

Fantasy Premier League Predictor (korišćenjem neuronskih mreža)

David Nestorović

Sažetak

Sama ideja da bi se mogao izgraditi neki računarski sistem, baziran na ljudskom mozgu, javila se još četrdesetih godina prošlog veka. Nedugo zatim, početkom pedesetih, Marvin Minski konstruisao je neuroračunar nazvan Snark, koji se smatra prekretnicom u razvoju neuronskih mreža. Ubrzo zatim Frenk Rozenblat osmislio je koncept jednoslojne neuronske mreže. U narednim godinama bilo je sporadičnih doprinosa razvoju neuronskih mreža, da bi se prva osetnija razlika napravila, uvođenjem neuronskih mreža kao predmeta na nekoliko prestižnih univerziteta u SAD-u. Danas neuronske mreže nalaze svoju primenu u različitim oblastima života počev od prepoznavanja oblika, raspoznavanja govora, autonomne vožnje, pa sve do vojne i medicinske primene. Cilj ovog rada jeste da se proširi domen upotrebe neuronskih mreža, pravljenjem modela koji bi predviđao ishode u popularnoj video igri *Fantasy Premier League*.

1. Uvod

Fantasy Premier League predstavlja popularnu igru u kojoj je cilj sastaviti fudbalski tim(11 igrača i 4 izmene), od igrača koji igraju u Engleskoj Premier ligi. Na samom početku, korisnik(menadžer) ima na raspolaganju 100 miliona(zamišljene valute) da sastavi tim. Cena tima tokom vremena ne mora ostati ista jer cene igrača mogu rasti/opadati shodno postignutim rezultatima. Obično se cene igrača kreću u opsegu od 4 do 12.5 miliona, shodno procenjenoj vrednosti od strane autora igre. Pri sastavljanju tima moraju se ispoštovati sledeća ograničenja:

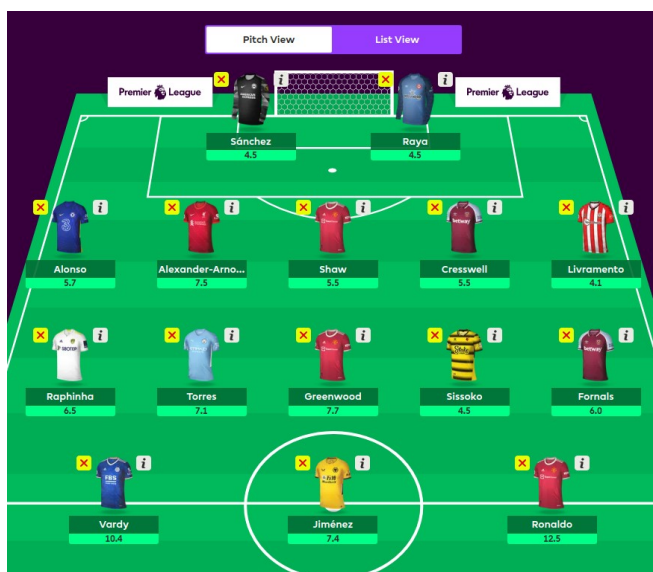
- Svih 15(11 u startnoj postavi i 4 rezerve) igrača moraju biti odabrani
- Tim mora sadržati tačno 2 golmana
- Tim mora sadržati tačno 5 odbrambenih igrača
- Tim mora sadržati tačno 5 veznih igrača
- Tim mora sadržati tačno 3 napadača
- U startnoj postavi može biti samo jedan golman
- U startnoj postavi mora biti minimum 3 odbrambena igrača
- U startnoj postavi mora biti minimum 3 vezna igrača
- U startnoj postavi mora biti minimum 1 napadač
- U timu se ne smeju naći više od 3 igrača iz istog fudbalskog kluba
- Tim ima tačno jednog kapitena i jednog zamenika kapitena

Igra se sastoji u sastavljanju tima koji će ispunjavati pomenute kriterijume a koji će ostvariti što je moguće više bodova. Bodovi se dobijaju prema stvarnim statistikama igrača iz svakog kola Engleske premier lige, i to na način prikazan u narednoj tabeli. Pritom doprinosi ukupnom broju bodova daju samo igrači koje je korisnik odabrao u startnu postavu. Uloga igrača na klupi sastoji se

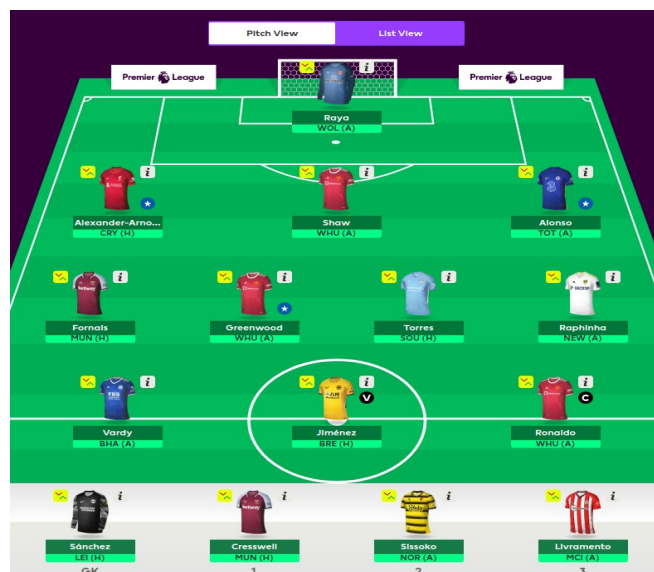
u tome da zamene igrača iz startne postave, ukoliko igrač nije igrao na utakmici svoga kluba. Redosled izmena određuje korisnik tako što prioritet izmena opada sa leva na desno. Pritom, gore definisana ograničenja moraju biti ispunjena. Uloga kapitena je da se njegovi poeni dupliraju, dok je uloga zamenika kapitena da se njegovi poeni dupliraju, ukoliko kapiten nije igrao svoju utakmicu.

