Usporedba modela za prognoziranje ishoda nogometnih utakmica

Škalić, Marko

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Marko Škalić

USPOREDBA MODELA ZA PROGNOZIRANJE ISHODA NOGOMETNIH UTAKMICA

ZAVRŠNI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Škalić

Matični broj: 0016149028

Studij: Informacijski i poslovni sustavi

USPOREDBA MODELA ZA PROGNOZIRANJE ISHODA NOGOMETNIH UTAKMICA

ZAVRŠNI RAD

Mentorica:

Jelena Gusić Munđar, mag. math.

Varaždin, rujan 2024.

Marko Škalić

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U radu se promatra mogućnost predviđanja rezultata nogometnih utakmica uz pomoć matematičkih modela. Modeli koji se koriste u radu su Poissonov model i binomni model. Odabrana liga za testiranje spomenutih modela bila je talijanska prva liga, Serie A, specifično su analizirani rezultati sezone 2020/2021. godine. Podaci koji su se koristili u istraživanju bili su podaci iz prvog dijela sezone, te su isti služili za procjenu parametara. Neki od korištenih podataka bili su, ukupan broj postignutih i primljenih pogodaka i broj odigranih utakmica, te isti ti podaci, ali podijeljeni na domaće i gostujuće utakmice. U praktičnom dijelu rada, koristeći programski jezik Python, implementirani su modeli, te testirani u slučajevima kao što je predviđanje konačnog rezultata pojedine utakmice dviju ekipa, predviđanje konačnog ishoda utakmice između dvije ekipe i predviđanje konačnog poretka na kraju sezone.

Ključne riječi: Poissonov model, binomni model, nogomet, obrada podataka, sportska analitika, prediktivni modeli

Sadržaj

1. Uvod ... 1
2. Metode i tehnike rada ... 2
3. Predviđanje rezultata u nogometu ... 3
   3.1. Podaci potrebni za predviđanje ... 4
   3.2. Metode za predviđanje ishoda ... 4
      3.2.1. Poissonov model ... 5
      3.2.2. Binomni model ... 6
   3.2.3. Parametri modela ... 7
   3.2.4. Ostale primjene modela ... 8
4. Implementacija modela za predviđanje rezultata u programskom jeziku Python ... 9
   4.1. Promatrana nogometna liga – Serie A ... 9
   4.2. Predviđanje konačnog rezultata utakmice ... 10
      4.3. Predviđanje rezultata utakmice koristeći Poissonov model ... 14
      4.4. Predviđanje rezultata utakmice koristeći Binomni model ... 17
   4.5. Predviđanje generalnog ishoda utakmice ... 21
      4.5.1. Generalni ishod Poissonovim modelom ... 21
      4.5.2. Generalni ishod binomnim modelom ... 22
   4.6. Predviđanje poretka na kraju sezone ... 24
5. Zaključak ... 32
Popis literature ... 33
Popis slika ... 35
Popis tablica ... 36

1. Uvod

Nogomet je najpopularniji sport na svijetu, ali je vrlo nepredvidljiv. Svojom popularnošću privlači pozornost raznih skupina, poput običnih navijača, medijskih skupina, ali i kladionica. Nepredvidljivost ishoda, u kojemu čak i porazom ekipa može trijumfirati ako se poklope ostali čimbenici, podiže uzbuđenje i strast oko sporta. No međutim, upravo ova nepredvidivost predstavlja velik izazov za kladioničare, analitičare ali i same momčadi.

Potraga za točnim predviđanjem ishoda rezultata nogometnih utakmica privukla je širok interes ne samo u navijačkom smislu. Razne discipline poput matematike, informatike, statistike i ostalih, mogu koristiti nogomet i rezultate kao podatke koji su validni za mnoga znanstvena istraživanja.

Sposobnost točnog predviđanja ishoda nogometne utakmice utječe na odluke koje donosi Uprava nogometnog kluba, na zaradu kladionica, na sportsku analitiku, ali i na angažman navijača.

Prvi dio rada je teoretski dio, te nas upoznaje s povijesti predviđanja rezultata sportskih događaja. Spominju se razne moguće vrste i metode predviđanja, gdje se sve predviđanja mogu koristiti, koji čimbenici utječu na predviđanje te koje su moguće prednosti dobre analize i uspješnog predviđanja rezultata. Uz to upoznajemo se s promatranim matematičkim modelima, a to su Poissonov i binomni model. Opisani su parametri koji će se koristiti u praktičnom dijelu rada.

Drugi dio rada odnosi se na praktični dio, točnije implementacija Poissonovog i binomnog modela za predviđanje rezultata, pomoću programskog jezika Python. Dobiveni rezultati se uspoređuju sa stvarnim povijesnim podacima te se na taj način ocjenjuje točnost modela u predviđanju konačnog ishoda.

2. Metode i tehnike rada

U teorijskom dijelu rada korišteni su isključivo web izvori iz raznih znanstvenih i stručnih radova, ali i ovlaštenih stranica. To znači da je korištena metoda sekundarnog istraživanja te su zapisane informacije dobivene na temelju istraživanja drugih

U praktičnom dijelu rada primjenjena su znanja i informacije iz teorijskog dijela. Pomoću formula korištenih u teorijskom dijelu moguće je bilo implementirati Poissonov i binomni model. Za implementaciju je korišten programski jezik Python, u online razvojnom okruženju Google Colab. Za vizualizaciju podataka korišteni su Microsoft Excel i Microsoft Power BI.

3. Predviđanje rezultata u nogometu

Nogometno klađenje je tržište vrijedno više milijardi dolara i prema istraživanju potrebno je imati inteligentan sustav za predviđanje rezultata. Kladioničarske prognoze u nedavnoj prošlosti uglavnom su se oslanjale na visokokvalificirane stručnjake iz područja i složene statističke metode. Prethodne studije u ovom području uvelike se razlikuju u razmatranim informacijama o igri i modelima predviđanja. [6]

Informacije prije utakmice mogu uključivati povijesne podatke o učinku momčadi, statistiku pojedinih igrača, stanje ozljeda, vremenske uvjete, ili čak psihološki pritisak momčadi. Podaci tijekom same utakmice, poput statistike posjeda lopte, broja stvorenih prilika, učinka igrača na terenu, ključni su za prilagodbu prognoza u stvarnom vremenu. Višesezonski podaci omogućuju analizu trenda učinka momčadi kroz duži vremenski period, što je korisno za razumijevanje stabilnosti i dugoročne forme momčadi, dok se podaci jedne sezone, primjerice tekuće, fokusiraju na trenutnu formu i specifičnosti tekuće sezone, poput utjecaja novih igrača, ozljeda ključnih igrača tijekom ključnih faza sezone ili varijacija u domaćim i gostujućim utakmicama.

Štoviše, izvan industrije klađenja, nogometni klubovi i menadžeri sve se više oslanjaju na pristupe temeljene na podacima kako bi stekli konkurentsku prednost. Prediktivni modeli mogu pomoći timovima u donošenju informiranih odluka u vezi s taktikom, odabirom igrača i strategijama u igri. Točna predviđanja također omogućuju klubovima da predvide strategije protivnika i prilagode svoje planove igre u skladu s tim, potencijalno preokrećući tok igre u svoju korist u ključnim utakmicama. [15]

Posljednjih godina, napredak u matematici, statistici i računskoj moći otvorio je nove puteve za poboljšanje točnosti nogometnih predviđanja. Istraživači i analitičari počeli su istraživati različite matematičke modele, poput Poissonove i binomne distribucije, kako bi predvidjeli ishode na temelju povijesnih podataka i varijabli specifičnih za utakmice. Ovi modeli nude strukturiraniji pristup predviđanju koji se temelji na podacima, udaljavajući se od oslanjanja na intuiciju i stručno mišljenje. [3]

Postoji nekoliko prednosti predviđanja sportskih događaja. Predviđanja mogu pomoći pri poboljšanju donošenja odluka, pružajući točnu procjenu čimbenika koji mogu utjecati na ishod sportskog događaja. Pružajući podatke i informacije o odabranim momčadima i igračima, predviđanje može pomoći u donošenju boljih odluka. Predviđanja sportskih događaja također može pomoći u prepoznavanju potencijalnih slabosti u protivničkoj momčadi. [11]

3.1. Podaci potrebni za predviđanje

Pojava velikih podataka i tehnologija strojnog učenja dodatno je revolucionirala područje sportske analitike. Uz dostupnost golemih količina podataka — od statistike igrača i metrike učinka momčadi do podataka utakmica u stvarnom vremenu — istraživači sada imaju alate za razvoj složenijih modela koji mogu učiti iz obrazaca i trendova tijekom vremena. [13]

Prilikom predviđanja sportskih događaja potrebno je uzeti u obzir nekoliko čimbenika. Povijesni podaci mogu pružiti korisne informacije o formi i učinku momčadi, dok vremenski uvjeti mogu utjecati na ishod sportskog događaja. Ozljede igrača mogu utjecati na učinak momčadi, a učinak igrača može utjecati na ishod utakmice. Osim toga, sudac također može utjecati na ishod utakmice. Suci su odgovorni za donošenje ključnih odluka tijekom utakmice i mogu utjecati na ishod utakmice. Vanjski utjecaji kao što su lokacija utakmice, broj gledatelja i moral momčadi također mogu utjecati na ishod utakmice. Također potrebno je uzeti u obzir i faktor sreće kada se predviđa ishod utakmice. [11]

