Izpit iz vaj (16. 1. 2024)

Tematika podatkov: Gladiatorske igre

Tokrat se boste srečali s tematiko, ki se navezuje na rimske čase in sicer na gladiatorske boje. To so bili boji, ki so se pretežno izvjali v arenah in so takrat predstavljali glavno zabavo ljudi. Gladiatorji so bili bojevniki, ki so se z orožjem pomerili med sabo. Večinoma so bili to sužnji ali pa sinovi propadlih družin. Glede na njihove karakteristike boste poskusili napovedati kdo izmed teh bojevnikov bo preživel.

Podatki

V štirih datotekah (`gladiator_personal_data.csv`, `gladiator_statistics.csv`, `gladiator_skills.xlsx`, `gladiator_status.txt`) se nahajajo podatki o gladiatorjih, njihovi bojni statistiki in veščinah. Prva datoteka vsebuje *splošne podatke o gladiatorju*, druga datoteka vsebuje *statistiko njegovih dosedanjih bojev*, tretja *podatke o njegovih lastnostih* in zadnja *podatke o tem ali je preživel ali ne*.

V pomoč pri razumevanju posameznih spremenljivk so vam lahko naslednji opisi ter vrsta podatkov (številski/kategorični podatek):

- Age - starost gladiatorja. *številski*
- Origin - od kod gladiator prihaja. *kategoričen*
- Height - višina gladiatorja. *številski*
- Weigh - teža gladiatorja. *številski*
- Wins - število do sedaj doseženih zmag. *številski*
- Losses - število porazov, ki jih je do sedaj doživel. *številski*
- Equipment Quality - kakovost njegove opreme (basic, superior, ...). *kategoričen*
- Public Favor - stopnja popularnosti gladiatorja med publiko. *številski*
- Injury History - zgodovina poškodb (low, high). *kategoričen*
- Mental Resilience - stopnja sposobnost gladiatorja, da prenese stres in psihološke pritiske. *številski*
- Tactical Knowledge - poznavanje bojnih taktik (advanced, expert, ...). *kategoričen*
- Battle Experience - koliko let bojevanja ima za sabo. *številski*
- Health Status - zdravstevno stanje (excellent, good, ...). *kategoričen*
- Personal Motivation - glavna motivacija za bojevanje (glory, survival, ...). *kategoričen*
- Previous Occupation - dejavnost s katero se je ukvarjal, preden je postal gladiator (laborer, criminal, soldier, ...). *kategoričen*
- Battle Strategy - prednostna strategija, ki jo gladiator uporablja v boju (balanced, aggressive, ...). *kategoričen*
- Crowd Appeal Techniques - tehnike, ki jih uporablja gladiator, da pritegne občinstvo (intimidating, humble, ...). *kategoričen*
- Survived - ali je gladiator preživel boje (yes, no). *kategoričen*

Naloga 1 *(15 T)*

V dataframe preberite vse štiri datoteke s podatki: `gladiator_personal_data.csv`, `gladiator_statistics.csv`, `gladiator_skills.xlsx` in `gladiator_status.txt`. Vse prebrane podatke iz datotek nato združite v en dataframe, glede na ime gladiatorja. Indeks stolpec naj bo poimenovan *GLADIATOR*.

- Na **dva**(!) različna načina izpišite **prve 3 vrstice** tega združenega datafram-a.
- izpišite koliko **stolpcev in vrstic** je v združenem datafram-u.
- Izpišite **podatkovne tipe** za stolpce od drugega do (vključno) petega.
- Izpišite koliko je posameznih **unikatnih vrednosti** v stolpcu Equipment Quality.
- Izpišite vse podatke za gladiatorja "Nero Menenius".
- Zapišite število gladiatorjev, ki so doživeli 4 ali 5 porazov (Losses). Rezultat poizvedbe mora biti številka!

Naloga 2 *(20 T)*

- Izrišite graf, ki bo prikazoval povprečno starost gladiatorjev, ki so umrli, glede na dejavnost s katero so se ukvarjali preden so postali gladiatorji (Previous Occupation).
- Izrišite graf, iz katerega bo razvidna mediana ter največje in najmanjše število zmag, ki so jih dosegli gladiatorji.
- Izrišite graf raztrosa (s črto), ki bo prikazoval višino gladiatorja glede na poljubno vrednost (izmed stolpcev sami izberite vrednost, ki bo dala smiseln rezultat), ločeno glede na zdravstveno stanje gladiatorja. Za vsako zdravstveno stanje naj bo prikazan ločen podgraf.
- V naraščajočem vrstnem redu izpišite povprečno težo gladiatorjev (zaokroženo na dve decimalki) glede na njihovo primarno izbrano bojno strategijo. Upoštevajte samo tiste gladiatorje, ki tekmujejo zaradi tega da bi si zagotovili svobodo ali preživetje (Personal Motivation).
- Koliko je takšnih gladatorjev, ki se na agresiven način (Battle Strategy) borijo zaradi maščevanja (Personal Motivation)?