Akcija	Bodovi
Za odigranih do 60 minuta	1
Za odigranih preko 60 minuta	2
Za postignut pogodak golmana ili odbrambenog igrača	6
Za postignut pogodak veznog igrača	5
Za postignut pogodak napadača	4
Za svaku asistenciju	3
Za neprimljen gol golmana ili odbrambenog igrača	4
Za neprimljen gol veznog igrača	1
Za svake tri odbrane gomana	1
Za svaki odbranjen penal golmana	5
Za svaki promašen penal igrača	-2
Za svaka dva primljena gola golmana ili odbrambenog igrača	-1
Za svaki žuti karton	-1
Za svaki crveni karton	-3
Za svaki autogol	-2
Bonus poeni za najbolje igrače na utakmici	1-3

Između svaka dva kola moguće je izvršiti najviše jednu *besplatnu* izmenu. Pritom, pravljenje izmene mora poštovati ograničenja sastavljanja tima. Ukoliko korisnik želi da napravi više od jedne izmene, to može učiniti, ali mu se za svaku dodatnu izmenu oduzimaju 4 boda. Ukoliko korisnik ne napravi izmenu, *besplatna izmena* se prenosi u naredno kolo(pri čemu korisnik ne može akumulirati više od dve besplatne izmene).



Odabir tima



Odabir startne postave

1.1 Cilj rada

Ukoliko bismo bili u mogućnosti da predvidimo koliko će igrači ostvariti poena u narednom kolu, mogli bismo da rešimo neki od sledećih optimizacionih problema:

- Najbolji tim za konkretnu nedelju
- Najbolji tim za celu sezonu (tim koji će zahtevati najmanje promena iz kola u kolo)
- Najbolji igrači po najnižoj ceni
- Par golmana koje možemo menjati tako da u svakom kolu imamo solidan broj poena
- ...

Obzirom na širok opseg problema, ovaj rad se fokusira na samom korenu problema, tj. predviđanju broja poena koje će igrači ostvariti u narednom kolu. Rešavanje nekog od navedenih problema ostavljeno je za dalji razvoj ovog projekta

2. Realizacija

Realizacija samog rada, može se podeliti u nekoliko celina, I to:

1. Prikupljanje podataka
2. Analiza prikupljenih podataka
3. Preprocesiranje podataka
 1. Preprocesiranje za individualne statistike
 2. Preprocesiranje za klupske statistike
 3. Spajanje podataka
 4. Čišćenje podataka
4. Linearna regresija
 1. Podela podataka na trening I test skup
 2. Normalizacija podataka
 3. Konstruisanje I obučavanje modela

Kao što je već navedeno, za izradu modela koristila se linearna regresija. Pokazalo se da prost linearni model može da daje solidne rezultate, mada je vrlo verovatno da bi se za postizanje boljih rezultata mogao isprobati neki drugi pristup kao što je rekurentna neuronska mreža.

2.1 Prikupljanje podataka

Podaci korišćeni u ovom projektu, preuzeti su sa gitHub-a. Za ažuriranje podataka postoje python skripte koje vrše skrejpovanje zvaničnog Fantasy Premier League sajta. Takođe, gitHub repozitorijum (čiji link se nalazi u korišćenoj literaturi) na kojem se nalaze podaci, ažurira se svake nedelje. Zbog jednostavnosti, za prikupljanje podataka korišćen je drugi pristup.

2.2 Analiza podataka

Kako bismo imali uvid u dimenziju skupa podataka sa kojim radimo, I njegovim karakteristikama, najpre je izvršena analiza podataka. Sam skup podataka sastoji se iz podataka prikupljenih iz prethodnih pet sezona Engleske Premier Lige. Takođe, prikupljene podatke možemo odvojiti u dve celine:

- Podatke vezane za individualne statistike igrača
- Podatke vezane za statistike klubova