3.2. Metode za predviđanje ishoda

Postoji nekoliko metoda predviđanja sportskih događaja. Fundamentalna analiza je metoda koja promatra temeljne elemente koji čine momčad kao što su sposobnosti igrača, njihova forma i ukupna izvedba momčadi. Tehnička analiza je metoda koja koristi grafikone i indikatore za prepoznavanje trendova u izvedbi tima, dok statistička analiza koristi razne tehnike za analizu podataka radi predviđanja. [11]

Druga metoda je korištenje mišljenja i predviđanja analitičara. Ova metoda koristi mišljenja stručnjaka za predviđanje. Također postoji i umjetna inteligencija koja koristi računalne algoritme za analizu podataka i stvaranje predviđanja. Sve ove metode imaju svoje prednosti i nedostatke, a najbolja metoda za predviđanje sportskih događaja varirat će ovisno o situaciji. S pravom kombinacijom metoda svatko može dati informiranija predviđanja i povećati svoje šanse za uspjeh. [11]

Algoritmi strojnog učenja, u kombinaciji s tradicionalnim statističkim metodama, mogu identificirati suptilne korelacije i skrivene obrasce koje ljudski analitičari mogu propustiti. Ove tehnologije imaju potencijal značajno poboljšati točnost predviđanja, osiguravajući dionicima konkurentsku prednost. [13]

Poissonova distribucija se pokazala učinkovitom u modeliranju broja golova postignutih u utakmici, dok se binomni model može primijeniti za predviđanje vjerojatnosti određenih binarnih ishoda, poput pobjede ili poraza. Ovi modeli, kada su ispravno kalibrirani, mogu pružiti uvide koji su i djelotvorni i pouzdani, poboljšavajući proces donošenja odluka za sve uključene. [17]

Prognoziranje Poissonovim modelom i binomnim modelom pripadaju jednostavnijim načinima prognoziranja. Od drugih vrlo je popularna metoda elo-rangiranja. Prvenstveno je razvijena za rangiranje i prognoziranje u šahovskim natjecanjima, a sve češće koristi se za rangiranje i u drugim sportovima. [1]

3.2.1. Poissonov model

Prvi modeli predviđanja nogometnih rezultata objavljeni su 80-ih godina. Modeliranje broja golova koje je momčad postigla kao Poissonova slučajna varijabla bio je jedan od prvih pristupa. Daljnji razvoj uključuje predviđanje rezultata lige i korištenje bivarijatne distribucije. [2]

Mnogi misle kako je broj golova koje je ekipa postigla u utakmici Poissonova varijabla. Posjed je važan aspekt nogometa i svaki put kada momčad ima loptu, ima priliku napasti i postići gol. Vjerojatnost P, da će napad rezultirati golom je naravno mala, ali broj puta kada momčad ima posjed lopte tijekom utakmice je velik. Upravo zato ako je P konstantan i napadi su neovisni, Poissonova aproksimacija se može vrlo dobro primijeniti. [3]

Poissonova distribucija diskretna je distribucija vjerojatnosti koja izražava vjerojatnost da će se određeni broj događaja dogoditi u fiksnom vremenskom intervalu, pod uvjetom da se ti događaji događaju neovisno s poznatom konstantnom brzinom. [4]

Neka ($X_{i,j}$, $Y_{i,j}$) predstavlja slučajni ishod utakmice, pri čemu je $X_{i,j}$ predstavlja broj golova koje je zabila ekipa i koja igra na domaćem terenu protiv ekipe j koja igra u gostima i koja je zabila $Y_{i,j}$ golova. Rezultat se modelira dvjema slučajnim varijablama s Poissonovom distribucijom, pri čemu očekivani broj golova domaće ekipe iznosi λ, a gostujuće ekipe μ. [1]

$$Xi,j \sim \mathbf{Poisson}(\lambda)$$

$$Xi,j \sim \mathbf{Poisson}(\mu)$$

Parametri λ i μ dobiju se pomoću povijesnih rezultata utakmica koje su te ekipe odigrale protiv ekipa sličnih snaga. Parametri α_i , β_i predstavljaju spremnost napada i obrane momčadi i te analogno za ekipu j. [1]

$$\lambda = \lambda p(1+\gamma)(1+\alpha i)(1-\beta j)$$

$$\mu = \lambda p(1-\gamma)(1+\alpha j)(1-\beta i)$$

3.2.2. Binomni model

Binomni model je dobro uspostavljen alat u sportskoj analitici, posebno kada je u pitanju predviđanje ishoda koji se mogu kategorizirati u dva različita rezultata, kao što su uspjeh ili neuspjeh. U kontekstu sportova kao što su nogomet, bejzbol ili košarka, ovaj se model često koristi za analizu događaja kao što su šanse za postizanje pogotka u nogometu, postizanje slobodnog bacanja u košarci ili *home run* u bejzbolu. [18]

U nogometu je binomni model koristan pri analizi postotka konverzije udaraca u pogotke. Odnosno, može se upotrijebiti za predviđanje broja golova koje bi momčad mogla postići na temelju ukupnog broja udaraca koje su izveli i njihove preciznosti pucanja. [19]

Vjerojatnost da pojedini gol, iz skupa ukupnog broja golova na utakmici, postigne prva (domaća) ekipa određujemo formulom [1]

$$\boldsymbol{p = \frac{\lambda}{\mu} = \frac{(1+\gamma)(1+\alpha i)(1-\beta j)}{(1+\gamma)(1+\alpha i)(1-\beta j)+(1-\gamma)(1+\alpha j)(1-\beta i)}}$$

3.2.3. Parametri modela

Prosječan broj golova po ekipi i po nogometnoj utakmici λp jednak je zbroju svih golova domaćih ekipa x i golova gostujućih ekipa y podijeljen brojem 2n, gdje je n broj odigranih utakmica. [1]

$$\boldsymbol{\lambda p = \frac{\sum x + \sum y}{2 \cdot n}}$$

Snaga napada ekipe dobije se kao omjer prosječnog broja golova koje je ekipa postigla po utakmici i prosječnog broja golova po ekipi i utakmici umanjen za 1.[1]

$$\boldsymbol{\alpha i = \frac{xi}{\lambda p} - 1}$$

Parametar za snagu pokazuje koliko je ekipa jača odnosno slabija od prosjeka. Ako je za snagu napada dobivena vrijednost pozitivna, to znači da ekipa postiže u prosjeku toliko više pogodaka po utakmici u odnosu na prosječan iznos za sve ekipe. Ako je vrijednost negativna to znači da toliko posto manje pogodaka postiže o utakmici u odnosu na prosjek svih ekipa.

Snaga obrane momčadi dobije se kao omjer prosječnog broja golova po ekipi i prosječnog broja golova koje ekipa primi po utakmici umanjen za 1.[1]

$$\boldsymbol{\beta i = \frac{yi}{\lambda p} - 1}$$

Parametar za snagu pokazuje koliko je ekipa jača odnosno slabija od prosjeka. Ako je za snagu obrane dobivena vrijednost pozitivna, to znači da ekipa prima u prosjeku toliko manje golova po utakmici u odnosu na prosječan iznos za sve ekipe.

Ako je vrijednost negativna to znači da toliko posto više golova prima po utakmici u odnosu na prosjek svih ekipa.

Prednost domaćeg terena - pretpostavljamo da je ista za sve ekipe i jednaka je omjeru prosječnog broja golova koje postiže domaća ekipa naspram prosječnog broja golova po ekipi po utakmici. [1]

$$\boldsymbol{\gamma = \frac{h}{\lambda p} - 1}$$

Empirijska distribucija golova po utakmici računa se kao relativni broj utakmi ca s k golova u ukupnom broj odigranih utakmica n. [1]

$$\boldsymbol{pk = \frac{fk}{n}}$$

3.2.4. Ostale primjene modela

Poissonova distribucija može biti korisna za modeliranje događaja kao što je broj studenata koji su na ispitu postigli nisku i visoku ocjenu, određivanje mjesta defekata odnosno nesavršenosti u materijalima, također za određivanje nasumičnih točaka u prostoru, kao na primjer, mjesta sudara asteroida sa zemljom i mjesto drveća u šumi. [14]

U kliničkim ispitivanjima binomni model često se koristi za analizu uspjeha ili neuspjeha liječenja. Na primjer, u studiji u kojoj se pacijentima daje novi lijek, ishod može biti hoće li svaki pacijent pozitivno reagirati na liječenje ili ne. Binomni model pomaže u procjeni vjerojatnosti uspjeha (npr. stope izlječenja) na određenom uzorku pacijenata. [18]

4. Implementacija modela za predviđanje rezultata u programskom jeziku Python

U nastavku je prikazana implementacija Poissonovog modela te binomnog modela za predviđanje rezultata pojedinih utakmica promatrane sezone u odabranoj nogometnoj ligi, te za predviđanje finalnog poretka ekipa na kraju sezone temeljenog na podacima iz prvog dijela sezone koristeći programski jezik Python. Svaki isječak koda bit će objašnjen te će biti prikazani rezultati istih.

