Model za predviđanje ishoda nogometnih utakmica utemeljen na Elo sustavu rangiranja

Ćurko, Nino

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1267

MODEL ZA PREDVIĐANJE ISHODA NOGOMETNIH UTAKMICA UTEMELJEN NA ELO SUSTAVU RANGIRANJA

Nino Ćurko

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1267

MODEL ZA PREDVIĐANJE ISHODA NOGOMETNIH UTAKMICA UTEMELJEN NA ELO SUSTAVU RANGIRANJA

Nino Ćurko

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1267

Pristupnik:	**Nino Ćurko (0036541403)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	doc. dr. sc. Stjepan Šebek

Zadatak: **Model za predviđanje ishoda nogometnih utakmica utemeljen na Elo sustavu rangiranja**

Opis zadatka:

U sklopu ovog zadatka potrebno je razviti model za predviđanje ishoda nogometnih utakmica. Model na kojem se treba bazirati predviđanje je model Poissonove regresije, a za jednu od glavnih varijabli poticaja u modelu Poissonove regresije potrebno je uzeti Elo rang pojedinih ekipa. Razvijeni model potrebno je testirati na podacima vezanima uz Prvu Hrvatsku nogometnu ligu. Prvi dio zadatka je prikupiti povijesne podatke i izračunati Elo rang pojedinih ekipa u Prvoj HNL, a zatim na temelju tih povijesnih rezultata istrenirati model Poissonove regresije. Razvijeni model potrebno je ispitati na rezultatima nadolazećih utakmica, a zatim je nužno obaviti i podrobnu statističku analizu rezultata.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod

U današnje moderno doba nogomet se smatra najpopularnijim sportom na planetu. Nogomet se igra na sljedeći način: dvije ekipe sastavljene od jedanaest igrača igraju jedna protiv druge devedeset minuta i pokušavaju postići što više golova. To je sport s malo postignutih pogodaka u kojem su moguća tri ishoda (gledajući iz perspektive određene ekipe koja igra utakmicu): pobjeda, neriješen rezultat i poraz. Mnogi čimbenici mogu utjecati na konačni rezultat, ali ova se teza koncentrira na Elo sustav rangiranja.

Elo sustav rangiranja je metoda za izračunavanje relativne razine vještine u igrama za dva igrača. Razvijen od strane Arpada Ela, ovaj sustav se prvobitno koristio za rangiranje šahista, ali je od tada proširen na mnoge druge sportske i nesportske kontekste. Bodovi koje jedna strana dobije na temelju ishoda utakmice, druga strana izgubi u istoj količini. U nogometu, Elo sustav može pomoći u procjeni snage ekipa na temelju njihovih rezultata, uzimajući u obzir i važnost utakmice i snagu protivnika.

Ova teza planira analizirati takvu implementaciju u nogometu, posebno u HNL-u (Hrvatska Nogometna Liga), kako bi se vidio utjecaj koji može imati na daljnje predikcije. HNL predstavlja idealan okvir za ovu analizu zbog svoje jedinstvene dinamike i konkurentnosti unutar lige.

Postavljanje takvog modela još uvijek nam predstavlja poteškoće jer mnogi neobični čimbenici igraju ulogu u nogometnim utakmicama. Čak ni čista sreća ne smije se zanemariti. Faktori poput vremenskih uvjeta, ozljeda ključnih igrača, taktičkih odluka trenera i drugih vanjskih utjecaja mogu značajno utjecati na ishod utakmica. Stoga, predviđanje rezultata nogometnih utakmica zahtijeva sofisticirane modele koji mogu uzeti u obzir ove varijable.

Najbolje trenutno rješenje modela koje je lakše razumjeti pruža Poissonova regresija.

Poissonova regresija je vrsta generalizirane linearne regresije koja je prikladna za modeliranje broja događaja koji se događaju u fiksnom vremenskom periodu. U kontekstu nogometa, ona se koristi za modeliranje broja postignutih golova s obje strane. Poissonova distribucija je prikladna zbog prirode rezultata u nogometu, gdje su golovi rijetki i diskretni događaji.

