UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna Matematika – 1. stopnja

Martin Kokošinek

Analiza nogometnih transferjev FC Barcelona (2000-2020)

Projektna naloga pri predmetu Programiranje 1

Predavatelj: prof. dr. Matija Pretnar

Ljubljana, 2020

Kazalo

Analiza nogometnih transferjev FC Barcelona (2000-2020)

Uvod

Pri predmetu programiranje 1 sem si za temo projektne naloge izbral analizo transferjev klub FC barcelona od sezone 2000/2001 do vključno sezone 2019/2020, brez poletnega prestopnega roka, saj v času pisanja, ta še ni bil zaključen. Kot posameznik, ki v prostem rad spremljam dogajanje na nogometnih zelenicah klub FC Barcelona, me je, glede na dogajanje zadnjih let, zanimalo kako potekajo transferji kluba. V sezoni 2016/2017, ko se je zgodil senzacionalen prestop Neymarja iz FC Barcelona v PSG, za ogromnih €222 mio., je cena prestopov na trgu igralec zelo zrasla. V luči tega dogodka, se mi je pojavilo nekaj vprašanj.

1. Kako je zrasla vrednost povprečenga nakupa po prodaji Neymarja (ali je zrasla?)
2. Število nakupov in prodaj (kako je bilo z nakupi v času 'zlate generacije' La Masie - manj?)
3. Ali Barcelona vedno več vlaga v nakupe in ali razlika med vrednostjo prihodov in odhodov raste?
4. So Brazilci v povprečju najdražji (pri prestopih) pri Barceloni?
5. Katere pozicije Barcelona največ 'kupuje' in katere 'prodaja'?

Za odgovor na ta in podobna vprašanja, sem potreboval pridobiti podatke iz tabel na straneh Wikipedie na url linkih oblike Transfer 2000-2001. Za ta link in vse nadaljne sem moral pridobiti tabelo prestopov in jo nato urediti oziroma očistiti. Podatke, ki sem jih želel izluščiti so sledeči:

1. Ime
2. Državljanstvo
3. Vloga na igrišču (pozicija)
4. Klub iz katerega je prišel oziroma v katerega je šel igralec
5. Cena prestopa

Tabele sem uvozil s pomočjo programskega jezika Python, knjižnice BeautifulSoup in priročnih orodij, ki nam jih je, za lažje delo, napisal profesor. Po urejanju tabel sem jih shranil v .csv datoteke in se lotil analize s pomočjo odprtokodne aplikacije Jupyter Notebook, kjer sem s knjižnico Pandas analiziral pridobljene podatke.

1 Uvoz podatkov

Projektna naloga je napisana v programskem jeziku python, kjer je glavni program za uvoz podatkov **Poberi_podatke.py**. Na začetku moramo shraniti html strani v neko mapo, da za vsako poizvedbo ne kličemo spletne strani ponovno. To izvedemo s pomočjo funkcije *shrani_spletno_stran*, ki je definirana v datoteki **orodja.py**. Sedaj je potrebno iz dobljenih html strani pridobiti tabele. Tabele so se glede na leto razlikovale, prav tako pa njihovo stevilo na strani, kar lahko vidimo na sliki **??**. Zato definiramo funkciji $table\_IN(i, soup)$ in $table\_OUT(i, soup)$, kjer bo **i** predstavljalo sezono (za 2000/2001 je $i = 2000$) in **soup** spletna stran pripravljena za branje s knjižnico BeautifulSoup.

```
def table_IN(i,soup): #kje se nahaja tabela wiki za transfer
    #i je letnica
    #soup je html datoteka pripravljena za beautifulsoup branje
    if i in [2000,2001,2002,2003,2004,2005,2009]:
        table = soup.find_all('table',class_='wikitable sortable', style="text-align: center;")[0]
        table = table.tbody
    elif i == 2013:
        table = soup.find_all('table',class_='wikitable sortable', style="text-align: center;")[2]
        table = table.tbody
    elif i == 2018:
        table = soup.find_all('table',class_='wikitable sortable')[1]
        table = table.tbody
    elif i == 2019:
        table = soup.find_all('table',class_='wikitable sortable')[2]
        table = table.tbody
    else:
        table = soup.find_all('table',class_='wikitable sortable', style="text-align: center;")[1]
        table = table.tbody
    return table
```

Slika 1: Zgled funkcije *table_IN*

Potem ko smo pridobili tabele, je bilo potrebno začeti iskati prave podatke, jih urediti in očistiti ter zapisati v tabelo primerno za shranjevanje v .csv datoteko.