4.1. Promatrana nogometna liga – Serie A

Serie A, elitni rang talijanskog nogometa, ima legendarnu povijest koja datira od njenog službenog osnivanja 1898. Tijekom svog postojanja, liga je doživjela 122 završene sezone, a trenutna sezona u tijeku 2024-25 je 123. sezona. Liga, koja djeluje u formatu kružnog natjecanja, uključuje 20 timova koji se natječu svake godišnje [10].

Juventus drži rekord za najviše ligaških naslova, s 36 Scudetta u svojoj povijesti. Ostale dominantne momčadi uključuju AC Milan i Inter Milan, a obje su značajno pridonijele povijesti lige, posebno tijekom značajnih razdoblja poput 1990-ih i ranih 2000-tih kada se Serie A smatrala jednom od najjačih liga na svijetu. [7]

Primarna motivacija pri odabiru Serie A kao promatrane lige bila je ta što se za nju vežu činjenice i rasprave kako je upravo ova liga najteža za predviđanje kako međusobnih rezultata, tako i finalnih poredaka. Uz to mnogi klubovi dolaze iz istih regija, pa je upravo zbog toga Serie A je poznata po svojim intenzivnim regionalnim derbijima, kao što su *Derby della Madonnina* između Inter Milana i AC Milana ili *Derby d'Italia* između Juventusa i Inter Milana. Bez obzira na moguća kraća putovanja na gostujuće utakmice prednost domaćeg terena i dalje postoji te ćemo saznati njen faktor u nastavku rada. [7]

4.2. Predviđanje konačnog rezultata utakmice

Prvi korak za predviđanje konačnog rezultata utakmice je implementacija spomenutih parametara u programskom jeziku Python. Kao što je spomenuto promatrana liga je talijanska nogometna liga Serie A, a promatrana sezona je sezona 2020/2021 godine.

U Tablici 1 prikazano je stanje ekipa na kraju prvog dijela sezone. Tablica 1 sadrži sve podatke koji su nam potrebni za implementaciju i izračun vrijednosti parametara kako bismo mogli što točnije predvidjeti ishode utakmica. Podaci koji su nama bitni su pozicija na tablici, broj odigranih utakmica u prvom dijelu sezone, broj postignutih pogodaka (GZ), te broj primljenih pogodaka (GP).

Tablica: 1:Prikaz top 3 vjerojatnosti završne pozicije na kraju sezone

Pozicija	**Ekipa**	**Odigrano**	**Pobjede**	**Neriješeno**	**Porazi**	**GZ**	**GP**	**GR**	**Bodovi**
1	AC Milan	19	13	4	2	39	19	20	43
2	Inter Milan	19	12	5	2	45	23	22	41
3	Roma	19	11	4	4	41	29	12	37
4	Juventus	19	10	6	3	37	18	19	36
5	Atalanta	19	10	6	3	45	23	22	36
6	Napoli	19	12	1	6	41	19	22	37
7	Lazio	19	10	4	5	32	26	6	34
8	Sassuolo	19	8	7	4	32	28	4	31
9	Hellas Verona	19	7	6	6	22	18	4	27
10	Sampdoria	19	8	2	9	28	29	-1	26
11	Benevento	19	6	4	9	21	36	-15	22
12	Bologna	19	5	5	9	24	33	-9	20
13	Udinese	19	4	6	9	20	28	-8	18
14	Spezia	19	4	6	9	26	36	-10	18
15	Fiorentina	19	4	6	9	18	29	-11	18
16	Genoa	19	4	6	9	18	30	-12	18
17	Cagliari	19	3	5	11	23	36	-13	14
18	Torino	19	2	8	9	29	38	-9	14
19	Parma	19	2	7	10	14	36	-22	13
20	Crotone	19	3	3	13	22	43	-21	12

(Izvor: [8], obrada: Autor)

Kako bismo pripremili potrebne podatke za predviđanje konačnog rezultata pojedine utakmice moramo izračunati varijable koje će nam služiti kao konstante. Te varijable su, prosjek broja golova po ekipi po utakmici, parametri spremnosti napada i obrane za jedinstveni za svaku ekipu, te prednost domaćeg terena.

Prosjek broja golova po ekipi na jednoj nogometnoj utakmici je parametar koji nam je vrlo važan te se koristi skoro u svakoj jednadžbi u nastavku. Taj prosjek, odnosno lambda-p, računamo kao omjer ukupnog broja golova, koji iznosi 577 golova u prvom dijelu sezone, odnosno u odigranih 190 utakmica**.**

$$\boldsymbol{\lambda p = \frac{577}{2 \cdot 190} = 1.518}$$

Kao što možemo vidjeti na Slici 1, kada smo implementirali formulu u Python dobili smo vrijednost lambde-p koju smo za potrebe daljnjeg računanje zaokružili na tri decimale.

```
# Izracun prosjecnog broja golova ekipe po utakmici (λp)
total_goals = df["Broj postignutih pogodaka"].sum() + df["Broj primljenih pogodaka"].sum()
total_matches = df["Odigranih utakmica"].sum()
lambda_p = total_goals / (2 * total_matches)
print(f"λp = {lambda_p}")

λp = 1.518421052631579
```

Slika 1: Isječak koda koji računa proječan broj golova po ekipi na nogometnoj utakmici

Sljedeći korak u pripremi podataka je izračunati parametre spremnosti napada i obrane za svaku momčad. Parametri spremnost napada (αi) i spremnost obrane (βi) za slučajno odabranu momčad, koja je u ovom primjeru, Lazio:

$$\boldsymbol{\alpha i = \left(\frac{32 \div 19}{1.58}\right) - 1 = 0.109} \qquad \boldsymbol{\beta i = \left(\frac{26 \div 19}{1.58}\right) - 1 = -0.099}$$

Pozitivna vrijednost spremnosti napada označava za koliko posto u prosjeku ekipa daje više golova od ostalih ekipa, a negativna vrijednost spremnosti obrane označava za koliko posto manje golova u prosjeku ekipa prima naspram protivničkih ekipa.

Na Slici 2 možemo vidjeti isječak koda gdje je implementirana logika računanja parametara spremnosti napada i obrane momčadi. Zbog lakšeg tumačenja podataka u Tablici 2 su uneseni izlazni podaci programskog koda. Također kao što možemo vidjeti ako pogledamo podatke za ekipu Lazio, možemo vidjeti kako smo uistinu dobili točan izračun spremnosti napada i obrane za momčad.

```
# Izracun parametara spremnosti za obje ekipe
df["α_i"] = ((df["Broj postignutih pogodaka"] / df["Odigranih utakmica"]) / lambda_p) - 1
df["β_i"] = ((df["Broj primljenih pogodaka"] / df["Odigranih utakmica"]) / lambda_p)  - 1
print(df[["Ekipa", "α_i", "β_i"]])
```

```
            Ekipa       α_i       β_i
0        AC Milan  0.351820 -0.341421
1     Inter Milan  0.559792 -0.202773
2            Roma  0.421144  0.005199
3        Juventus  0.282496 -0.376083
4        Atalanta  0.559792 -0.202773
5          Napoli  0.421144 -0.341421
6           Lazio  0.109185 -0.098787
7        Sassuolo  0.109185 -0.029463
8   Hellas Verona -0.237435 -0.376083
9       Sampdoria -0.029463  0.005199
10      Benevento -0.272097  0.247834
11        Bologna -0.168111  0.143847
12        Udinese -0.306759 -0.029463
13         Spezia -0.098787  0.247834
14     Fiorentina -0.376083  0.005199
15          Genoa -0.376083  0.039861
16       Cagliari -0.202773  0.247834
17         Torino  0.005199  0.317158
18          Parma -0.514731  0.247834
19        Crotone -0.237435  0.490468
```

Slika 2: Isječak programskog koda, te rezultati spremnosti napada i obrane za ekipe

Tablica: 2:Parametri spremnosti napada i obrane za ekipe

Ekipa	**Parametar spremnosti napada α_i**	**Parametar spremnosti obrane β_i**
AC Milan	0.35182	-0.341421
Inter Milan	0.559792	-0.202773
Roma	0.421144	0.005199
Juventus	0.282496	-0.376083
Atalanta	0.559792	-0.202773
Napoli	0.421144	-0.341421
Lazio	0.109185	-0.098787
Sassuolo	0.109185	-0.029463
Hellas Verona	-0.237435	-0.376083
Sampdoria	-0.029463	0.005199
Benevento	-0.272097	0.247834
Bologna	-0.168111	0.143847
Udinese	-0.306759	-0.029463
Spezia	-0.098787	0.247834
Fiorentina	-0.376083	0.005199
Genoa	-0.376083	0.039861
Cagliari	-0.202773	0.247834
Torino	0.005199	0.317158
Parma	-0.514731	0.247834
Crotone	-0.237435	0.490468

Posljednja konstanta koju moramo izračunati je prednost domaćeg terena. Prednost domaćeg terena se računa tako da sumiramo broj golova domaćih ekipa (319), te podijelimo s brojem odigranih utakmica (190). Zatim dobivenu vrijednost prosječnog broja golova domaćina podijelimo s $\boldsymbol{\lambda p}$, te na kraju od ukupne vrijednosti oduzmemo 1.

$$\gamma = \left(\frac{\mathbf{319 \div 190}}{\mathbf{1.518}}\right) - \mathbf{1} = \mathbf{0.106}$$

```
# Izracun prednosti domaceg terena
gamma = ((home_team_goals/total_matches) / lambda_p)-1
print(gamma)

0.10571923743500866
```

Slika 3: Isječak koda te rezultat vrijednosti prednosti domaćeg terena

Kao što je vidljivo na Slici 3, nakon implementacije formule za izračun vrijednosti prednosti domaćeg terena dobili smo vrijednost koju smo i očekivali, te smo ju za daljnje potrebe zaokružili na tri decimale.