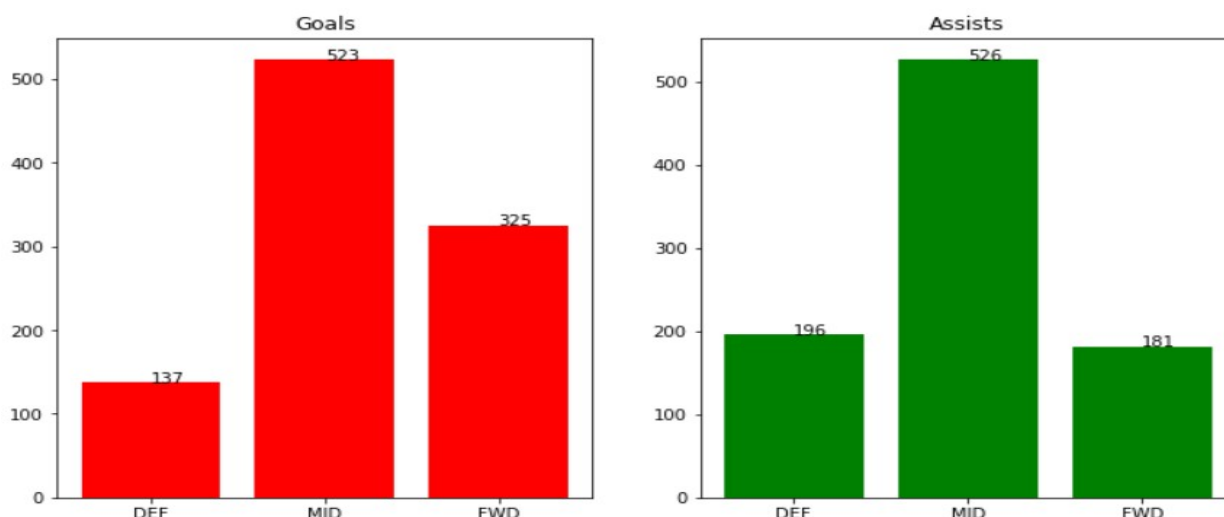
Najpre, podaci vezani za individualne karakteristike igrača sastoje se iz 36 kolona u kojima se nalaze različite statistike igrača. **Prva pretpostavka** koja je doneta na osnovu statičke analize podataka je da postoje statistike koje nam verovatno nisu od pomoći(kasnije se ispostavlja da se uvrštavanjem ovih statistika kompleksnost modela povećala, a samim tim se I kvalitet modela pogoršao, te je odlučeno da se te statistike izbace iz skupa podataka). Statistike koje nam po pretpostavci nisu od koristi su: *vreme odigravanja utakmice*, *runda*(redundantan podatak obzirom da postoji kolona koja nam govori koja je nedelja u toku), i *koji klubovi su akteri meča*(opet redundantna karakteristika, obzirom da postoje kolone protivnik, I znamo za koga posmatrani igrač nastupa).

Posmatranjem skupa možemo zaključiti da se u skupu nalazi dosta korisnih podataka, samo u pogrešnom obliku. Naime, za svakog igrača I svako kolo jedne sezone date su neke statistike. Među najbitnijim statistikama izdvajaju se: postignuti *golovi*, *primljeni golovi*, *asistencije*, *pozicija igrača*, *novčana vrednost igrača*, *koliko korisnika je odabralo nekog igrača*... Razlog za **drugu pretpostavku**(da su podaci u pogrešnom) obliku je sledeći: *ako se broj osvojenih poena igrača(što je ciljani atribut) računa na osnovu postignutih golova, asistencija, žutih kartona*... takvi podaci ne smeju ostati u skupu, jer bi naš model mogao da dodeli težine parametrima tako da tačno računa osvojene poene na osnovu tih podataka(tj model bi naučio sistem bodovanja igrice). Rešenje koje se nameće je da se za svaku nedelju I svakog igrača računaju forme. Forme mogu biti kratkoročne(recimo 5 utakmica) I dugoročne(recimo 10 utakmica). Na osnovu ove pretpostavke, neophodno je zameniti sve kolone u kojima se nalaze *kritične statistike*¹ sa njihovim odgovarajućim formama. Tako se recimo sve kritične statistike u dvadesetoj nedelji zamenjuju sa formama sa prethodnih 5(tj. 10) kola. Sama pretpostavka nameće se I prirodnim razmišljanjem: ukoliko mi sastavljamo tim, ne znamo podatke za kolo koje se trenutno igra(kritične statistike), ali znamo koji igrač se nalazi u kakvoj formi I šta je uradio na prethodnim utakmicama.

Takođe, može se uočiti da pojedine kolone ne mogu imati forme. Besmisleno je recimo praviti formu za kolone poput: tekuće nedelje, ime igrača, pozicija, tim, da li se utakmica igra kod kuće ili u gostima... Činjenica da te kolone nemaju svoje forme, ne čini ih manje bitnim, što se može primetiti nekim narednim analizama.

Recimo da želimo da vidimo produktivnost igrača, tj. broj golova I asistencija na nivou cele sezone. Na slici 1 prikazani su grafici zavisnosti broja golova I asistencija od pozicije na kojoj igrač igraju.

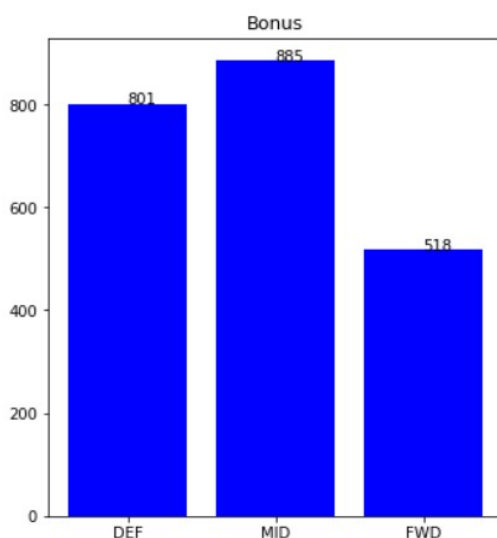
1 Kritične statistike su one statistike na osnovu kojih bismo naveli naš model da nauči samo sistem bodovanja, a ne da predviđa stvarnu važnost nekih statistika.