Poissonova regresija igrat će veliku ulogu u pozadini za treniranje podataka za buduće predikcije zbog svoje značajke izračunavanja određenog broja postignutih golova s obje strane. Detaljniji tehnički detalji implementacije, naravno, bit će prikazani u daljnjim poglavljima koja pružaju više tehničkih činjenica. Ono na čemu će se naša regresija temeljiti jest nezavisnot golova zabijenih od obje ekipe kao u radu [1].

Na kraju analize regresije, ovaj model će se usporediti s primitivnijim modelom koji koristi čiste formule bez ikakvih složenih izračuna formiranih regresijom. Na taj način, cilj je procijeniti koliko sofisticirani modeli poput Poissonove regresije mogu unaprijediti točnost predikcija u usporedbi s jednostavnijim pristupima.

Cilj ove teze je pružiti uvid u učinkovitost Elo sustava rangiranja kombiniranog s Poissonovom regresijom u predviđanju rezultata nogometnih utakmica. Rezultati istraživanja mogu imati široku primjenu u sportskom menadžmentu, klađenju, kao i u analitičkim timovima unutar nogometnih klubova.

2. Izrada modela

2.0.1. Prikupljanje podataka

Jedan od prvih koraka u stvaranju željenog modela bio je prikupljanje potrebnih podataka koji će se koristiti za treniranje. Najlakši način bio je korištenje Python tehnika za web scraping. Web scraping je proces automatskog izdvajanja informacija s web stranica. To uključuje dohvaćanje web stranica, parsiranje sadržaja i izdvajanje specifičnih podataka sa stranica. U ovom slučaju, alat koji se koristi zove se BeautifulSoup. Mnoge stranice ne dopuštaju izdvajanje njihovih podataka, pa parsiranje može biti blokirano. Stranica s koje su podaci za ovaj model izdvojeni je SuperSport (https://hnl.hr/). Scrapane sezone koje su korištene za treniranje modela su: 2021/2022, 2022/2023 i 2023/2024 (za zadnju sezonu koriste se samo utakmice iz 2023. godine, ostatak je za testiranje modela). Podaci se spremaju lokalno u .csv datoteku radi lakših modifikacija u Microsoft Excelu kada je potrebno. Dio koda za scraping weba može se vidjeti u 2..1, gdje Pythonov BeautifulSoup dolazi do izražaja.

Listing 2..1: Dio koda za web scraping

```
from bs4 import BeautifulSoup
import requests
import csv

url = 'https://hnl.hr/supersport-hnl/raspored-i-rezultati/'
data = requests.get(url).text

soup = BeautifulSoup(data, 'html.parser')
```

```
table = soup.find_all('table')

td = soup.find_all('td')
dateList = []
hostList = []
hostScoreList = []
guestScoreList = []
guestList = []

for i in range(0, 1433, 8):
   dateList.append(str(td[i].text).replace(".", "/"))
   hostList.append(str(td[i + 1].text))
   hostScoreList.append(str(td[i + 2].text))
   guestScoreList.append(str(td[i + 4].text))
   guestList.append(str(td[i + 5].text))




for i in range(0, len(dateList)):
   dateList[i] = str(dateList[i]).split(" ")[0]

header = ['Date', 'Host', 'HostScore', 'GuestScore', 'Guest']
```

Prikupljeni podaci bit će stavljeni u .csv datoteku s kolonama "Date", što je datum kada je utakmica odigrana, "Host" je ime tima koji igra kod kuće, "Guest" je ime tima koji igra u gostima, "HostScore" su golovi koje je postigao domaći tim, a "GuestScore" su golovi koje je postigao protivnički tim. Ova .csv datoteka s nazivima kolona (i nekoliko dodatnih kolona) bit će korištena u daljnjim sekcijama.

2.0.2. Definiranje Elo sustava

Elo bodovi za svaki tim su proizvoljno određeni za početak sezone 2021/2022. Svaki klub iz lige počet će s određenim brojem bodova. Budući da nogomet, gledan kroz oči tima, ima tri moguća ishoda: pobjeda, neriješeno i poraz, bodovi koji se dodaju ili oduzimaju moraju se dodijeliti svakom timu u skladu s ishodom utakmice kako bi se održala ravnoteža za njihove rangove.

