

Intelligentni sistemi, 1. seminarska naloga

Matic Bernik in Robert Tovornik

5. december 2016

1 Uvod

Namen seminarske naloge je, da na podlagi podanih podaktov tekem NBA, zgradimo napovedni model, ga preverimo, ter z njim napovemo zmagovalca nove, prihodnje tekme, ter končno razliko točk v koših med ekipama.

2 Podatki

V okviru seminarske naloge, so nam bili podani podatki tekem v NBA za pretekli sezoni 2008/09 ter 2009/10. Ti se nahajajo v dveh datotekah:

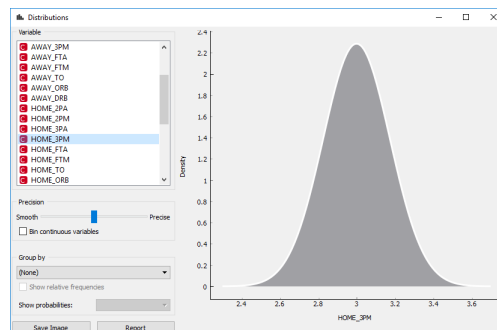
-nba0809.txt

-nba0910.txt

Skupaj obsegajo 2459 primerov, ter so opisani z 31 začetnimi atributi. Med podatki ni manjkajoči vrednosti (N/A).

3 Pregled podatkov

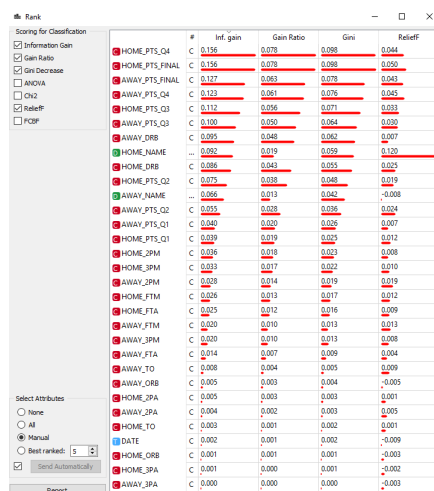
Ob začetku, sva pregledala razporeditev podanih podatkov, s pomočjo orodja ORANGE (razvit na FRI), ter ugotovila, da so podatki razporejeni normalno, kar je zaželeno (razen podatkov o home in away team).



Slika 1: Prikaz razporeditve posamezni podatkov v orodju Orange (distributions).

4 Iskanje napovednih atributov

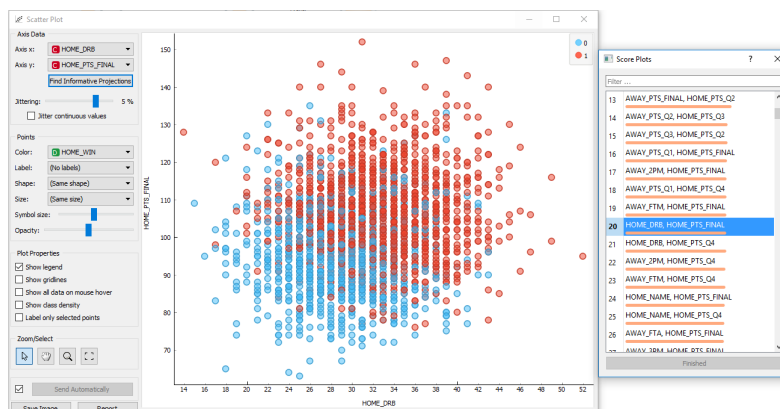
Da bi lažje določila pomembnost atributov, ki določajo rezultat tekme, sva uporabila orodje Orange, za izračun teže informacije, oziroma ocene koliko določen atribut doprinese h končni napovedi. Le-te sva izračunala z ocenami atributov: "Information gain", "Gain Ratio", "Gini" ter "ReliefF".