1.1 Čiščenje podatkov

S pomočjo ukaza **vrsticeIN** $= tableIN.find\_all('tr')$, kjer **tableIN** predstavlja že pridobljeno tabelo, $find\_all('tr')$ pa poišče vse vrstice (table row) v tabeli. Nato s pomočjo ukaza **dfIN** $= pd.DataFrame(columns = stolpci)$ ustvarimo novo tabel (dataframe). Tu nam *pd* predstavlja okrajšavo za knjižnico Pandas, $columns =$ **stolpci** pa določi imena stolpcov, ki jih "pobere"iz seznama **stolpci**. Enako naredimo tudi za **vrsticeOUT** in **dfOUT**. Primer IN lahko vidimo na sliki **??**.

```
#dodam v BSoup
soup = BeautifulSoup(tekst, 'html.parser')

#poimenujem stolpce df
stolpci = ['Ime', 'Državljanstvo', 'Pozicija', 'Klub', 'Cena', 'Leto']

#table transfer IN
tableIN = table_IN(i,soup)
vrsticeIN = tableIN.find_all('tr')
dfIN = pd.DataFrame(columns=stolpci)
```

Slika 2: Zgled priprave tabele

1.2 Primer izluščevanja, čiščenja in shranjevanja podatkov

Sedaj lahko zaženemo funkcijo $ustvari\_df$(**vrsticeIN**, **dfIN**, **i**, _), kjer _ zavzame vrednost $'IN'$ ali $'OUT'$. V tej funkciji moramo za vsako vrednost **'tr'** poiskati vse vrednosti **'td'**. Zato izvedemo zanko for na intervalu od 1 do dolžine $vrsticeIN$. Ker se tabele po letih razlikujejo (2018 in 2019 sta drugačni), je bilo potrebno ločiti tabele po letu 2017 in tabele do vključno leta 2017 in odvisno od smeri transferja prav tako.

V poročilu bom predstavil primer za sestavljanje tabel prihodov do leta 2017.
Za pridobivanje vrednosti v tabelah uporabimo $vrednost = [k.textforkintd]$, kjer $k.text$ vrne vrednost, ki je v tabeli in ne celotne vrstice td. Tu se nam pojavijo prvi problemi, za podatek o državljanstvu imamo na voljo ne ikono zastavice države iz katere je igralec. Država se nahaja pod tretjim stolpcem torej **td**[**3**]. Problem rešimo tako, da poiščemo kje v $< td >$ se nahaja ime države in odkrijemo, da se skriva pod **title**=ImeDržave. Nato z nekaj operacijami izluščimo ime države in ga shranimo v seznam **vrednost**, ki vsebuje podatke, ki jih zbiramo.
Za ostale podatke je dela manj, a vseeno gre omeniti vrednost prestopa, za katero je potrebno določiti nek skupen format izbral sem zapis v obliki miljonov in decimalno piko (brez valut, vedno gre za €). Naslednja posebnost, ki se je pojavila je bila neštevilčna vrednost prestopa (Free, Loan, Undisclosed, Youthsystem, ...). Tu je potrebna obravnava za vsako posebnost posebej in izbrati najbol smiselno vrednost, ki ji pripada. V primeru Loan pa nogometaša odstranimo iz tabele, saj ta ne vpliva na ceno transferjev in ni dolgoročna okrepitev, torej analiza ni smiselna. Na slikah **??** in **??** lahko vidimo kako smo izluščili državljanstvo in dobili ostale vrednosti ter kako smo ocistili vrednost prestopov, shranili oziroma dodali podatke v tabelo $dfIN$ in jih, po koncu izvajanja for zanke, zapisali v csv datoteko.

```
#list posameznih vrednosti ('stolpcev')
vrednost = [k.text.replace('\n','') for k in td]

#drzava je le flagicon zato iscem iz title
drzava = str(td[2]).split('title=')
drzava = drzava[1].split('"><img alt=')
drzava = drzava[0].replace('"','')
```