Za izračunavanje koliko će se bodova dodati ili oduzeti od trenutnih bodova, uzimamo razliku bodova između kluba na koji se referiramo i kluba protiv kojeg igra. Na primjer, ako gledamo iz perspektive Dinama (200 bodova) i igraju protiv Rijeke (190 bodova), razlika koja će se koristiti bit će 10 bodova (-10 ako bismo gledali iz perspektive Rijeke).

Razlika koju smo dobili ranije sada će se koristiti u sljedećoj formuli :

$$E = \frac{1}{1 + 10^{\frac{-R_{\text{Home}} + R_{\text{Away}}}{400}}}$$

tako da je broj skaliran na odgovarajući način, ali dobivamo broj koji je povezan s vjerojatnošću pobjede. Rezultat koji dobijemo kao **E** mora se oduzeti od rezultata utakmice, a zatim se rezultat mora pomnožiti s nekim koeficijentom koji biramo. U ovom slučaju taj broj će biti

$$K = 20$$

Formula će izgledati ovako:

$$\Delta Elo1x2 = K\,(R - E)$$

gdje **R** može biti jednak 1 ako je klub pobijedio u utakmici, 0.5 ako je neriješeno bio ishod ili čista 0 ako su izgubili utakmicu. Nakon što se Δ **Elo1x2** izračuna, samo se mora dodati Elo bodovima tima s kojima je tim ušao u utakmicu.
Kao što je opisano, Elo nam pruža jednostavnu metodu za izračunavanje snage svakog tima. Što je veći, tim je bolji i vjerojatnije će pobijediti protiv niže rangiranog protivnika. Naravno, na ovaj način, ako je Elo razlika veća i ishod ide u korist niže rangiranog tima, bodovi koji se dodaju/oduzimaju bit će veći. Ulomak koda u 2..2 je jednostavna implementacija takvih izračuna u Pythonu.

Listing 2..2: Kalklurliranje Elo bodova u Pythonu

```
for i in range(len(home)):
   if not math.isnan(homeElo[i]):
      eloDiff = float(homeElo[i]) - float(guestElo[i])
      e = 1 / ((pow(10, -eloDiff/400) + 1))
      if homeScore[i] > awayScore[i]:
         clubElos[home[i]] = homeElo[i] + (1-e)*20
      elif homeScore[i] == awayScore[i]:
         clubElos[home[i]] = homeElo[i] + (0.5-e)*20
      else:
         clubElos[home[i]] = homeElo[i] + (0-e)*20

   if not math.isnan(guestElo[i]):
      eloDiff = guestElo[i] - homeElo[i]
      e = 1 / ((pow(10, -eloDiff/400) + 1))
      if awayScore[i] > homeScore[i]:
         clubElos[away[i]] = guestElo[i] + (1-e)*20
      elif homeScore[i] == awayScore[i]:
         clubElos[away[i]] = guestElo[i] + (0.5-e)*20
      else:
         clubElos[away[i]] = guestElo[i] + (0-e)*20

for i in range(len(home)):
   if math.isnan(homeElo[i]):
      homeElo[i] = clubElos[home[i]]
      guestElo[i] = clubElos[away[i]]

      eloDiff = homeElo[i] - guestElo[i]
      e = 1 / ((pow(10, -eloDiff/400) + 1))
      if homeScore[i] > awayScore[i]:
```

```
        clubElos[home[i]] = homeElo[i] + (1-e)*20
    elif homeScore[i] == awayScore[i]:
        clubElos[home[i]] = homeElo[i] + (0.5-e)*20
    else:
        clubElos[home[i]] = homeElo[i] + (0-e)*20

    eloDiff = guestElo[i] - homeElo[i]
    e = 1 / ((pow(10, -eloDiff/400) + 1))
    if awayScore[i] > homeScore[i]:
        clubElos[away[i]] = guestElo[i] + (1-e)*20
    elif homeScore[i] == awayScore[i]:
        clubElos[away[i]] = guestElo[i] + (0.5-e)*20
    else:
        clubElos[away[i]] = guestElo[i] + (0-e)*20
```