Naloga 3 *(5 T)*

- Izpišite koliko je manjkajočih podatkov v posameznih stolpcih.
- Nato manjkajoče podatke iz stolpcev zapolnite s sledečo strategijo:
  - `Weight` zapolnite tako, da od višine specifičnega gladiatorja odštejete vrednost 100 (w=h-100),
  - `Mental Resilience` in `Public Favor` zapolnite s povprečno vrednostjo stolpca,
  - `Previous Occupation` zapolnite z vrednostjo "Criminal",
  - `Crowd Appeal Techniques` z najpogosteje pojavljeno vrednostjo stolpca,
  - ostale vrstice z manjkajočimi vrednostmi izbrišite.
- Ponovno izpišite koliko je manjkajočih vrednosti za stolpce `Weight`, `Mental Resilience`, `Public Favor`, `Previous Occupation` in `Crowd Appeal Techniques`.

Naloga 4 *(10 T)*

- Ustvarite dve kopiji dataframa `dfRegresija` in `dfKlasifikacija`:
  - `dfKlasifikacija` je dataframe, ki ga boste uporabili za klasifikacijo, in sicer boste napovedovali ali bo določen gladiator preživel boje (Survived).
  - `dfRegresija` je dataframe, ki ga boste uporabili za regresijo, in sicer boste napovedovali število zmag gladiatorja (Wins).
- Podatke v obeh dataframih **ustrezno predprocesirajte(!)** - kategorične vrednosti pretvorite z LabelEncoderjem, številske vrednosti pa morajo biti standardizirane.
- Izpišite zadnjih 5 vrstic iz vsakega dataframa.

Naloga 5 *(10 T)*

S pomočjo regresija poskusite napovedati koliko zmag bo dosegel posamezni gladiator (`Wins`). Za podatke uporabite predprocesiran dataframe `dfRegresija`. Iz vhodnih podatkov izpustite tudi podatek `Losses`. Velikost učne množice naj bo 75%. Na naključno stanje uporabite 789. Za regresor uporabite regresijsko drevo.

- Kako dobro se je naučil model ocenite s **povprečno absolutno napako** in **r2 score**. Oboje zaokrožite na eno decimalko.

Naloga 6 *(20 T)*

S pomočjo klasifikacije napovejte ali bo gladiator preživel spopade ali ne (`Survived`). Iz vhodnih podatkov odstranite še stolpec `Wins` in `Losses`. Podatke iz predprocesiranega `dfKlasifikacija` delite na učne in testne in sicer s pomočjo stratificirane delitve na 6 foldov.

Nad podatki preizkusite dva klasifikatorja - *naključni gozd* in *logistično regresijo*. Ker želimo doseči najvišjo možno točnost klasifikacije to izvedite s pomočjo iskanja najboljših nastavitev parametrov po principu naključnega iskanja (RandomizedSearchCV). Omejite ga na 5 iteracij.

Za nakjučni gozd preizkusite:

- naključna število dreves med 5 in 10,
- kriterij "gini" in "entropy".

Za logistično regresijo pa:

- penalty "l2" ali None.

Najboljše izračunane vrednosti točnosti za oba klasifkatorja prikažite v stolpičnem **grafu**.

Naloga 7 *(10 T)*

Za konec naredite še gručenje nad enakim datasetom, kot ste ga uporabili za regresijo. Podatke transformirajte s pomočjo FastICA dekompozicije. Kot algoritem gručenja uporabite KMeans.

- Da boste vedeli koliko je najbolj optimalno število gruč na katere je smiselno deliti podatke pred gručenjem izrišite **graf z izračunanimi inerciami** za od 1 do (vključno) 8 gruč, nad transformiranimi podatki. Po pravilu komolca iz grafa preberite najbolj optimalno število gruč in ga uporabite v algoritmu.
- Izrišite **graf**, v katerem prikažete **transformirane podatke**, ki so obarvani glede na **gručo**, v katero so razvrščeni.