Slika 1

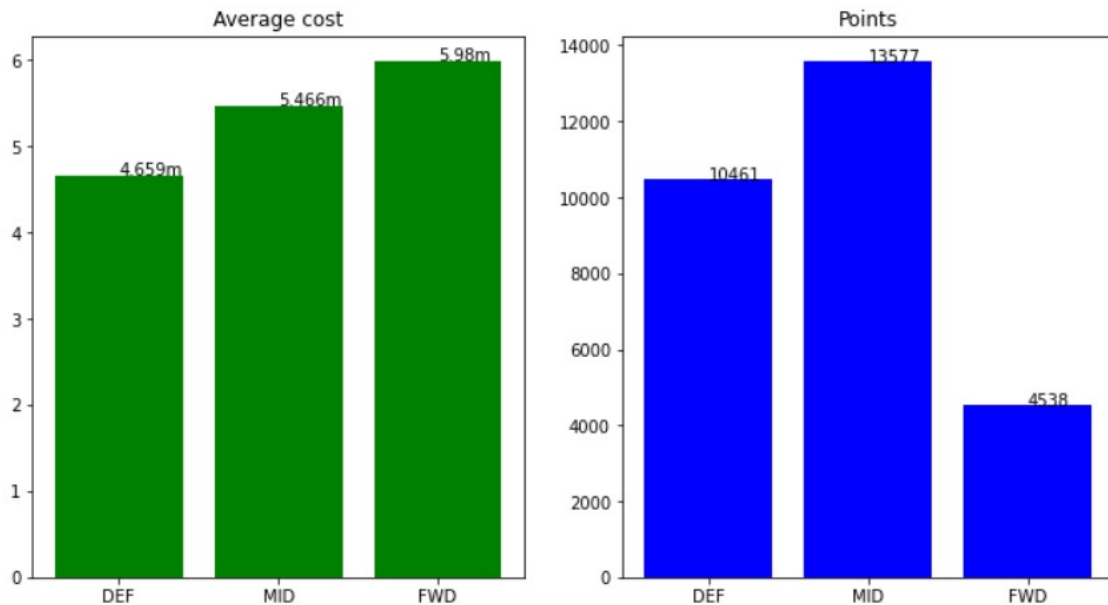
Ono što na prvi pogled nije izgledalo kao očekivano, je da su napadači postigli manje golova od veznih igrača tokom sezone. Takođe, očekivano je da su vezni igrači postigli više asistencija od ostalih, ali nije očekivano da su odbrambeni igrači imali više asistencija od napadača. Međutim čak i ova statistika može da zavarava. Naime, igra je koncipirana tako da je većina igrača svrstana u kategoriju veznih igrača, nešto manje je odbrambenih, a najmanje je napadača. Ipak, čak i ovakva statistika nas navodi na pretpostavku da će vezni igrači verovatno ostvarivati više poena, jer su njihove brojke neuporedivo veće u odnosu na ostale.

Možda nešto neočekivaniji podaci dobijaju se kada se posmatraju bonus poeni po pozicijama. Bonus poene, igrač dobija na osnovu nekih naprednijih statistika kao što su: broj uspešnih driblinga, broj dobijenih duela, broj iznuđenih faulova... Na slici 2 možemo videti raspodelu bonus poena po pozicijama. Ono što je pomalo iznenađujuće je činjenica da su odbrambeni igrači ostvarili više bonus poena u odnosu na napadače. Iz ovoga zaključujemo da je verovatno i najveća prednost odbrambenih igrača u tome što često dobijaju bonus poene, te ovaj segment nadomeštava činjenicu da se odbrambeni igrači ne nalaze toliko često u situacijama da postignu gol ili asistiraju.



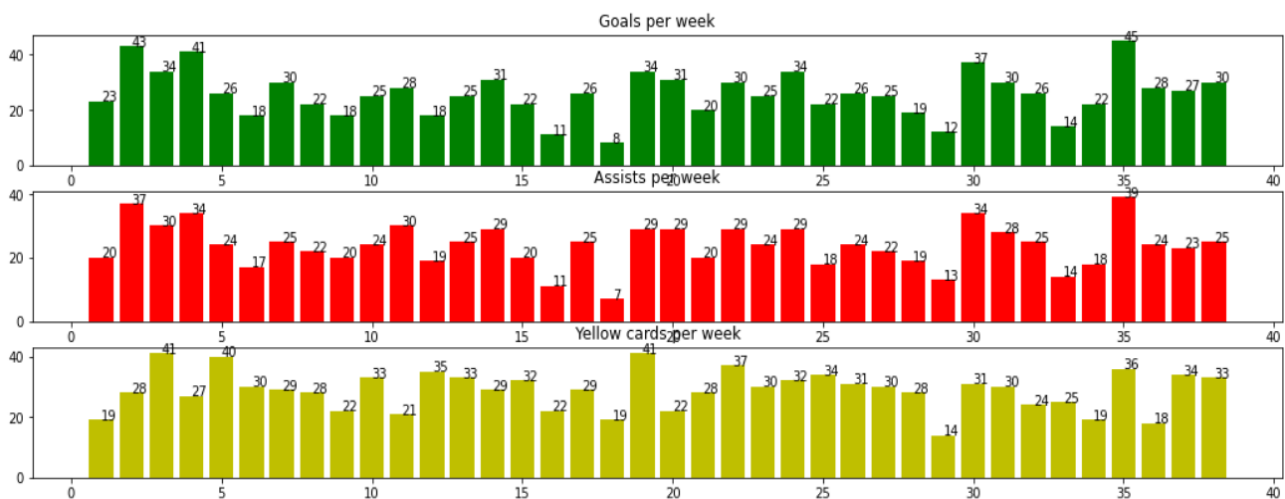
Slika 2

Još neke od bitnih statistika koje vredi pomenuti su prosečna cena igrača I ukupan broj poena koje su igrači ostvarili zavisno od pozicija na kojima igraju. Na slici 3 možemo videti da najvišu prosečnu cenu imaju napadači, dok je broj poena koje su ostvarili neuporedivo manji od odbrambenih I veznih igrača.