#	Attribute	Info. gain	Gain Ratio	Gini	ReliefF
1	HOME_PTS_Q4	0.156	0.078	0.088	0.044
2	HOME_PTS_FINAL	0.156	0.078	0.088	0.050
3	AWAY_PTS_FINAL	0.127	0.063	0.078	0.043
4	AWAY_PTS_Q4	0.123	0.061	0.076	0.045
5	HOME_PTS_Q3	0.112	0.056	0.071	0.033
6	AWAY_PTS_Q3	0.100	0.050	0.064	0.030
7	AWAY_DRB	0.095	0.048	0.062	0.007
8	HOME_NAME	0.092	0.019	0.059	0.130
9	HOME_DRB	0.088	0.041	0.055	0.024
10	HOME_PTS_Q2	0.075	0.038	0.048	0.019
11	AWAY_NAME	0.066	0.013	0.042	-0.008
12	AWAY_PTS_Q2	0.053	0.028	0.056	0.024
13	AWAY_PTS_Q1	0.049	0.020	0.026	0.007
14	HOME_PTS_Q1	0.039	0.019	0.029	0.012
15	HOME_3PM	0.036	0.018	0.023	0.008
16	HOME_3PM	0.033	0.017	0.022	0.010
17	AWAY_3PM	0.028	0.014	0.019	0.019
18	HOME_FTM	0.026	0.013	0.017	0.012
19	HOME_FTA	0.025	0.012	0.016	0.009
20	AWAY_FTM	0.023	0.010	0.013	0.013
21	AWAY_3PM	0.020	0.010	0.013	0.008
22	AWAY_FTA	0.014	0.007	0.009	0.004
23	AWAY_TO	0.008	0.004	0.005	0.009
24	AWAY_ORB	0.005	0.003	0.004	-0.005
25	HOME_3PA	0.005	0.003	0.003	0.001
26	AWAY_3PA	0.004	0.002	0.003	0.005
27	HOME_TO	0.003	0.001	0.002	0.001
28	DATE	0.002	0.001	0.002	-0.009
29	HOME_ORB	0.001	0.001	0.001	-0.003
30	HOME_3PA	0.001	0.000	0.001	-0.002
31	AWAY_3PA	0.000	0.000	0.000	-0.003

Slika 2: Ocena teže oziroma informacijskega doprinosa atributov k napovedi.

Presenetilo naju je, da poleg uspešnosti zadevanja koša, močno vplivajo tudi nekateri drugi atributi, med katerimi močno izstopa atribut "Defensive Rebounds". Po dodatnem testu, s funkcionalnostjo scatter plot, se je izkazalo, da močno nakazuje na končno število zadetih košev.

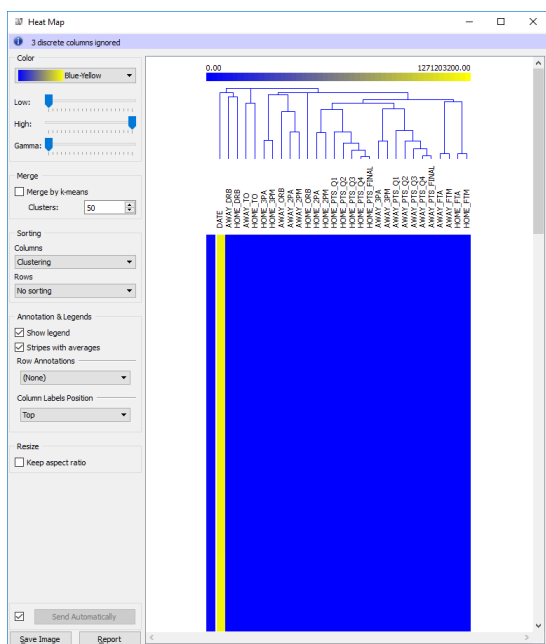


Slika 3: Scatter plot - relacija med "Defensive rebounds" in "Final points scored".

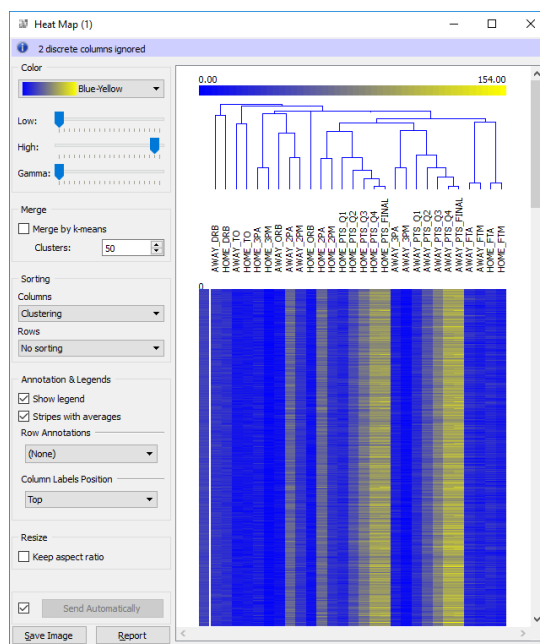
5 Ustvarjanje novih atributov

Ko sva ocenila informacijske lastnosti atributov, sva se lotila iskanja možnosti združevanja in prepoznavanja sorodnosti atributov. Za to sva uporabila funkcionalnost orodja Orange, Heat-

Map, ter hkrati clustering in opazila anomalijo, ki jo povzroča datum, ki tekme unikatno določa. Zato sva le-tega odstranila iz učnih podatkov, ter ponovno zagnala prepoznavanje.



