisov v listu.

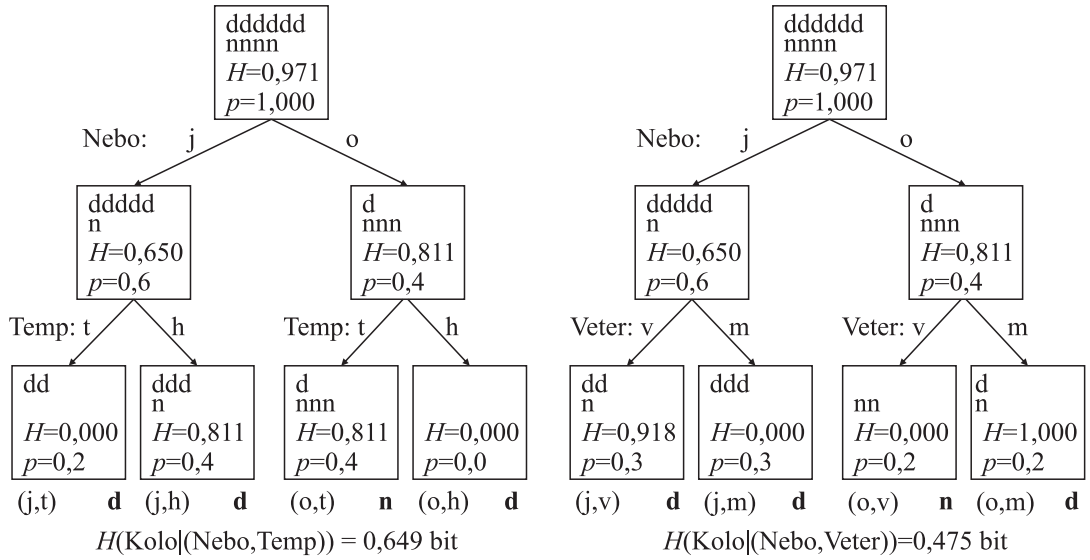
Primer: prvi korak

Za zapise iz tabele 1 zgradimo odločitveno drevo. V razred d sodi šest zapisov, v razred n pa štirje, tako je nedoločenost $H(\text{Kolo}) = H(6/10, 4/10) = 0{,}971$ bit. Zapise poskusimo razdeliti glede na vse tri značilnice. Delitev po značilnici Nebo nam da dva lista, (j) in (o). Kot vidimo na sliki 1, dobimo v listu (j) pet zapisov iz razreda d in en zapis iz razreda n, kar da nedoločenost lista $H(\text{Kolo}|(\text{j})) = H(5/6, 1/6) = 0{,}650$ bit. Na enak način izračunamo nedoločenost v listu (o), $H(\text{Kolo}|(\text{o})) = H(1/4, 3/4) = 0{,}811$ bit. V listu (j) imamo skupaj šest zapisov, $p((\text{j})) = 0{,}6$, v listu (o) pa štiri, $p((\text{o})) = 0{,}4$. Povprečna nedoločenost je tako $H(\text{Kolo}|(\text{Nebo})) = p((\text{j}))H(\text{Kolo}|(\text{j})) + p((\text{o}))H(\text{Kolo}|(\text{o})) = 0{,}715$ bit.

Na enak način izračunamo še povprečne nedoločenosti za značilnici Temp in Veter. Vidimo, pa je povprečna entropija najmanjša za Nebo, zato za najpomembnejšo značilnico izberemo $A_1^* = \text{Nebo}$. V listu (j) je največ zapisov iz razreda d, zato listu priredimo razred d (na sliki odebeljen). Iz istega razloga listu (o) priredimo razred n.

Primer: drugi korak

Potem, ko smo v prvem koraku izbrali značilnico $A_1^* = \text{Nebo}$, lahko v drugem koraku izbiramo samo še med značilnicama Temp in Veter. Zapise iz listov v prvem koraku razdelimo še glede na preostali značilnici. Za značilnici Temp in Veter dobimo razporeditvi, prikazani na sliki 2.



Slika 2: Izbiranje značilnic, drugi korak.

V desnem drevesu smo zapise uvrstili v štiri liste, katerih imena so določena iz vrednosti značilnic Nebo in Veter: (j,v), (j,m), (o,v) in (o,m). Število zapisov v posameznem listu določa verjetnost za list: $p((\text{j}, \text{v})) = p((\text{j}, \text{m})) = 0{,}3$, $p((\text{o}, \text{v})) = p((\text{o}, \text{m})) = 0{,}2$. V vsakem listu iz števila zapisov, ki sodijo v razred d, oziroma v razred n, določimo še entropijo $H((\text{j}, \text{v})) = H(2/3, 1/3) = 0{,}918$ bit, $H((\text{j}, \text{m})) = H(3/3, 0/3) = 0$ bit, $H((\text{o}, \text{v})) = H(0/2, 2/2) = 0$ bit in $H((\text{o}, \text{m})) = H(1/2, 1/2) = 1$ bit. Uteženo povprečje entropij vseh štirih listov nam da

$H(\text{Kolo}|(\text{Nebo}, \text{Veter})) = 0{,}475$ bit. Na enak način postopamo tudi za značilnico Temp. V drugem koraku izberemo značilnico $A_2^* = \text{Veter}$, s katero dobimo manjšo entropijo.