Slika 3

Takođe, na našem skupu mogli bismo uočiti kako se statistike ponašaju zavisno od dela sezone u kojem se igra. Na slici 4 možemo videti da je raspodela svih statistika približno ravnomerna, iako se moglo pretpostaviti da će se kako se bliži kraj sezone(I borba za titulu/opstanak) brojke uvećati, određena konstantnost se ipak zadržala.



Slika 4

Zaključak statičke analize skupa podataka je da ne bismo mogli tako lako da uočimo neke pravilnosti, ili bismo čak doneli neke pogrešne pretpostavke(kao što je ona vezana za bonus poene igrača).

2.3 Preprocesiranje podataka

2.3.1 Preprocesiranje za individualne statistike

Kako je već I napomenuto prema prvoj pretpostavci u statičkoj analizi skupa podataka, najpre je neophodno odbaciti statistike koje nam nisu od interesa. Stoga, prvi korak u preprocesiranju datog skupa podataka bilo je uklanjanje suvišnih statistika poput: vremena početka utakmice, kola odigravanja, protivnicima na susretu... Nakon toga, pristupilo se I realizaciji druge pretpostavke po kojoj se u datom skupu podataka nalaze se jako bitne statistike, ali date u(znači) neodgovarajućem obliku. Dakle, umesto podataka o tome šta je igrač uradio u posmatranoj nedelji, pravimo statistike o tome kakve statistike je igrač postigao na prethodnih 5 odnosno 10 utakmica. Za pojedine statistike, nije bilo moguće(ili nije imalo smisla praviti forme) te su takve statistike izostavljene iz ovog dela obrade. Napomenimo samo još I to da su prva I druga faza međusobno nezavisne, te je njihov redosled nebitan po rad algoritma(nije bitno da li ćemo izbaciti neku statistiku pa onda praviti forme, ili prvo napraviti forme a onda svakako izbaciti nebitne statistike. Svakako se kroz te statistike ne prolazi pri kreiranju formi).

2.3.2 Preprocesiranje za klupske statistike

Prvi problem sa kojim se susrećemo već pri prvom pogledu skupa podataka vezanog za timove jeste neusklađenost imena u tom skupu podataka I skupu sa individualnim statistikama igrača. Obzirom da je broj timova sa takvim nesuglasicama mali, pribeglo se prostom mapiranju naziva timova. Kao I za skup podataka sa individualnim statistikama, I u ovom skupu smo odabrali neke bitne statistike a ostale smo odbacili. Nakon toga, za preostale kolone(statistike) računamo forme ukoliko je to moguće.

2.3.3 Spajanje skupova podataka

Sledeći korak nakon što imamo konzistentne podatke, je spajanje dva skupa podataka prema nazivu kluba I nedelji odigravanja. Obzirom da se ove dve kolone pojavljuju u oba skupa podataka, dolazi do njihovog dupliranja, te se jedan par mora odstraniti.

2.3.4 Čišćenje podataka

Najpre, potrebno je proveriti da li se u posmatranom skupu nalaze neke nedostajuće vrednosti ili vrednosti koje su NaN². Primenom jednostavnih funkcija iz biblioteke pandas, primećeno je da na trenutnom skupu nemamo nedostajuće niti NaN vrednosti.

Sledeći korak u čišćenju podataka bilo bi enkodiranje kategoričkih promenljivih(kao što je ime kluba ili ime igrača). Primenom dummy enkodiranja, dobili smo po jednu kolonu za svaki klub I svako ime igrača, gde se u svim kolonama nalazi vrednost 0 osim u koloni koja odgovara predašnjem imenu kluba I igrača za red iz tabele.

Na posletku, neke promenljive koje su bile logičkog tipa, pretvorene su u celobrojni tip(0 ili 1) kako bi se olakšao rad neuronskoj mreži.

2.4 Linearna regresija

2.4.1. Podela podataka na trening I test skup

Po već dobro poznatoj praksi, polazna tačka razvoja linearnog regresionog modela bila bi podela skupa podataka na trening I test skup. Pritom, od ovog momenta pa na dalje ni u kom trenutku podaci izdvojeni u test skup ne smeju uticati na treniranje niti formiranje našeg modela. Obzirom da podataka ima dosta, za test skup odvojeno je 20% ukupnog skupa, dok je ostatak korišćen za treniranje modela. **Obzirom da je cilj rada predvideti broj poena koje će igrač ostvariti, kolona sa ovom statistikom uklanja se I iz trening I iz test skupa.**

2.4.2. Normalizacija podataka

Obzirom na nesrazmernost pojedinih podataka kao što su cena igrača(koja se meri u milionima) I recimo broja golova koja je jednocifrena I nekih naprednih statistika koje uzimaju vrednosti reda stotine, neophodno je izvršiti normalizaciju podataka kako pojedini atributi ne bi bili favorizovani samo zbog njihovog specifičnog opsega vrednosti.