Slika 3: Izluščevanje državljanstva in pridobivanje vrednosti

```
            #zapišemo v vrstico, če je prestop samo posoja ga ne upoštevam
            if str(vrednost[10]) not in ['N/A','Loan','–'] and str(vrednost[7]) not in ['Loan','Loan return']:
                vrednost = [vrednost[3], drzava, vrednost[1],vrednost[6],vrednost[10],i]
                df.loc[j-1] = vrednost

df.to_csv(str('.\\podatki\\'+str(i)+kam+'.csv'), index=False)
```

Slika 4: Čiščenje vrednosti prestopa in zapis v csv

2 Analiza podatkov

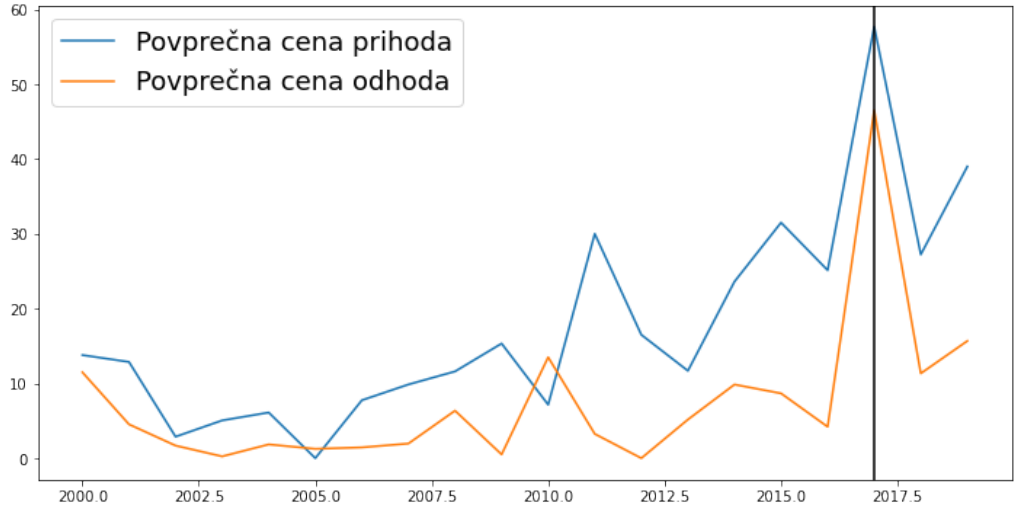
Sedaj, ko imamo podatke zbrane in očiščene ter shranjene v csv tabelah lahko začnemo z analizo. Najprej rabimo združiti tabele v skupni tabeli, eno za prihode in drugo za odhode kot je prikazano na sliki **??**.

```
#združimo csvje v 2 df-ja: prihod in odhod
stolpci = ['Ime', 'Državljanstvo', 'Pozicija', 'Klub', 'Cena', 'Leto']
prihod = pd.DataFrame(columns=stolpci)
odhod = pd.DataFrame(columns=stolpci)

#povprečna cena pristopa
stevilo_prihodov = []
stevilo_odhodov = []

for i in range (2000,2020):
    transferIN = pd.read_csv(str(path+str(i)+'IN.csv'))
    prihod=prihod.append(transferIN)

    transferOUT = pd.read_csv(str(path+str(i)+'OUT.csv'))
    odhod=odhod.append(transferOUT)

prihod = prihod.reset_index(drop=True)
odhod = odhod.reset_index(drop=True)
```

Slika 5: Izluščevanje državljanstva in pridobivanje vrednosti

2.1 Vpliv prestopa Neymarja na povprečno ceno

Lotimo se prvega zastavljenega vprašanja torej ali in kako je zrasla povprečna cena prestopov po prestopu Neymarja. Po prestopu Neymarja je nogometni klub porabil več denarja kot ga je dobil od prestopa, prav tako pa lahko vidimo naraščajoč trend povprečnih cen prestopov, torej hipoteza drži. To lahko vidimo na sliki **??**, kjer črna navpična črta predstavlja prestop Neymarja.