4.3. Predviđanje rezultata utakmice koristeći Poissonov model

Sada kada imamo potrebne konstante koje ćemo koristiti u predviđanju rezultata nogometne utakmice, možemo implementirati promatrane modele. Započeti ćemo s Poissonovim modelom. Kao slučajni primjer uzeo sam utakmicu između Hellas Verone i Intera koja se igrala u 14. kolu prvog dijela sezone.

Na početku inicijaliziramo već poznate vrijednosti, kao što su prosječan broja golova po ekipi na jednoj nogometnoj utakmici, prednost domaćeg terena te spremnosti napada i obrane za obje ekipe. Zatim računamo očekivani broj golova za obje ekipe. Za Veronu (*lambda_verona*) iznosi 1.54, a za Inter (*mu_inter*) iznosi 2.91.

Nakon toga postavljamo matricu vjerojatnosti. Maksimalan broj golova koji ekipa može postići ćemo postaviti na 4 zbog realnosti stavke. Pomoću *for* petlje iteriramo kroz mogući broj golova za Veronu (i) i mogući broj golova za Inter (j), te za svaku iteraciju gola računamo vjerojatnost pomoću formule za Poissonov model.

Na kraju normaliziramo matricu kako bi zbrojene vrijednosti davale sumu od 100%, to radimo zbog toga što smo stavili limit na maksimalno 4 gola po ekipi, a kako postoji šansa da ekipa zabije više golova, ali nismo tu šansu uključili u promatrani skup, matrica se normalizira kako bismo dobili što precizniju vrijednost za promatrane rezultate.

```
import numpy as np
import scipy.stats as stats
import pandas as pd

lambda_p = 1.518
gamma = 0.106

# Vrijednosti parametara spremnosti napada i obrane
alpha_verona = -0.237435
beta_verona = -0.376083
alpha_inter = 0.559792
beta_inter = -0.202773

# Izračun λ za domacu ekipu te μ za gostujuću ekipu
lambda_verona = lambda_p * (1 + gamma) * (1 + alpha_verona) * (1 -
beta_inter)
mu_inter = lambda_p * (1 - gamma) * (1 + alpha_inter) * (1 -
beta_verona)

max_goals = 5
prob_matrix = np.zeros((max_goals, max_goals))

# Petlja koja racuna vjerojatnost svih mogucih kombinacija rezultata
(limit 4 zabijena gola po ekipi)
for i in range(max_goals):
    for j in range(max_goals):
        prob_verona = stats.poisson.pmf(i, lambda_verona)
        prob_inter = stats.poisson.pmf(j, mu_inter)
        prob_matrix[i, j] = prob_verona * prob_inter

total_prob = np.sum(prob_matrix)
# Normalizacija matrice (da vrijednosti daju 100%)
prob_matrix_normalized = (prob_matrix / total_prob) * 100

df = pd.DataFrame(prob_matrix_normalized,
                  columns=[f'Inter - {x}' for x in
range(max_goals)],
                  index=[f'Hellas Verona| {x}' for x in
range(max_goals)])

print("Matrica vjerojatnosti rezultata (u %):\n")
print(df)

# Find the top 3 most likely scores
flat_probs = prob_matrix_normalized.flatten()
top_3_indices = flat_probs.argsort()[-3:][::-1]
top_3_scores = [(index // max_goals, index % max_goals) for index in
top_3_indices]

print("\nTop 3 najvjerovatnija rezultata:")
for idx, (verona_goals, inter_goals) in enumerate(top_3_scores):
    print(f"{idx + 1}: | Hellas Verona {verona_goals} -
{inter_goals} Inter | s vjerojatnosti
{prob_matrix_normalized[verona_goals, inter_goals]:.2f}%")
```

Na Slici 4 prikazani su izlazni podaci spomenute implementacije Poissonovog modela. Tri najvjerojatnija rezultata bila su 1:2 (9.36%), 1:3 (9.09%) i 2:2 (7.21%). Utakmica između Verone i Intera odigrana 23. prosinca 2020. završila je rezultatom od 1:2, a upravo je taj rezultat bio najvjerojatniji ishod pomoću Poissonovog modela.

```
Matrica vjerojatnosti rezultata (u %):

                  Inter - 0  Inter - 1  Inter - 2  Inter - 3  Inter - 4
Hellas Verona| 0   1.433128   4.174510   6.079895   5.903308   4.298887
Hellas Verona| 1   2.206847   6.428252   9.362321   9.090398   6.619779
Hellas Verona| 2   1.699142   4.949375   7.208435   6.999070   5.096839
Hellas Verona| 3   0.872160   2.540484   3.700046   3.592581   2.616177
Hellas Verona| 4   0.335756   0.978012   1.424409   1.383038   1.007151

Top 3 najvjerovatnija rezultata:
1: | Hellas Verona 1 - 2 Inter | s vjerojatnosti 9.36%
2: | Hellas Verona 1 - 3 Inter | s vjerojatnosti 9.09%
3: | Hellas Verona 2 - 2 Inter | s vjerojatnosti 7.21%
```

Slika 4: Izlazni podaci predviđanja rezulatata utakmice Verona-Inter temeljem Poissonovog modela

Tablica: 3 Prikaz vjerojatnosti pojedinog rezulatata utakmice Verona-Inter temeljem Poissonovog modela

Poisson		Inter				
Hellas Verona		0	1	2	3	4
	0	1.43%	4.17%	6.08%	5.9%	4.3%
	1	2.21%	6.43%	9.36%	9.09%	6.62%
	2	1.7%	4.95%	7.21%	7%	5.1%
	3	0.87%	2.54%	3.7%	3.59%	2.62%
	4	0.34%	0.98%	1.42%	1.38%	1.01%

4.4. Predviđanje rezultata utakmice koristeći Binomni model

Rezultat iste utakmice iz prethodnog poglavlja ćemo sada pokušati predvidjeti koristeći binomni model, kako bismo mogli usporediti točnost i korisnost oba modela. Da bismo implementirali binomni model potrebna nam je empirijska distribucija golova po utakmici koju nismo koristili u Poissonovom modelu, zato ćemo ju sada izračunati.

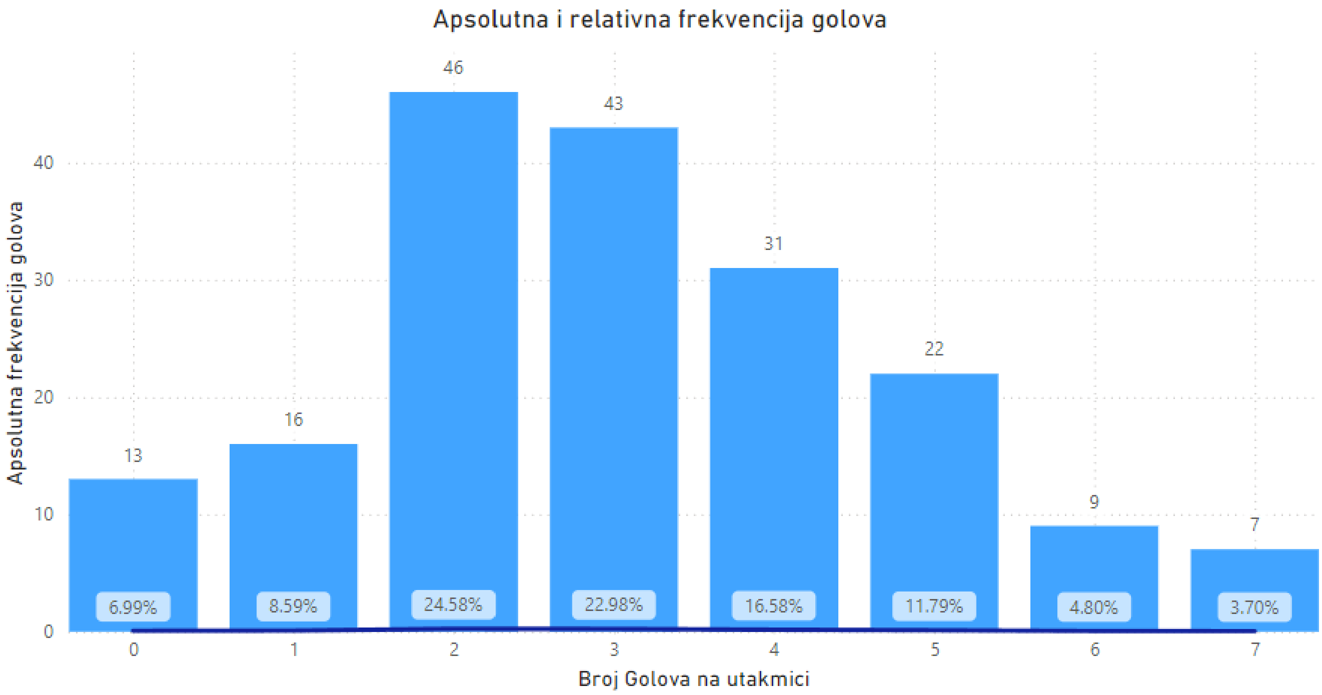
Za analizu empirijske distribucije golova odlučio sam se gledati samo utakmice koje su imale maksimalno sedam golova. Iz analize su izuzete dvije utakmice koje su imale više od sedam golova, te jedna utakmica koja se nije odigrala, već je rezultat od 3:0 bio dodijeljen jednoj ekipi.