Najpre definišimo par osnovnih matematičkih pojmova

- Srednja vrednost $\bar{x} = \frac{1}{n} \cdot \sum_{k=1}^n x_k$
- Standardna devijacija $dev = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2}$

Nakon što smo definisali značenja srednje vrednosti I standardne devijacije, našu normalizaciju vršimo po formuli

$$\frac{x - \bar{x}}{dev}$$

Osim prednosti, ovaj pristup(vrlo brzo se ispostavilo) ima mane. Naime, ako je standardna devijacija u nekom trenutku postigla vrednost 0, imamo deljenje sa nulom te samim tim dobijamo NaN vrednost. Ova, naizgled ne tako očigledna greška, prouzrokovala je da kasnije cela neuronska mreža daje vrednosti NaN za svoje funkcije preciznosti I greške. Srećom, ovakva situacija ne događa se toliko često, te se brзом proverom da li u skupu postoje NaN vrednosti I njihovim otklanjanjem, model ponovo dove u funkcionalno stanje.

2.4.3 Konstruisanje I obučavanje modela

Korišćenjem biblioteke keras, za već pripremljene, I normalizovane podatke formiran je sekvencijalni model neuronske mreže sa 8 slojeva I 6 slojeva za uprošćavanje(dropout). Funkcia aktivacije korišćena u ovom modelu je *relu*. Pored ovog modela, testirano je još dosta modela, ali se pokazalo da dodavanjem novih slojeva ne dobijamo na tačnosti na test skupu, tj. dolazi do prilagođavanja.

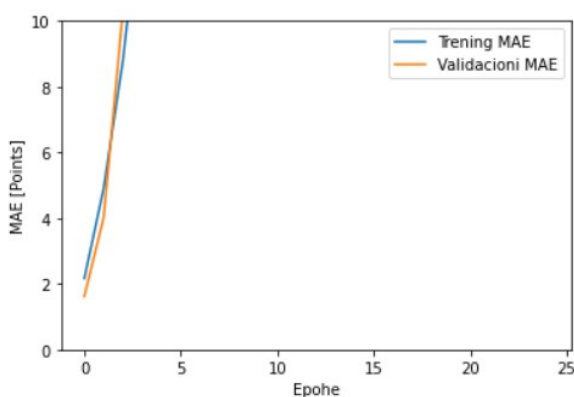
Za funkciju gubitka korišćena je srednja kvadratna logaritamska greška. Modeli su testirani I sa drugim funkcijama greške ali pojedine funkcije gubitka nisu uspevale da daju validne rezultate tj. na izlaz su propagirale NaN vrednosti. Na svim modelima koji su ispitivani korišćen je optimizator *Adam*, kao optimizator koji se u literaturi označava kao najbolji u najviše praktičnih slučajeva.

3. Analiza dobijenih rezultata

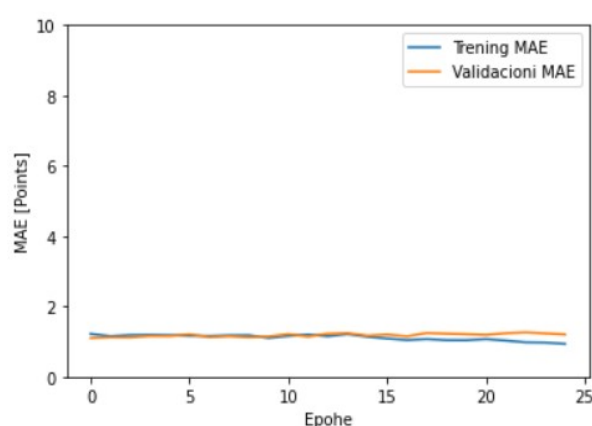
Obzirom na prirodu problema, ne postoje objavljeni relevantni radovi sa kojima bi se dobijeni model mogao porediti. Razlog za to je verovatno ekonomska isplativost obzirom da igru igra preko 6 miliona igrača, I da mnogi zainteresovani da po svaku cenu u svoj tim dovedu što kvalitetnije igrače, za koje se predviđa da će imati dobro naredno kolo.

Shodno tome, analiza podataka svešće se na procenu analize efikasnosti.

Ako zanemarimo modele koji su davali očajne rezultate(a bilo je modela koji su u porceni grešili I za po više od 100 poena), centralno pitanje kvaliteta modela vrti se oko par modifikacija sličnog modela. Najpre, uočimo razliku kako isti modeli(sa stanovištva slojeva) daju drastično velike razlike u kvalitetu ukoliko variramo parametar³ koji prima *Adam* optimizator. Na slikama 5 I 6 nalaze se grafici promene funkcije srednje apsolutne greške(apsolutno odstupanje u broju poena). Na slici 5 kao parametar za optimizator *Adam* korišćena je vrednost 0.01, dok je na slici 6, parametar imao vrednost 0.001.