Slika 6: Graf povprečnih cen prestopov

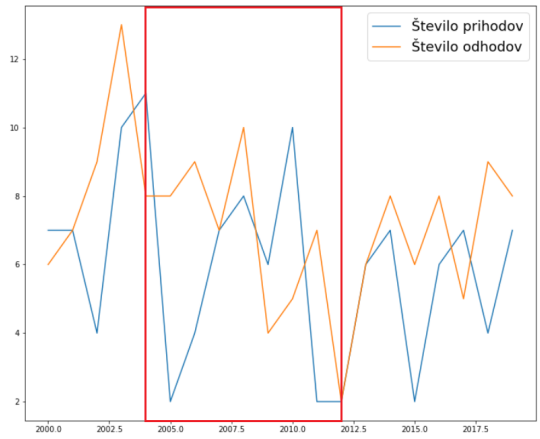
Za pridobitev tega podatka moramo najprej narediti poizvedbo po tabeli vseh prihodov imenovano **prihod**, jo združiti po letih *groupby*('*Leto*') in izračunati povprečje *mean*(). Enako izvedemo na tabeli **odhod**. Za izris grafa pa uvozimo knjižnico *matplotlib.pyplot* in z ukazom *plot*() izrišemo graf. Primer je prikazan na sliki **??**. Za nadaljne primere bomo le obrazložili kako smo prišli do željenih podatkov in ne kako poteka izris grafa.

```
#poglejmo povprečno ceno prihodov in odhodov

avg_cena_prihod = prihod.groupby('Leto').mean()
avg_cena_odhod = odhod.groupby('Leto').mean()

import matplotlib.pyplot as pypt

pypt.figure(figsize=(12,6))
pypt.plot(range(2000,2020),avg_cena_prihod,label="Povprečna cena prihoda")
pypt.plot(range(2000,2020),avg_cena_odhod,label="Povprečna cena odhoda")
pypt.axvline(x=2017, ymin=0, ymax=1,color='Black')
pypt.legend(loc=2, prop={'size': 18})

pypt.show()
```

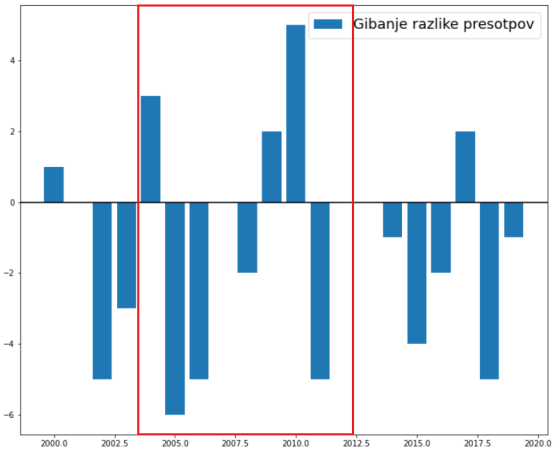
Slika 7: Izsek programske kode za izračun povprečnih vrednosti in izris grafa

2.2 Gibanje števila nakupov in prodaj

Naslednje vprašanje, ki smo si ga zastavili je gibanje števila nakupov in prodaj s hipotezo kako je bilo z nakupi v času 'zlate generacije' La Masie jih je bilo manj. Za obdobje zlate generacije določimo obdobje od leta 2004 do leta 2012. Na sliki **??** lahko vidimo, da je graf prihodov v tem obdobju z izjemo leta 2010 pod grafom odhodov. Ta razlika pa je še bolj očitna na sliki **??**. Na obeh slikah je obdobje v rdečem pravokotniku.