gdje podatkovni okvir "df" predstavlja .csv datoteku sa svojim kolonama koje su opisane u 2.0.1. Dodane kolone koje se mogu vidjeti u kodu su "EloHost" i "EloGuest", koje predstavljaju Elo bodove domaćeg tima i gostujućeg tima. Prva petlja u kodu služi za popunjavanje rječnika "clubElos" gdje je ključ ime kluba, a vrijednost su Elo bodovi nakon ishoda utakmice. Druga petlja iterira kroz svaku utakmicu iz zadanih podataka i izračunava nove Elo bodove koje zatim stavlja u liste "homeElo" i "guestElo". Te dvije liste koriste se za stvaranje privremenih .csv datoteka čije se kolone, kada su stvorene, ručno kopiraju i lijepe u glavnu .csv datoteku. gdje su "helper1" i "helper2" liste listi gdje je svaka lista unutar samo jedna Elo vrijednost. To se radi kako bi bilo lakše zapisati ih kao stupac koji se kasnije može kopirati i zalijepiti.

U našem modelu, sve početne vrijednosti Elo bodova bit će preuzete sa stranice Football Club Elo Ratings[2] koja daje uvid i u nadolazeće utakmice, ali i ima API koji nam omogućuje da za određeni datum preuzmemo Elo rang određenog kluba

2.0.3. Realizacija primitivnog modela

U sklopu ovog rada, a da bi se kasnije moglo naš model usporediti s drugim modelom na istom setu podataka, napravljen je jedan malo primitivniji (jednostavniji model). Naime, ovaj model koristi isključivo Elo bodove domaćeg kluba s kojima klub ulazi u igru i bodove protivničkog. S pomoću ovih bodova idućim formulama računa postotak pobjede domaćeg tima, postotak neriješenog ishoda, kao i postotak pobjede gostujuće strane:

$$a = 10^{\frac{\text{EloHome}-\text{EloAway}}{400}},$$

gdje a ima svrhu brojnika za potrebnu formulu, dok iduće varijable ($b1$ i $b2$):

$$b1 = 10^{\frac{\text{EloAway}-\text{EloHome}}{400}}$$
$$b2 = 10^{\frac{\text{EloHome}-\text{EloAway}}{400}}$$

će imati svrhu nazivnika. S tim rečenim, naša formula za računanje postotka pobjede domaćeg tima ima idući oblik:

$$w = \frac{a}{b1 + 1 + b2} \times 100$$

Kako bi se dobio postotak za pobjedu protivničke strane, potrebno je zamijeniti predznake za "EloHome" i "EloAway" u formuli za a, te dobiveni broj ubaciti u w. Intuitivno, postotak izjednačenog rezultata kao ishoda se dobije da se od broja 100 oduzme šansa pobjede domaćeg tima i šansa pobjede gostujućeg tima.

2.0.4. Regresija i realizacija sofisticiranijeg modela

Poissonova regresija - općenito

Budući da je vrlo teško predvidjeti ishod nogometne utakmice jer različiti faktori igraju ulogu, nijedan opći model ne može se koristiti sa 100% točnosti. Model na koji smo se oslonili temelji se na Poissonovoj regresiji.

Poissonova regresija je vrsta generaliziranog linearnog modela (GLM) koji se koristi za modeliranje podataka o brojanju[3]. Pretpostavlja da zavisna varijabla Y (u našem slučaju, golovi koje postigne svaki tim igrajući jedan protiv drugog) slijedi Poissonovu distribuciju. Ovaj model je prikladan kada podaci predstavljaju brojeve ili broj pojavljivanja nekog događaja unutar fiksnog intervala vremena ili prostora.
Poissonova distribucija je dana sa idućom formulom:

$$P(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

gdje je λ i stopa parametra (sredina i varijanca distribucije) i y je nenegativan cijeli broj (tj., $y = 0, 1, 2, ...$). U Poissonovoj regresiji modeliramo logaritam očekivane vrijednosti zavisne varijable Y kao linearnu kombinaciju prediktorskih varijabli. Model se može napisati kao:

$$\log(\mathbb{E}[Y \mid \mathbf{X}]) = \mathbf{X}\boldsymbol{\beta}$$

gdje:

- $\mathbb{E}[Y \mid \mathbf{X}]$ je očekivana vrijednost Y s obzirom na prediktorske varijable $\mathbf{X}$.
- $\mathbf{X}$ je vektor prediktorskih varijabli.
- $\boldsymbol{\beta}$ je vektor koeficijenata.