Slika 5

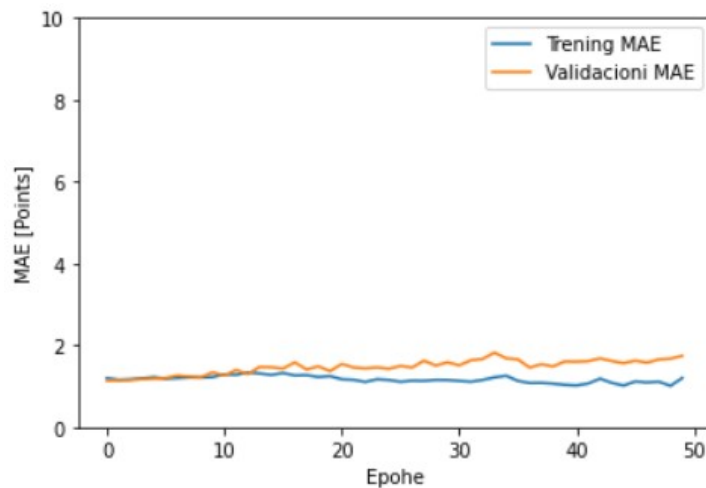


Slika 6

Iz posmatranog možemo zaključiti da je parametar od 0.01 preveliki I da odmah naš model “odluta” od lokalnog ekstremuma što za posledicu ima lošiji kvalitet modela. Sa stanovištva broja poena u kojima odstupa, model sa parametrom 0.01 odstupa u predviđanju za preko 3000 poena, te se ovaj rezultat smatra katastrofalnim. Za isti broj epoha, I isti model, samo sa parametrom 0.001, odstupanje u proceni broja poena je oko 1.63 poena, što je enormno bolji rezultat od prethodno dobijenog.

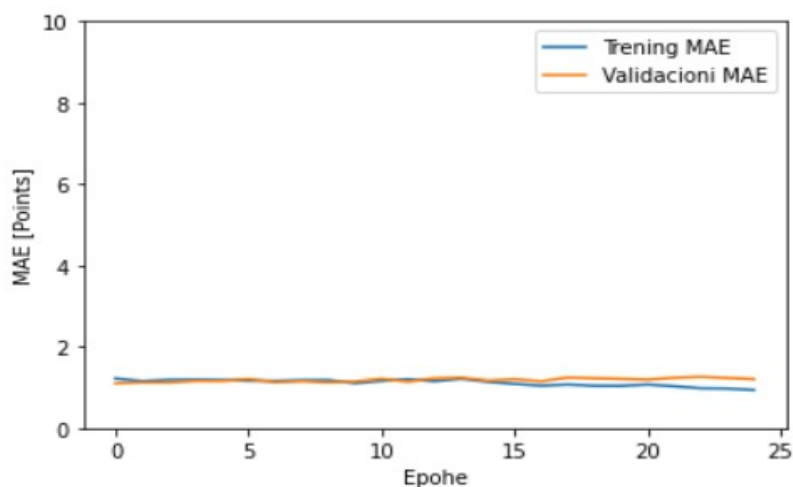
Takođe, jedan od problema sa kojima se razvoj modela susreo bilo je I prilagođavanje. Naime, ukoliko bismo povećavali broj slojeva neuronske mreže, naš model bi na trening podacima pokazivao značajno poboljšanje u predviđanjima(odstupao bi za oko 0.5 poena što je, sa stanovišta upotrebljivosti na realnoj igri, fenomenalan rezultat), međutim, na test podacima odstupanje u predviđanju bi se samo pogoršavalo kako napredujemo kroz epohe. Na slici 7 prikazan je primer kada je jedan od modela prilagođen.

3 Ovaj parametar nam govori koliko usitnjavamo pretragu. Povećanjem ovog parametra pravićemo veće oscilacije, u pretrazi, što za posledicu ima da se nećemo zaglavljivati u lokalnim ekstremumima. Nasuprot tome, loša strana je što ćemo nekad možda, kada smo na dobrom putu, odskočiti previše od rešenja I nećemo naći odgovarajući optimum uopšte. Stoga treba pažljivo odabrati ovaj parametar tako da ne bude ni preveliki(kako ne bismo pretragu navodili gotovo nasumično zbog čestih oscilacija) ni premali(kako se ne bismo zaglavili u lokalnim ekstremumima).



Slika 7

Predstavimo sada dva modela koja su se pokazala sasvim solidno. Najpre, prikazimo nešto prostiji model sa 4 sloja u mreži. Iako čak i on pokazuje znake preprilagođavanja, i dalje ostaje najbolji model koji se mogao dobiti. Na slici 8 prikazan je grafik funkcije greške ovog modela. Sa stanovišta odstupanja, ovaj model na test skupu odstupa oko 1.63 poena, što je sasvim solidan rezultat.