Slika 8: Graf gibanja števila prihodov in odhodov

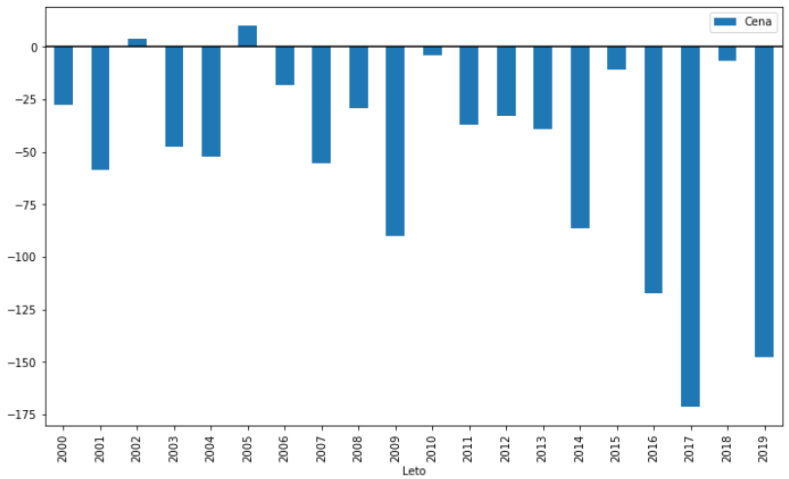


Slika 9: Histogram gibanja razlike med prihodi in odhodi

Opazimo pa tudi, da ni bilo le obdobje zlate dobe La Masie tisto, ko je Barcelona več prodajala kot kupovala, ampak večinoma velja kar za celotno obdobje zadnjih 20 let. Torej hipoteza drži ampak, velja še več, saj je nakupov v splošnem manj kot prodaj. Do teh podatkov smo prišli s pomočjo ukaza $prihod.groupby('Leto').size()$, ki nam na tabeli **prihod** združi prestope po letih, $size()$ pa vrne število vrstic, ki pripadajo posameznim letom. Razlika tega ukaza in analognega za odhod nam vrne gibanje števila razlike prihodov in odhodov, ki jo nato narišemo na graf.

2.3 Cene prestopov

Sedaj se nam pojavi tretje vprašanje. Torej kaj se dogaja s cenami prestopov in ali Barcelona v zadnjih dvajsetih letih služi s transferji. Za začetek si poglejmo razliko med vsoto odhodov po letih in vsoto prihodov po letih. Vsota vseh prihodov po letih lahko dobimo z ukazom $prihod.groupby('Leto').sum()$. Na sliki **??** vidimo kakšna je ta razlika po letih, bolj natančne podatke pa na sliki **??**.



Slika 10: Histogram gibanja razlike potrošnje za prestope

	Cena		
2000	-27.60	**2011**	-37.25
2001	-58.40	**2012**	-33.00
2002	3.65	**2013**	-39.10
2003	-47.35	**2014**	-86.45
2004	-52.45	**2015**	-11.00
2005	10.00	**2016**	-117.15
2006	-18.00	**2017**	-171.30
2007	-55.30	**2018**	-6.90
2008	-29.25	**2019**	-147.60
2009	-90.00	**Skupno**	-1018.45
2010	-4.00		

Slika 11: Histogram gibanja razlike potrošnje za prestope

Izkaže se, da barcelona ne služi s transferji ampak ustvarja kar velik minus. Zanimivo je opaziti vpliv prestopa Neymarja, ko je Barcelona v istem letu porabila skoraj še enkrat več kot je zanj dobila, da bi pokrila luknjo, ki jo je ustvaril njegov odhod. Druga največja razlika se pojavi leta 2019. Verjetno najboljši igralec nogometa Lionel Messi se bliža koncu svoje kariere, zato je klub želel z njim osvojiti še čim več lovorik. Iz tega razloga in slabših rezlutatov po letu 2017 klub veliko več kupuje kot prodaja, a žal rezultati ne kažejo na izboljšanje.