Listom priredimo razrede glede na večinsko zastopanost zapisov. Pri listu (o,m) izberemo katerikoli razred, saj sta enako zastopana.

Uporaba modela

Zdaj, ko je odločitveno drevo zgrajeno, lahko za poljubne zapise preverimo, kako dobro model razvršča. Glede na vrednosti značilnic se sprehodimo po odločitvenem drevesu in zapisu določimo razred, ki smo ga prej priredili listu. Uspešnost modela ovrednotimo z deležem pravilno uvrščenih zapisov ali točnostjo (angl. accuracy).

Primer: uvrščanje z značilnico Nebo

Najprej uporabimo rešitev, ki smo jo dobili v prvem koraku. V tabeli 2 iz značilnice Nebo zapisom določimo pričakovani razred. Zapisani so v vrstici z oznako Kolo*. Zapisi, pri katerih ima značilnica Nebo vrednost j, pripadajo vozlišču (j), zato jim določimo razred d. Zapisom z vrednostmi o pa iz istega razloga določimo razred n. Vidimo, da smo z

Tabela 2: Vpliv vremenskih podatkov na kolesarjenje.

Nebo	j	j	j	j	j	j	o	o	o	o
Kolo*	d	d	d	d	d	d	n	n	n	n
Kolo	d	n	d	d	d	d	n	n	d	n

odločitvenim drevesom pravilno uvrstili osem zapisov, dva pa narobe. Zaključimo, da je točnost odločitvenega drevesa enaka

$$\eta = \frac{8}{10} = 0{,}8 \quad . \tag{5}$$

Primer: uvrščanje z značilnicama Nebo in Veter

Postopamo podobno kot prej, le da zdaj razred določamo glede na vrednost značilnic Nebo in Veter. Uvrstitve v razrede prikazuje tabela 3 v vrstici z oznako Kolo**. Tudi v tem

Tabela 3: Vpliv vremenskih podatkov na kolesarjenje.

Nebo	j	j	j	j	j	j	o	o	o	o
Veter	v	v	v	m	m	m	v	v	m	m
Kolo**	d	d	d	d	d	d	n	n	d	d
Kolo	d	n	d	d	d	d	n	n	d	n

primeru je točnost

$$\eta = \frac{8}{10} = 0{,}8 \quad . \tag{6}$$

Ta model je zagotovo boljši od prejšnjega, saj je povprečna nedoločenost bistveno manjša. Na žalost je nabor zapisov takšen, da se kvaliteta modela ne odraža tudi v točnosti.

Naloga

V datoteki `naloga1.py` v jeziku Python napišite funkcijo z imenom `naloga1`, ki določi:

- povprečno entropijo $H(R|A_K^*)$ za zgornji nabor (`entropija`) in
- točnost odločitvenega drevesa (`tocnost`).

Vhodni argumenti funkcije so

- slovar značilnic s pripadajočimi seznami vrednosti (`znacilnice`),
- seznam razredov (`razredi`) in
- število korakov (`koraki`).

Prototip funkcije:

```
def naloga1(znacilnice, razredi, koraki):

    entropija = float('inf')
    tocnost = float('inf')

    return (entropija, tocnost)
```

Testni primeri
Na učilnici se nahajajo trije testni primeri podatkov, za katere imate podane tudi rešitve: povprečno informacijo `entropija` (v bitih) in točnost `tocnost`. Podatki so podani v obliki datotek `.json`. Priloženo imate tudi funkcijo `test_naloga1`, ki jo lahko uporabite za preverjanje pravilnosti rezultatov. Pri testiranju vaših funkcij upoštevajte naslednje omejitve:

- rezultat je pravilen, če se od danega razlikuje za manj kot 10^{-4},
- izvajanje funkcije je časovno omejeno na 30 sekund.
- Vaš program lahko uporablja samo tiste pakete, ki so del standardne knjižnice Python 3.11 (https://docs.python.org/3.11/library/) in paketa `numpy` ter `scipy`. Na sistemu za preverjanje vaših rešitev drugi paketi niso nameščeni.

Točkovanje

Pri vsaki nalogi dobite pol točke za pravilen izračun entropije in pol točke za pravilen izračun točnosti.

Namigi

Delajte sami

Pri preverjanju nalog uporabljamo tudi sistem za zaznavanje plagiatorstva. Verjemite, da prepisovanje težko zakrijete.

Uporabne funkcije in razredi

- `zip`,
- `sum`,
- `math.log2`,
- `collections.Counter`

Literatura

[1] D.G. Luenberger: Information Science, Princeton University, str. 43-44, 2006.