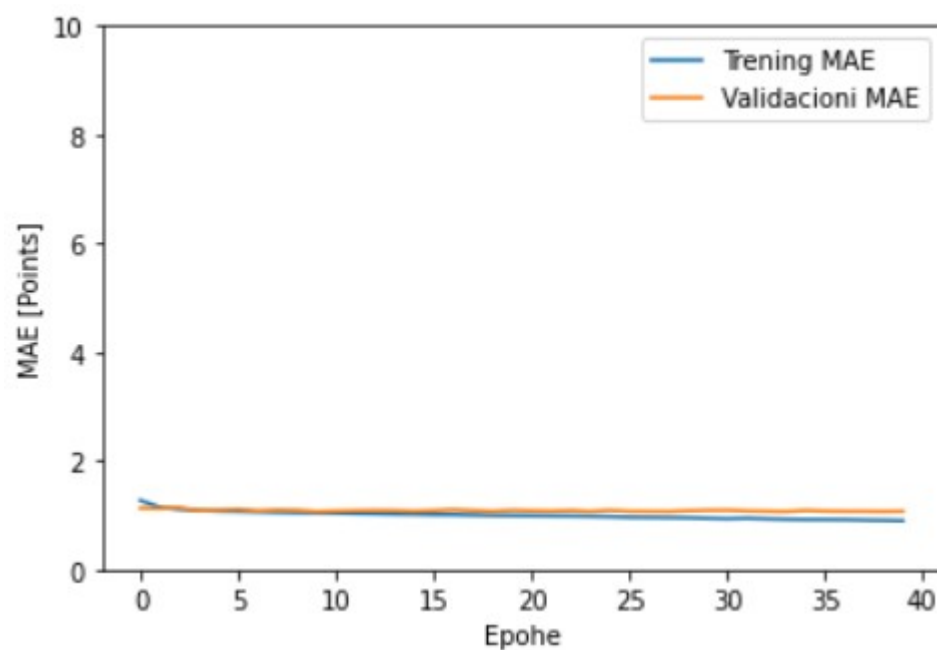


Slika 8

Konačno, najbolji model do kojeg se došlo nad ovim podacima bio je gore pomenuti model sa 8 slojeva. On je, primenjen na 40 epoha, davao rezultate po kojima se na test skupu odstupa za 1 poen u predviđanju. Naglasak je na 40 epoha jer se pri povećanju epoha na 60 dobilo preprilagođavanje, te se stoga naš finalni model obučavao na 40 epoha. Na slici 9 prikazana je struktura modela, dok je na slici 10 prikazana funkcija srednje apsolutne greške.

Layer (type)	Output Shape	Param #
dense_128 (Dense)	(None, 128)	102016
dropout_63 (Dropout)	(None, 128)	0
dense_129 (Dense)	(None, 64)	8256
dropout_64 (Dropout)	(None, 64)	0
dense_130 (Dense)	(None, 32)	2080
dropout_65 (Dropout)	(None, 32)	0
dense_131 (Dense)	(None, 32)	1056
dropout_66 (Dropout)	(None, 32)	0
dense_132 (Dense)	(None, 32)	1056
dropout_67 (Dropout)	(None, 32)	0
dense_133 (Dense)	(None, 32)	1056
dropout_68 (Dropout)	(None, 32)	0
dense_134 (Dense)	(None, 16)	528
dense_135 (Dense)	(None, 8)	136
dense_136 (Dense)	(None, 1)	9
Total params: 116,193		
Trainable params: 116,193		
Non-trainable params: 0		

Slika 9



Slika 10

4. Zaključak

Iz svega viđenog u prethodnom poglavlju, zaključak je da naš model, koliko god jednostavan bio, uz dobro pripremljen skup podataka, daje solidne rezultate. Kao jedno od glavnih poboljšanja koje se razmatra jeste proširenje skupa podataka. Za trenutnu verziju, korišćeni su podaci iz samo jedne sezone Engleske Premier Lige, dok bi se kvalitet modela mogao poraviti dodavanjem ostalih raspoloživih sezona. Problem koji se javlja pri spajanju ovih skupova jeste taj što klubovi konstantno ispadaju a novi klubovi ulaze u ligu, tako da bi se najpre morao taj problem rešiti (verovatno izbacivanjem redova koji sadrže igrače iz klubova koji ne igraju u tekućoj sezoni u Premier ligi).

Još jedno moguće poboljšanje bilo bi ako bismo isprobali i druge modele neuronskih mreža, kao što su rekurentne neuronske mreže. Povod za ovakvo razmišljanje leži u samoj prirodi problema. Naime u realnim situacijama imamo neke česte i retke ekstremume. U česte ekstremume spada situacija kada igrač dobije crveni karton, i na narednoj utakmici ne može da igra. Retki ekstremumi su situacije kada se igrač recimo povredi i shodno tome propusti nekoliko utakmica. Tu nastupa rekurentna neuronska mreža, gde bismo mi imali uvidu u to kako se igrač ponašao prethodni put kada je bio suspendovan ili povređen kako bismo takve već izračunate situacije upotrebili za računanje neke nove ekstremne situacije.

Pored stvari vezanih za sam kvalitet modela, aplikacija bi se mogla unaprediti i sa strane nekih praktičnijih aspekata. Naime, naša aplikacija samo predviđa broj poena koje će neki igrač ostvariti u narednom kolu, ali se ovaj model može nadograditi tako da korišćenjem ograničenja aplikacija:

- Sastavlja najboljih 11 igrača prema ograničenjima utvrđenim na početku igre tako da broj bodova bude maksimalan moguć.
- Procenjuje kojeg je igrača najbolje staviti za kapitena.
- Određuje par golmana koji će se smenjivati tokom sezone tako da postignu maksimalan broj poena.
- Određuje tim koji će iz kola u kolo zahtevati minimalan broj izmena a postići najveći broj poena tokom sezone (celokupna poenta igre).
- Procenjuje promene cena igrača tako da sugeriše kada treba izvršiti koju izmenu kako ne bismo gubili novac.

5. Literatura

- Github repozitorijum za dataset, <https://github.com/vaastav/Fantasy-Premier-League>
- *Computational Intelligence - An Introduction*, Andries Engelbrecht, John Willey & Sons, 2007.
- [Vestacka inteligencija](#), Predrag Janicic, Mladen Nikolic
- Pandas https://pandas.pydata.org/docs/user_guide/index.html#user-guide