2.4 So Brazilci v povprečju najdražji?

Sedaj nas zanima povprečna cena prestopa po državah, da vidimo katero državljanstvo imajo igralci, ki so najvec vpleteni v prestope. Ker gledamo povprečne vrednosti je treba paziti tudi koliko prestopov je bilo v opazovanem obdobju za posamezno državljanstvo. Zato si poglejmo vrednosti ko imamo vsaj 5 prestopov za državljanstvo in nato še za vsaj 10. Če bi izbrali brez omejitev je najdražji prestop Rusi (Malcom 2019) z vrednostjo v povprečju 40 milijonov € ampak je ta prestop le en. Na slikah **??** in **??** je tabelarični prikaz povprečnih vrednosti prestopov po državljanstvih z vsaj 5 prestopi in vsaj 10 prestopi. Na sliki **??** pa imamo vsote vseh prestopov, kjer vrednost preseže 50 milijonov €.
Opazimo, da v primeru, ko imamo vsaj 5 prestopov so v povprečju nadražji prestopi Portugalci s povprečno vrednostjo 26.28 milijona €. Za vrednosti nad 10 pa naša hipoteza drži saj so na najdražjem mestu Brazilci z vrednostjo 18.45 milijona € in tudi na najdražjem mestu vsote vrednosti prestopov z vsoto 719.40 milijona €. Dokaj prepričljivo bi lahko rekli, da hipoteza drži.

Državljanstvo	Cena
Mexico	3.550000
Argentina	5.947059
Italy	6.200000
Germany	7.080000
Spain	7.139583
Netherlands	8.131579
Sweden	13.000000
France	15.695652
Brazil	18.446154
Portugal	26.281250

Slika 12: Povprečna vrednost prestopa (vsaj 5 prestopov)

Državljanstvo	Cena
Argentina	5.947059
Spain	7.139583
Netherlands	8.131579
France	15.695652
Brazil	18.446154

Slika 13: Povprečna vrednost prestopa (vsaj 10 prestopov)

Državljanstvo	Cena
Sweden	78.00
England	88.45
Chile	100.50
Uruguay	100.75
Argentina	101.10
Netherlands	154.50
Portugal	210.25
France	361.00
Spain	685.40
Brazil	719.40

Slika 14: Skupna vrednost prestopov

Željene podatke dobimo, tako da najprej združimo tabeli **prihod** in **odhod** v tabelo **vsi** z ukazom $prihod.append(odhod)$, ki doda tabeli **prihod** tabelo **odhod**. Nato razporedimo vse prestope prek ukaza $groupby.('Drzavljanstvo')$ jih seštejemo prek ukaza $sum()$, shranimo pod x in izvedemo ukaz $[x > 50].sort\_values('Cena').dropna()$. Ta ukaz nam vrne vse vsote nad 50 milijonov €, jih razporedi po ceni (padajoče) in zbriše vrednosti NaN. NaN bodo tiste vrednosti, kjer je vsota manjša od 50 milijonov €.

2.5 Najbolje trgovane pozicije

Za zaključek si odgovorimo še na zadnje vprašanje katere pozicije na igrišču so najbolj trgovane. Na slikah **??**, **??** in **??** lahko po vrsti vidimo tabelo prihodov, odhodov in skupno. Kar je tu pomembno opozoriti je, da pri skupni tabeli opazujemo po igralcih tako, da je prestop štet le enkrat tudi, če je igralec prestopil večkrat. To dosežemo z ukazom $vsi.drop\_duplicates(subset =' Ime')$.

	Število vseh prestopov
GK	20
FW	44
MF	60
DF	66
Skupno	190

Slika 15: Število vseh prestopov po pozicijah

	Število prihodov
GK	14
FW	32
MF	35
DF	38
Skupno	119

Slika 16: Število prihodov po pozicijah

	Število odhodov
GK	14
MF	41
FW	43
DF	47
Skupno	145

Slika 17: Število odhodov po pozicijah