Imajući to na umu u Tablici 4 možemo vidjeti raspodjelu podataka. Stupac „Apsolutna frekvencija golova“ sadrži broj utakmica s određenim brojem golova, npr. u 13 utakmica nije postignut nijedan pogodak, a „Relativna frekvencija golova“ sadrži vrijednost proporcionalnu vrijednosti određenog broja golova, izraženu u postotcima, npr. 7% utakmica je završilo bez pogotka. Vrijednost relativne frekvencije dobije se tako što smo vrijednost apsolutne frekvencije podijelili s ukupnim brojem utakmica te pomnožili sa 100 kako bismo dobili vrijednosti u postocima

Tablica 4: Apsolutna i relativna frekvencija broja golova na utakmicama prvog dijela sezone

Broj Golova na utakmici	Apsolutna frekvencija golova	Relativna frekvencija golova
0	13	7%
1	16	8.6%
2	46	24.6%
3	43	23%
4	31	16.6%
5	22	11.8%
6	9	4.8%
7	7	3.7%

(Izvor: [9], obrada: Autor)



Grafikon 1: Apsolutna i relativna frekvencija golova

Implementaciju binomnog modela započinjemo isto kao i Poissonovog. Na početku inicijaliziramo već poznate vrijednosti, kao što su prosječan broja golova po ekipi na jednoj nogometnoj utakmici, prednost domaćeg terena te spremnosti napada i obrane za obje ekipe. Zatim računamo očekivani broj golova za obje ekipe. Za Veronu (*lambda_verona*) iznosi 1.54, a za Inter (*mu_inter*) iznosi 2.91.

Novost je to što računamo vjerojatnost da pojedini gol postigne Verona, a ona iznosi p = 0.346, te inicijaliziramo vrijednosti empirijske vjerojatnosti iz Tablice 4.

Zatim opet postavljamo matricu, ali ovaj put vrijednost max_goals postavljamo na 8 zato što gledamo da je maksimalan mogući broj golova na utakmici 7. Vanjska *for* petlja služi za iteriranje kroz mogući broj golova na utakmici, a u unutarnjoj *for* petlji računamo vjerojatnosti za pojedini broj golova promatranih ekipa.

```
import numpy as np
import pandas as pd
import math

gamma = 0.106
lambda_p = 1.518
alpha_verona = -0.237435
beta_verona = -0.376083
alpha_inter = 0.559792
```

```
beta_inter = -0.202773

# Izračun λ za domacu ekipu te μ za gostujuću ekipu
lambda_verona = lambda_p * (1 + gamma) * (1 + alpha_verona) * (1 -
beta_inter)
mu_inter = lambda_p * (1 - gamma) * (1 + alpha_inter) * (1 -
beta_verona)

p = lambda_verona / (lambda_verona + mu_inter)  # Vjerojatnost da
Verona zabija gol
empirical_probs = [0.07, 0.086, 0.246, 0.23, 0.166, 0.118, 0.048,
0.037]

max_goals = 8
result_matrix = np.zeros((max_goals, max_goals))

# Racunamo vjerojatnost za svaki od golova u empirijskoj
vjerojatnosti Z
for Z in range(max_goals):
    for k in range(Z + 1):
        binom_coeff = math.comb(Z, k)
        prob_k = binom_coeff * (p ** k) * ((1 - p) ** (Z - k))
        overall_prob = prob_k * empirical_probs[Z]
        result_matrix[k, Z - k] += overall_prob

# Normalizacija matrice
total_prob = np.sum(result_matrix)
result_matrix_normalized = (result_matrix / total_prob) * 100

df = pd.DataFrame(result_matrix_normalized,
                  columns=[f'Inter - {x}' for x in
range(max_goals)],
                  index=[f'Hellas Verona| {x}' for x in
range(max_goals)])

print("Matrica vjerojatnosti rezultata (u %):\n")
print(df)

flat_probs = result_matrix_normalized.flatten()
top_3_indices = flat_probs.argsort()[-3:][::-1]
top_3_scores = [(index // max_goals, index % max_goals) for index in
top_3_indices]

print("\nTop 3 najvjerovatnija rezultata:")
for idx, (verona_goals, inter_goals) in enumerate(top_3_scores):
    print(f"{idx + 1}: | Hellas Verona {verona_goals} -
{inter_goals} Inter | s vjerojatnosti
{result_matrix_normalized[verona_goals, inter_goals]:.2f}%")
```

Na Slici 5 prikazani su izlazni podaci spomenute implementacije binomnog modela. U Tablici 5 prikazani su rezultati ishoda gdje je svaka ekipa zabila maksimalno 4 pogotka, kako bismo imali istu usporedbu kao i s Poissonovim modelom.

Tri najvjerojatnija rezultata bila su 1:1 (11.12%), 0:2 (10.52%) i 1:2 (10.20%).Kao što je spomenuto promatrana utakmica. završila je rezultatom od 1:2, a taj je rezultat bio treći najvjerojatniji ishod pomoću binomnog modela.

```
Matrica vjerojatnosti rezultata (u %):

                   Inter - 0  Inter - 1  Inter - 2  Inter - 3  Inter - 4  \
Hellas Verona| 0    6.993007   5.620265  10.516855   6.432370   3.036992
Hellas Verona| 1    2.971144  11.119435  10.201385   6.422003   3.732905
Hellas Verona| 2    2.939135   5.392944   5.092472   3.946787   1.575385
Hellas Verona| 3    0.950324   1.794751   2.086462   1.110433   0.979903
Hellas Verona| 4    0.237198   0.551502   0.440271   0.518024   0.000000
Hellas Verona| 5    0.058310   0.093099   0.164312   0.000000   0.000000
Hellas Verona| 6    0.008203   0.028954   0.000000   0.000000   0.000000
Hellas Verona| 7    0.002187   0.000000   0.000000   0.000000   0.000000

                   Inter - 5  Inter - 6  Inter - 7
Hellas Verona| 0    1.412245   0.375804   0.189502
Hellas Verona| 1    1.192010   0.701261   0.000000
Hellas Verona| 2    1.112161   0.000000   0.000000
Hellas Verona| 3    0.000000   0.000000   0.000000
Hellas Verona| 4    0.000000   0.000000   0.000000
Hellas Verona| 5    0.000000   0.000000   0.000000
Hellas Verona| 6    0.000000   0.000000   0.000000
Hellas Verona| 7    0.000000   0.000000   0.000000

Top 3 najvjerovatnija rezultata:
1: | Hellas Verona 1 - 1 Inter | s vjerojatnosti 11.12%
2: | Hellas Verona 0 - 2 Inter | s vjerojatnosti 10.52%
3: | Hellas Verona 1 - 2 Inter | s vjerojatnosti 10.20%
```

Slika 5:Izlazni podaci predviđanja rezulatata utakmice Verona-Inter temeljem binomnog modela

Tablica: 5: Prikaz vjerojatnosti pojedinog rezulatata utakmice Verona-Inter temeljem binomnog modela

Binomni	Inter					
Hellas Verona		0	1	2	3	4
	0	7%	5.62%	10.52%	6.43%	3.04%
	1	2.97%	11.12%	10.20%	6.42%	3.73%
	2	2.94%	5.39%	5.09%	3.95%	1.58%
	3	0.95%	1.79%	2.09%	1.11%	0.98%
	4	0.24%	0.55%	0.44%	0.52%	0%

4.5. Predviđanje generalnog ishoda utakmice

Zasada smo implementirali Poissonov i binomni model u Pythonu, kako bismo pokušali predvidjeti stvarni rezultat na utakmici. Ali što ako nas ne zanima stvarni rezultat utakmice, nego samo generalni ishod utakmice, odnosno zanima nas vjerojatnost pobjede domaće ekipe, neriješenog rezultata ili pobjede gostujuće ekipe. Ako nas zanima općeniti ishod pojedine utakmice također možemo koristiti Poissonov i binomni model, a implementacija traženog u programskom jeziku Python, bit će prikazana u nastavku.

4.5.1. Generalni ishod Poissonovim modelom

Implementaciju Poissonovog modela za izračun općenitog ishoda utakmice započinjemo tako da računamo vjerojatnosti pobjede domaćeg tima, neodlučenog rezultata, te pobjede gostujućeg tima.

Pobjeda domaćeg tima računa se tako da se u matrici zbrajaju vrijednosti po stupcima i redovima ispod glavne dijagonale.

Neodlučeni ishod utakmice računa se tako da se u matrici zbrajaju vrijednosti koje se nalaze na glavnoj dijagonali.

Pobjeda gostujućeg tima računa se tako da se u matrici zbrajaju vrijednosti po stupcima i redovima iznad glavne dijagonale.