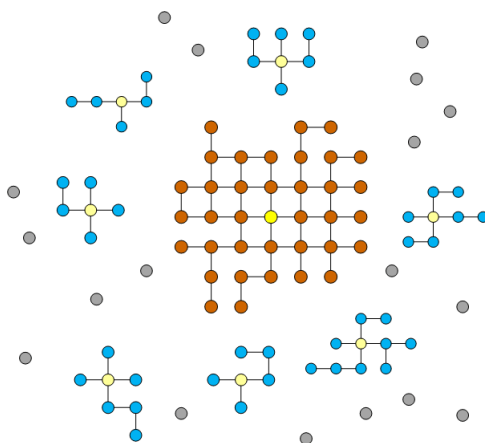
Slika 4: Heatmap z datumom



Slika 5: Heatmap, datum odstranjen

6 Metode

Tu opišeš, na kakšen način si rešil nalogo (tehnike in metode, ki si jih uporabil). Lahko vključiš tudi zanimiv del programske kode, ki si jo morda pri tem razvil ali pa v poročilo dodatno vključiš sliko, kot je na primer slika 6. Vse slike in tabele, ki jih vključiš v poročilo, morajo biti navedene v besedilu oziroma se moraš na njih sklicati.



Slika 6: Vsako sliko opremi s podnapisom, ki pove, kaj slika prikazuje.

V to poglavje lahko tudi vključiš kakšen metodološko zanimiv del kode. Primer vključitve kode oziroma implementirane funkcije v programskem jeziku Python je:

```
def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

Izris te kode je lahko sicer tudi lepši, poskušaš lahko najti še primernejši način vključevanja kode v Pythonu oziroma v tvojem izbranem programskem jeziku v okolje L^AT_EX.

7 Rezultati

V tem poglavju podaš rezultate s kratkim (enoodstavčnim) komentarjem. Rezultate lahko prikažeš tudi v tabeli (primer je tabela 1).

Odstavke pri pisanju poročila v L^AT_EX-u ločiš tako, da pred novim odstavkom pustiš prazno vrstico. Tudi, če pišeš poročilo v kakšnem drugem urejevalniku, morajo odstavki biti vidno ločeni. To narediš z zamikanjem ali pa z dodatnim presledkom.

Tabela 1: Atributi in njihove zaloge vrednosti.

ime spremenljivke	definijsko območje	opis
cena	[0, 500]	cena izdelka v EUR
teža	[1, 1000]	teža izdelka v dag
kakovost	[slaba—srednja—dobra]	kakovost izdelka

Podajanje rezultati naj bo primerno strukturirano. Če ima naloga več podnalog, uporabi podpoglavja. Če bi želel poročati o rezultatih izčrpno in pri tem uporabiti vrsto tabel ali grafov, razmisli o varianti, kjer v tem poglavju prikažeš in komentiraš samo glavne rezultate, kakšne manj zanimive detajle pa vključi v prilogo (glej prilogi A in B).

8 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.

Priloge

A Podrobni rezultati poskusov

Če je rezultatov v smislu tabel ali pa grafov v nalogi mnogo, predstavi v osnovnem besedilu samo glavne, podroben prikaz rezultatov pa lahko predstaviš v prilogi. V glavnem besedilu ne

pozabi navesti, da so podrobni rezultati podani v prilogi.

B Programska koda

Za domače naloge bo tipično potrebno kaj sprogramirati. Če ne bo od vas zahtevano, da kodo oddate posebej, to vključite v prilogo. Čisto za okus sem tu postavil nekaj kode, ki uporablja Orange (<http://www.biolab.si/orange>) in razvrščanje v skupine.

```
import random
import Orange

data_names = ["iris", "housing", "vehicle"]
data_sets = [Orange.data.Table(name) for name in data_names]

print "%10s_%3s_%3s_%3s" % (" ", "Rnd", "Div", "HC")
for data, name in zip(data_sets, data_names):
    random.seed(42)
    km_random = Orange.clustering.kmeans.Clustering(data, centroids = 3)
    km_diversity = Orange.clustering.kmeans.Clustering(data, centroids = 3,
        initialization=Orange.clustering.kmeans.init_diversity)
    km_hc = Orange.clustering.kmeans.Clustering(data, centroids = 3,
        initialization=Orange.clustering.kmeans.init_hclustering(n=100))
    print "%10s_%3d_%3d_%3d" % (name, km_random.iteration, \
        km_diversity.iteration, km_hc.iteration)
```