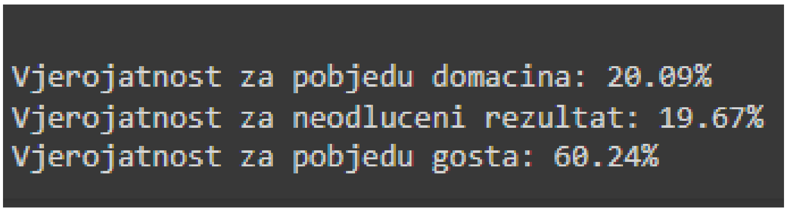
Zatim nam je bitno da zbrojimo sve vrijednosti u matrici, kako bismo dobili cjelinu, odnosno zbroj od 100%. Nakon toga izračunate vrijednosti pobjede domaćina, neodlučenog rezultata te pobjede gostujućeg tima podijelimo s ukupnim zbrojem vrijednosti u matrici. Na kraju dobivene vrijednosti pomnožimo sa 100 kako bismo dobili vjerojatnost za svaki ishod u postotcima.

```
# Izracun generalnog ishoda utakmice
p_domaci_pobjeda = np.sum(np.tril(prob_matrix_normalized, -1))
p_nerijeseno = np.sum(np.diag(prob_matrix_normalized))
p_gost_pobjeda = np.sum(np.triu(prob_matrix_normalized, 1))

total_prob = p_domaci_pobjeda + p_nerijeseno + p_gost_pobjeda
p_domaci_pobjeda /= total_prob
p_nerijeseno /= total_prob
p_gost_pobjeda /= total_prob

p_domaci_pobjeda *= 100
p_nerijeseno *= 100
p_gost_pobjeda *= 100

print(f"\nVjerojatnost za pobjedu domacina:
{p_domaci_pobjeda:.2f}%")
print(f"Vjerojatnost za neodluceni rezultat: {p_nerijeseno:.2f}%")
print(f"Vjerojatnost za pobjedu gosta: {p_gost_pobjeda:.2f}%")
```



Slika 6: Izlazni podaci implementacije za izračun općenitog ishoda utakmice Poissonovim modelom

4.5.2. Generalni ishod binomnim modelom

Implementacija binomnog modela za izračun općenitog ishoda utakmice zapravo je ista kao i implementacija Poissonovog modela, stoga nema potrebe ponavljati korake.

```
# Izracun generalnog ishoda utakmice
p_home_win = np.sum(np.tril(result_matrix_normalized, -1))
p_draw = np.sum(np.diag(result_matrix_normalized))
p_away_win = np.sum(np.triu(result_matrix_normalized, 1))

total_prob = p_home_win + p_draw + p_away_win
```

```
p_home_win /= total_prob
p_draw /= total_prob
p_away_win /= total_prob

p_home_win *= 100
p_draw *= 100
p_away_win *= 100

print(f"\nVjerojatnost za pobjedu domacina: {p_home_win:.2f}%")
print(f"Vjerojatnost za neodluceni rezultat: {p_draw:.2f}%")
print(f"Vjerojatnost za pobjedu gosta: {p_away_win:.2f}%")
```

```
Vjerojatnost za pobjedu domacina: 18.24%
Vjerojatnost za neodluceni rezultat: 24.32%
Vjerojatnost za pobjedu gosta: 57.45%
```

Slika 7:Izlazni podaci implementacije za izračun općenitog ishoda utakmice binomnim modelom

Tablica 6: Usporedba rezultata ishoda utakmice promatranih modela

Ishod utakmice	Poissonov model	Binomni model
Pobjeda domaćina	20.09%	18.24%
Neodlučeni rezultat	19.67%	24.32%
Pobjeda gosta	60.24%	57.45%

Kao što je vidljivo iz Tablice 6, oba modela su predvidjeli slične vjerojatnosti za promatrane ishode. Kada bismo tumačili dobivene rezultate sa top 3 najvjerojatnija rezultata za svaki model, mogli bismo uočiti povezanost s prva dva najvjerojatnija rezultata koristeći Poissonov model, te većim postotkom pobjede gostujuće momčadi naspram binomnog modela. Također veći postotak za neodlučeni ishod koristeći binomni model možemo povezati s činjenicom da je najvjerojatniji rezultat bio neodlučeni rezultat 1:1.

4.6. Predviđanje poretka na kraju sezone

Za kraj praktičnog dijela implementirati ćemo Poissonov model kako bismo mogli predvidjeti finalno stanje na tablici na kraju sezone. Implementacija funkcionira na sljedećem principu. U Tablici 7 nalaze se potrebni podaci podaci, ti podaci uključuju mjesto na tablici na kraju prvog dijela sezone, broj utakmica odigrano kod kuće te u gostima, i broj golova postignutih kod kuće te broj golova postignutih na gostovanjima.

Započinjemo tako što unesemo Tablicu 7 zajedno s podacima, te prvo računamo prosječan broj golova na utakmici, tako što zbrojimo sve golove te dobivenu vrijednost podjelimo sa ukupnim brojem utakmica. Nakon toga nam je bitno izračunati prosječan broj golova za domaću ekipu i za gostujuću ekipu. Za domaću ekipu zbrojimo sve vrijednosti u stupcu *postignuto doma* te ih podjelimo sa ukupnim brojem utakmica odigranih na domaćem terenu. Na istom principu računamo i prosjek za gostujuću ekipu.

Zatim računamo faktore vezane za pojedinu ekipu, a to se radi tako da se ukupan broj golova i ukupan broj utakmica podijeli međusobno, te se dobivena vrijednost podijeli sa vrijednosti '*e*' koja je očekivani broj golova za svaku ekipu.

Na kraju uz pomoć već poznate formule za *lambdu*, računamo Poissonove parametre za utakmicu na neutralnom stadionu (*Lambda*), za utakmicu na domaćem stadionu (*Lambda Domacin*) i za utakmicu na gostujućem stadionu (*Lambda Gost*). Izlazni podaci se nalaze na Slici 8, te u Tablici 8 zbog lakše preglednosti.

Table 7: Stanje na tablici uz broj golova nakon prvog dijela sezone

Ekipa	**Odigrano doma**	**Odigrano u gostima**	**Postignuto doma**	**Postignuto u gostima**	**Bodovi**
AC Milan	10	9	22	17	43
Inter Milan	9	10	30	15	41
Roma	10	9	24	17	37
Juventus	10	9	25	12	36
Atalanta	10	9	27	18	36
Napoli	9	10	24	17	37
Lazio	9	10	16	16	34
Sassuolo	9	10	16	16	31
Hellas Verona	9	10	11	11	27

Sampdoria	9	10	16	12	26
Benevento	9	10	10	11	22
Bologna	9	10	13	11	20
Udinese	9	10	8	12	18
Spezia	9	10	10	16	18
Fiorentina	10	9	10	8	18
Genoa	9	10	11	7	18
Cagliari	9	10	14	9	14
Torino	10	9	13	16	14
Parma	10	9	9	5	13
Crotone	10	9	12	10	12

(Izvor: [9], obrada: Autor)

```
import pandas as pd

df = pd.read_csv('/content/final_rankings_seriea.csv')

# Prvo izracunamo prosjek golova po utakmici
total_goals = df['Postignuto doma'].sum() + df['Postignuto u
gostima'].sum()
total_games = df['Odigrano doma'].sum() + df['Odigrano u
gostima'].sum()
lambda_avg = total_goals / total_games

# Zatim izracunamo prosjek golova za domace i gostujuce ekipe
lambda_h_avg = df['Postignuto doma'].sum() / df['Odigrano
doma'].sum()
lambda_a_avg = df['Postignuto u gostima'].sum() / df['Odigrano u
gostima'].sum()

# Izracunamo prednost domaceg terena
h = lambda_h_avg / lambda_avg - 1

# Racunamo faktore specificne za ekipu (t)
df['Goals_Total'] = df[' Postignuto doma'] + df[' Postignuto u
gostima']
df['Games_Total'] = df['Odigrano doma'] + df['Odigrano u gostima']
df['g'] = df['Goals_Total'] / df['Games_Total']

# Racunamo ocekivani broj golova za svaku ekipu (e)
df['e'] = (lambda_h_avg * df['Odigrano doma'] + lambda_a_avg *
df['Odigrano u gostima']) / df['Games_Total']
df['t'] = df['g'] / df['e'] - 1

# Racunamo λ za svaku ekipu
df['lambda_h'] = lambda_avg * (1 + h) * (1 + df['t'])
df['lambda_a'] = lambda_avg * (1 - h) * (1 + df['t'])
df['lambda'] = lambda_avg * (1 + df['t'])  # Vrijednost za neutralni
stadion

df['lambda_h'] = df['lambda_h'].round(3)
df['lambda_a'] = df['lambda_a'].round(3)
```

```
df['lambda'] = df['lambda'].round(3)

#
poisson_param = df[['Ekipa', 'lambda', 'lambda_h', 'lambda_a']]
poisson_param.set_index('Ekipa', inplace=True)
print("Poisson parametri za prvi dio sezone:")
print(poisson_param)
```

```
Poisson parametri za prvi dio sezone:
                lambda  lambda_h  lambda_a
Ekipa
AC Milan         2.037     2.290     1.783
Inter Milan      2.381     2.677     2.084
Roma             2.141     2.408     1.875
Juventus         1.932     2.173     1.692
Atalanta         2.350     2.643     2.058
Napoli           2.169     2.439     1.899
Lazio            1.693     1.904     1.482
Sassuolo         1.693     1.904     1.482
Hellas Verona    1.164     1.309     1.019
Sampdoria        1.481     1.666     1.297
Benevento        1.111     1.249     0.973
Bologna          1.270     1.428     1.112
Udinese          1.058     1.190     0.926
Spezia           1.376     1.547     1.204
Fiorentina       0.940     1.057     0.823
Genoa            0.952     1.071     0.834
Cagliari         1.217     1.368     1.065
Torino           1.515     1.703     1.326
Parma            0.731     0.822     0.640
Crotone          1.149     1.292     1.006
```

Slika 8: Izlazni podaci implementacije izračuna Poissonovih parametara za predviđanje finalnog poretka

Tablica 8: Poissonovi parametri za prvi dio sezone

Ekipa	**Lambda**	**Lambda Domaćin**	**Lambda Gost**
AC Milan	2.037	2.290	1.783
Inter Milan	2.381	2.677	2.084
Roma	2.141	2.408	1.875
Juventus	1.932	2.173	1.692
Atalanta	2.350	2.643	2.058
Napoli	2.169	2.439	1.899
Lazio	1.693	1.904	1.482
Sassuolo	1.693	1.904	1.482

Hellas Verona	1.164	1.309	1.019
Sampdoria	1.481	1.666	1.297
Benevento	1.111	1.249	0.973
Bologna	1.270	1.428	1.112
Udinese	1.058	1.190	0.926
Spezia	1.376	1.547	1.204
Fiorentina	0.940	1.057	0.823
Genoa	0.952	1.071	0.834
Cagliari	1.217	1.368	1.065
Torino	1.515	1.703	1.326
Parma	0.731	0.822	0.640
Crotone	1.149	1.292	1.006

Drugi dio implementacija radi na sljedećem principu. Učitamo podatke i Poissonovim parametrima (Tablica 8), te učitamo podatke o trenutnoj poziciji na tablici. Zatim na temelju učitanih podataka definiramo funkciju za simuliranje utakmice (`def simulate_match`). Funkcija radi na principu da koristeći Poissonove parametre na temelju očekivanih prosjeka golova odabere random vrijednost broja golova, te na temelju toga računa konačan ishod utakmice. Također, kako bismo što točnije predvidjeli konačnu poziciju na kraju sezone, definiramo funkciju koja će nam uzeti u obzir poziciju ekipe na prvom dijelu sezone (`def adjust_points`).

Sljedeće radimo funkciju koja će simulirati ostatak sezone (`def simulate_season`). Funkcija na početku postavlja bodove na nula, te simulira utakmice među ekipama na temelju izračunatih parametara, te na kraju nadoda prilagođene bodove iz prvog dijela sezone.

Za kraj stvaramo funkciju koja će izvesti simulaciju 10.000 puta na temelju Monte Carlo principa (`def monte_carlo_simulation`). Deset tisuća puta se pokrene funkcija `def simulate_season`, te se ekipe sortiraju prema poziciji od prvog do zadnjeg mjesta. Zapisane pozicije se zatim koriste u funkciji (`def determine_final_rankings`) koja nam računa koliko je puta koja ekipa završila na određenoj poziciji, te sam ispisuje prve tri najvjerojatnije pozicije za svaku ekipu.

```
lambdas = df.set_index('Ekipa')[['lambda', 'lambda_h',
'lambda_a']].T.to_dict()

# Postavljanje trenutne pozicije na tabeli na polovici sezone
mid_season_rankings = {
    'AC Milan': 1, 'Inter Milan': 2, 'Roma': 3,  'Juventus': 4,
'Atalanta': 5, 'Napoli': 6, 'Lazio': 7, 'Sassuolo': 8, 'Hellas
Verona': 9, 'Sampdoria': 10,
    'Benevento': 11, 'Bologna': 12, 'Udinese': 13,  'Spezia': 14,
'Fiorentina': 15, 'Genoa': 16, 'Cagliari': 17, 'Torino': 18,
'Parma': 19, 'Crotone': 20
}

# Simulacija za jednu preostalu utakmicu
def simulate_match(home_team, away_team):
    lambda_h = lambdas[home_team]['lambda_h']
    lambda_a = lambdas[away_team]['lambda_a']

    home_goals = np.random.poisson(lambda_h)
    away_goals = np.random.poisson(lambda_a)

    # Predikcija opcenitog rezulatata
    if home_goals > away_goals:
        return 3, 0  # Pobjeda domace ekipe
    elif home_goals < away_goals:
        return 0, 3  # Pobjeda gostujuce ekipe
    else:
        return 1, 1  # Neodluceni rezultat

# Prilagodba bodova na temelju rankinga na polovici sezone
def adjust_points(points):
    adjusted_points = {}
    for team, pts in points.items():
        rank_bias = (len(lambdas) - mid_season_rankings[team]) /
len(lambdas)
        adjusted_points[team] = pts + rank_bias * 5
    return adjusted_points

# Simulacija ostatka sezone
def simulate_season():
    points = {team: 0 for team in lambdas}

    for home_team in lambdas:
        for away_team in lambdas:
            if home_team != away_team:
                home_points, away_points = simulate_match(home_team,
away_team)
                points[home_team] += home_points
                points[away_team] += away_points

    points = adjust_points(points)
    return points

def monte_carlo_simulation(num_simulations=10000):
    results = {team: [] for team in lambdas}
```

```
    for _ in range(num_simulations):
        points = simulate_season()
        sorted_teams = sorted(points.keys(), key=lambda team:
points[team], reverse=True)

        for rank, team in enumerate(sorted_teams, start=1):
            results[team].append(rank)

    return results

def determine_final_rankings(results, num_simulations):
    final_rankings = {}

    for team in lambdas:
        rank_counts =
pd.Series(results[team]).value_counts().sort_index()
        probabilities = (rank_counts /
num_simulations).sort_values(ascending=False)
        top_3 = probabilities.head(3)
        normalized_probabilities = top_3 / top_3.sum()  # Normalize
to sum to 1
        final_rankings[team] =
list(zip(normalized_probabilities.index,
normalized_probabilities.values))

    return final_rankings

num_simulations = 20000
results = monte_carlo_simulation(num_simulations)
final_rankings = determine_final_rankings(results, num_simulations)

formatted_output = []
for team, rankings in final_rankings.items():
    formatted_output.append([
        team,
        f"{rankings[0][0]} ({rankings[0][1]:.3f})",
        f"{rankings[1][0]} ({rankings[1][1]:.3f})",
        f"{rankings[2][0]} ({rankings[2][1]:.3f})"
    ])

output_df = pd.DataFrame(formatted_output, columns=['Team', 'Prva
najvjerojatnija', 'Druga najvjerojatnija', 'Treca najvjerojatnija'])
output_df.set_index('Team', inplace=True)

print("Top 3 Najvjerojatnije završne pozicije:")
print(output_df)
```

Top 3 Najvjerojatnije završne pozicije:

Team	Prva najvjerojatnija	Druga najvjerojatnija	Treca najvjerojatnija
AC Milan	2 (0.343)	4 (0.329)	5 (0.327)
Inter Milan	1 (0.463)	2 (0.307)	3 (0.231)
Roma	4 (0.356)	2 (0.322)	3 (0.322)
Juventus	5 (0.357)	6 (0.323)	4 (0.320)
Atalanta	1 (0.420)	2 (0.292)	3 (0.288)
Napoli	3 (0.351)	2 (0.333)	4 (0.316)
Lazio	7 (0.348)	8 (0.329)	6 (0.322)
Sassuolo	8 (0.348)	9 (0.327)	6 (0.325)
Hellas Verona	12 (0.347)	15 (0.331)	13 (0.322)
Sampdoria	8 (0.351)	10 (0.349)	9 (0.300)
Benevento	15 (0.342)	13 (0.330)	16 (0.328)
Bologna	11 (0.338)	13 (0.338)	12 (0.323)
Udinese	17 (0.353)	19 (0.328)	16 (0.319)
Spezia	11 (0.338)	10 (0.332)	9 (0.329)
Fiorentina	19 (0.367)	18 (0.335)	17 (0.298)
Genoa	19 (0.384)	17 (0.317)	18 (0.299)
Cagliari	15 (0.346)	14 (0.330)	16 (0.324)
Torino	9 (0.348)	10 (0.345)	8 (0.307)
Parma	20 (0.642)	19 (0.226)	18 (0.133)
Crotone	18 (0.337)	17 (0.334)	16 (0.328)

Slika 9: Rezultat predviđanja finalne pozicije pomoću Poissonovog modela

Rezultati predviđanja sažeti su u Tablici 9. Pozlaćene ćelije su one ćelije koje sadrže finalni poziciju u jednoj od tri najvjerojatnije pozicije. Sveukupno nam je model predvidio točno 10 pozicija, od toga četiri puta prvu najvjerojatniju i šest puta treću najvjerojatniju. Taj rezultat od 50% je zadovoljavajući zbog činjenice da je talijanska liga poznata po tome da je teško predvidjeti rezultate.

Dvije ekipe koje bih izdvojio s najvećom razlikom u predviđenim pozicijama i ostvarenim pozicijama su Genoa i Torino. Za Genou je model predvidio da će ostati pri dnu tablice zbog niskih vrijednosti izračunatih parametara, no Genoa je završila s čak 6 mjesta razlike od najbliže predviđene pozicije.

Promatrajući Torino samo na temelju izračunatih parametara model je predvidio da će se Torino podignuti na ljestvici za minimalno 8 mjesta, no to nije bio slučaj. Torino se podigao samo za jedno mjesto gore u usporedbi s pozicijom na središnjem dijelu sezone, što je 7 mjesta razlike od najbliže predviđene pozicije.

Tablica 9: Prikaz top 3 vjerojatnosti završne pozicije na kraju sezone

Ekipa	Prva najvjerojatnija pozicija	Druga najvjerojatnija pozicija	Treća najvjerojatnija pozicija	Pozicija u polusezoni	Finalna pozicija
Inter Milan	1 (0.463)	2 (0.307)	3 (0.231)	2	1
AC Milan	2 (0.343)	4 (0.329)	5 (0.327)	1	2
Atalanta	1 (0.420)	2 (0.292)	3 (0.288)	5	3
Juventus	5 (0.357)	6 (0.323)	4 (0.320)	4	4
Napoli	3 (0.351)	2 (0.333)	4 (0.316)	6	5
Lazio	7 (0.348)	8 (0.329)	6 (0.322)	7	6
Roma	4 (0.356)	2 (0.322)	3 (0.322)	3	7
Sassuolo	8 (0.348)	9 (0.327)	6 (0.325)	8	8
Sampdoria	8 (0.351)	10 (0.349)	9 (0.300)	10	9
Hellas Verona	12 (0.347)	15 (0.331)	13 (0.322)	9	10
Genoa	19 (0.384)	17 (0.317)	18 (0.299)	16	11
Bologna	11 (0.338)	13 (0.338)	12 (0.323)	12	12
Fiorentina	19 (0.367)	18 (0.335)	17 (0.298)	15	13
Udinese	17 (0.353)	19 (0.328)	16 (0.319)	13	14
Spezia	11 (0.338)	10 (0.332)	9 (0.329)	14	15
Cagliari	15 (0.346)	14 (0.330)	16 (0.324)	17	16
Torino	9 (0.348)	10 (0.345)	8 (0.307)	18	17
Benevento	15 (0.342)	13 (0.330)	16 (0.328)	11	18
Crotone	18 (0.337)	17 (0.334)	16 (0.328)	20	19
Parma	20 (0.642)	19 (0.226)	18 (0.130)	19	20

5. Zaključak

U ovom radu cilj mi je bio istražiti i procjeniti učinkovitost dvaju matematičkih modela u predviđanju ishoda nogometnih utakmica. Glavni razlog žašto je fokus bio na prvu talijansku ligu Serie A je taj što je objektivno poznato da je ta liga jedna od najtežih za predviđanje, a subjektivno jer je najzanimljivija za gledati sa najsadržajnijim utakmicama.

Praktična implementacija Poissonovog i Binomnog modela provedena je s pomoću programskog jezika Python. Povijesni podaci iz prve polovice sezone Serie A korišteni su za procjenu parametara modela, uključujući prosječan broj golova, momčadski napad i snagu obrane, kao i prednost domaćeg terena. Zatim se testirala sposobnost modela u predviđanja vjerojatnih ishoda odabrane utakmice, uz njihovu izvedbu u usporedbi sa stvarnim povijesnim rezultatima.

Rezultati su pokazali da je Poissonov model općenito dobru predviđao ishode utakmica. Binomni model, iako je također učinkovit, pokazao je malo drugačije tendencije. Oba modela pokazala su prednosti i slabosti u različitim aspektima predviđanja ishoda utakmice, pri čemu niti jedan nije bio superiorniji od drugog.

Osim toga, modeli su korišteni za predviđanje konačnog poretka na kraju sezone. To je učinjeno simuliranjem preostalih utakmica sezone više puta, na temelju parametara izračunatih iz prve polovice sezone. Predviđeni poredak je zatim uspoređen sa stvarnim konačnim poretkom, nudeći uvid u snagu predviđanja ovih modela tijekom cijele sezone. Rezultati su bili više nego zadovoljavajući te iznad očekivanja.

Nepredvidivost svake nogometne utakmice je ono što donosi užitak gledanja, a značajan broj slučajnih čimbenika koji utječu na ishod, predstavlja se kao izazov za sve uključene, poput same momčadi, analitičara i kladionica. Upravo je zbog te nepredvidivosti potrebno razviti sofisticirane modele koji se sa velikom razinom točnosti mogu prilagođavati prirodi sporta.

Popis literature

[1] Munđar, D. (2023). Prognoza ishoda nogometne utakmice. *Poučak: časopis za metodiku i nastavu matematike*, *24*(94), 69-76.

[2] Munđar, D., & Šimić, D. (2016). Croatian First Football League: teams' performance in the championship. *Croatian Review of Economic, Business and Social Statistics*, *2*(1), 15-23.

[3] Maher, M. J. (1982). Modelling association football scores. *Statistica Neerlandica*, *36*(3), 109-118.

[4] Loukas, K., Karapiperis, D., Feretzakis, G., & Verykios, V. S. (2024). Predicting Football Match Results Using a Poisson Regression Model. *Applied Sciences*, *14*(16), 7230.

[5] Penn, M. J., & Donnelly, C. A. (2022). Analysis of a double Poisson model for predicting football results in Euro 2020. *Plos one*, *17*(5), e0268511.

[6] Domingues, J., Lopes, B., Mihaylova, P., & Georgieva, P. (2019, July). Incremental learning for football match outcomes prediction. In *Iberian Conference on Pattern Recognition and Image Analysis* (pp. 217-228). Cham: Springer International Publishing.

[7] Wikipedia (2024.) Serie A https://en.wikipedia.org/wiki/Serie_A

[8] Rezultati (2024.) Serie A 2020/2021
https://www.rezultati.com/nogomet/italija/serie-a-2020-2021/rezultati/

[9] SoccerSTATS (2024.) Italy – Serie A 2020/2021
https://www.soccerstats.com/results.asp?league=italy_2021&pmtype=resultsgrid

[10] Footballcritic (2024.) Serie A https://www.footballcritic.com/serie-a/history/8

[11] 5portal (19.07.2023.) Predviđanje sportskih događaja
https://5portal.hr/vijesti_detalj.php?id=43611

[12] Tax, N., & Joustra, Y. (2015). Predicting the Dutch football competition using public data: A machine learning approach. *Transactions on knowledge and data engineering*, *10*(10), 1-13.

[13] Bunker, R. P., & Thabtah, F. (2019). A machine learning framework for sport result prediction. *Applied computing and informatics*, *15*(1), 27-33.

[14] Wikipedija (2024). Poisson distribution
https://en.wikipedia.org/wiki/Poisson_distribution

[15] Forrest, D., & Simmons, R. (2000). Forecasting sport: the behaviour and performance of football tipsters. *International journal of Forecasting*, *16*(3), 317-331.

[16] SportsMole (2024) Previews https://www.sportsmole.co.uk/

[17] Coles, S., Bawa, J., Trenner, L., & Dorazio, P. (2001). *An introduction to statistical modeling of extreme values* (Vol. 208, p. 208). London: Springer.

[18] The DATA JOCKS (20.09.2022.) Teaching the binomial distribution with sports https://thedatajocks.com/teaching-the-binomial-distribution-with-sports/

[19] American soccer analysis (24.03.2014) Predicting Goals Scored using the Binomial Distribution https://www.americansocceranalysis.com/home/2014/03/24/predicting-goals-scored-using-the-binomial-distribution

[]

Popis slika

Slika 1: Isječak koda koji računa projećan broj golova po ekipi na nogometnoj utakmici 11

Slika 2: Isječak programskog koda, te rezultati spremnosti napada i obrane za ekipe 12

Slika 3: Isječak koda te rezultat vrijednosti prednosti domaćeg terena 14

Slika 4: Izlazni podaci predviđanja rezulatata utakmice Verona-Inter temeljem Poissonovog modela 16

Slika 5:Izlazni podaci predviđanja rezulatata utakmice Verona-Inter temeljem binomnog modela 20

Slika 6: Izlazni podaci implementacije za izračun općenitog ishoda utakmice Poissonovim modelom........ 22

Slika 7:Izlazni podaci implementacije za izračun općenitog ishoda utakmice binomnim modelom 23

Slika 8: Izlazni podaci implementacije izračuna Poissonovih parametara za predviđanje finalnog poretka........ 26

Slika 9: Rezultat predviđanja finalne pozicije pomoću Poissonovog modela........ 30

Popis tablica

Tablica: 1:Prikaz top 3 vjerojatnosti završne pozicije na kraju sezone 10

Tablica: 2:Parametri spremnosti napada i obrane za ekipe 13

Tablica: 3 Prikaz vjerojatnosti pojedinog rezulatata utakmice Verona-Inter temeljem Poissonovog modela 16

Tablica 4: Apsolutna i relativna frekvencija broja golova na utakmicama prvog dijela sezone 17

Tablica: 5: Prikaz vjerojatnosti pojedinog rezulatata utakmice Verona-Inter temeljem binomnog modela 21

Tablica 6: Usporedba rezultata ishoda utakmice promatranih modela........................ 23

Table 7: Stanje na tablici uz broj golova nakon prvog dijela sezone 24

Tablica 8: Poissonovi parametri za prvi dio sezone.. 26

Tablica 9: Prikaz top 3 vjerojatnosti završne pozicije na kraju sezone 31