Jednako tako, možemo izravno izraziti očekivanu vrijednost kao:

$$\mathbb{E}[Y \mid \mathbf{X}] = \exp(\mathbf{X}\boldsymbol{\beta})$$

Log-likelihoood funkcija za Poissonov regresijski model izvedena je iz Poissonove funkcije mase vjerojatnosti. Za dani skup podataka s n opažanja, log-likelihoood je dan s:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i \log(\lambda_i) - \lambda_i - \log(y_i!))$$

gdje je $\lambda_i = \exp(\mathbf{X}_i\boldsymbol{\beta})$. Koeficijenti $\boldsymbol{\beta}$ procjenjuju se maksimiziranjem log-likelihoood funkcije. To se obično radi korištenjem iterativnih algoritama kao što su Newton-Raphson ili Fisher Scoring.
Koeficijenti $\boldsymbol{\beta}$ u Poissonovom regresijskom modelu imaju multiplikativni učinak na zavisnu varijablu. Konkretno, za promjenu prediktora X_j, za jednu jedinicu, očekivana vrijednost zavisne varijable Y mijenja se za faktor $\exp(\beta_j)$.

Ako imamo jednu prediktorsku varijablu X, Poissonov regresijski model može se napisati kao:

$$\log(\mathbb{E}[Y \mid X]) = \beta_0 + \beta_1 X$$

ili jednako tako:

$$\mathbb{E}[Y \mid X] = \exp(\beta_0 + \beta_1 X)$$

U ovom slučaju, β_0 je presretanje i β_1 je koeficijent za prediktor X.

Prilagođavanje regresije

U našem konkretnom slučaju, imat ćemo nekoliko prediktora. Prvi prediktor bit će varijabla **Home** čija vrijednost može biti nula ako tim igra kao gost ili jedan ako tim igra kao domaćin. Sljedeći prediktori će, naravno, biti **EloHost** i **EloGuest** koji će predstavljati Elo bodove s kojima timovi ulaze u utakmicu ("EloHost" je reprezentacija Elo bodova za domaćina, "EloGuest" reprezntira Elo bodove za goste). Sljedeća dva prediktora odabrana su uzimajući u obzir zadnjih nekoliko utakmica koje je svaki tim odigrao; konkretno zbroj golova postignutih u zadnjih pet utakmica, što se dinamički mijenja nakon

svake utakmice. Iduće dvije varijable koje dolaze u obzir su tržišne vrijednosti oba tima. Vrijednosti su preuzete s web stranice https://www.transfermarkt.com/ i ostaju iste tijekom cijele sezone. Uz dane prediktore, dodatno je odlučeno da se stavi i "HomePts" te "AwayPts" koji su tu da daju točan broj bodova koje tim ima na tablici prije ulaska u konkretni okršaj. Bodovi se u ligi mijenjaju na idući način: za pobjedu ekipa dobiva 3 boda, za izgubljenu utakmicu ostaje na prijašnjim bodovima dok neriješenim ishodom dobiva 1 bod. Zadnja, nama najvažnija vrijednost čije ishode ovaj model i pokušava predvidjeti jest Y, koja će biti ukupni broj golova koje je svaki tim postigao. Uz sve to rečeno, naš regresijski model će izgledati ovako:

$$\begin{aligned}\mathbb{E}[Y \mid X] = \exp(&\beta_0 + \beta_1 \text{Home} + \beta_2 \text{EloHost} + \beta_3 \text{EloGuest} \\ &+ \beta_4 \text{HomeInLast5} + \beta_5 \text{AwayInLast5} \\ &+ \beta_6 \text{HomeVal} + \beta_7 \text{GuestVal} + \beta_8 \text{HomePts} + \beta_9 \text{AwayPts})\end{aligned}$$

ili kao u ulomku koda iz Pythona:

Listing 2..3: Postavljanje regresije

```
def fit_model(
    model_data: pd.DataFrame,
) -> sm.regression.linear_model.RegressionResultsWrapper:
    return smf.glm(
        formula="Total~Home+EloHost+EloGuest+HomeVal+AwayVal+
                HomeIn5+AwayIn5+HomePts+AwayPts",
        data=model_data,
        family=sm.families.Poisson(),
    ).fit()

trainingModel = fit_model(pd.read_csv("trainingModel.csv"))
```

gdje su svi potrebni paketi da bi se postavila regresija i kasnije korektno izračunali svi potrebni podaci na slici 2.1.:

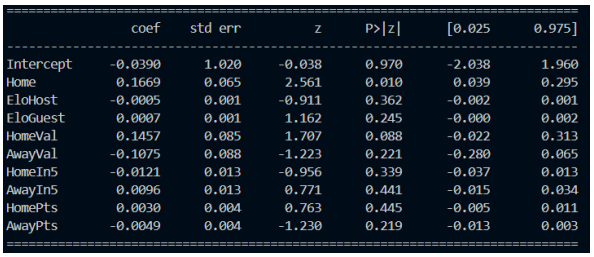
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
import pandas as pd
from pandas import *
import numpy as np
from scipy.stats import poisson

data = read_csv("trainingModelv2.csv")
```

Slika 2.1. Potrebne knjižnice

3. Rezultati i rasprava

3.0.1. Statistički osvrt na trenirani model

Kada smo uspješno prilagodili model za treniranje i provukli ga kroz kod 2..3, dobili smo sažetak tražene regresije koji je prikazan na slici 3.1.:



	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.0390	1.020	-0.038	0.970	-2.038	1.960
Home	0.1669	0.065	2.561	0.010	0.039	0.295
EloHost	-0.0005	0.001	-0.911	0.362	-0.002	0.001
EloGuest	0.0007	0.001	1.162	0.245	-0.000	0.002
HomeVal	0.1457	0.085	1.707	0.088	-0.022	0.313
AwayVal	-0.1075	0.088	-1.223	0.221	-0.280	0.065
HomeIn5	-0.0121	0.013	-0.956	0.339	-0.037	0.013
AwayIn5	0.0096	0.013	0.771	0.441	-0.015	0.034
HomePts	0.0030	0.004	0.763	0.445	-0.005	0.011
AwayPts	-0.0049	0.004	-1.230	0.219	-0.013	0.003

Slika 3.1. Sažetak modela

Rezultate statistički tumačimo na idući način:

- **Intercept**: Procijenjeni koeficijent za intercept je -0.0390. Ova vrijednost nije statistički značajna ($p = 0.970$), što znači da intercept nije značajno različit od nule. Interval pouzdanosti od 95% kreće se od -2.038 do 1.960.
- **Home**: Koeficijent za varijablu **Home** je 0.1669, što je statistički značajno ($p = 0.010$). Ovo sugerira da igranje kod kuće dovodi do povećanja zavisne varijable. Interval pouzdanosti za ovaj učinak je [0.039, 0.295].
- **EloHost**: Koeficijent za **EloHost** je -0.0005, što nije statistički značajno ($p = 0.362$). Ovo ukazuje da Elo rating domaćeg tima nema značajan utjecaj na zavisnu varijablu. Interval pouzdanosti je [-0.002, 0.001].
- **EloGuest**: Koeficijent za **EloGuest** je 0.0007, što također nije statistički značajno

($p = 0.245$). Ovo implicira da Elo rating gostujućeg tima nema značajan utjecaj na zavisnu varijablu. Interval pouzdanosti je [-0.000, 0.002].

- **HomeVal**: Koeficijent za **HomeVal** je 0.1457 s p-vrijednošću od 0.088. Ovaj rezultat nije statistički značajan na razini 0.05, ali sugerira potencijalni pozitivan učinak. Interval pouzdanosti je [-0.022, 0.313].

- **AwayVal**: Koeficijent za **AwayVal** je -0.1075, što nije statistički značajno ($p = 0.221$). Ovo ukazuje na to da vrijednost gostujućeg tima nema značajan utjecaj na zavisnu varijablu. Interval pouzdanosti je [-0.280, 0.065].

- **HomeIn5**: Koeficijent za **HomeIn5** je -0.0121, što nije statistički značajno ($p = 0.339$). Ovo sugerira da broj domaćih utakmica u posljednjih 5 nema značajan utjecaj na zavisnu varijablu. Interval pouzdanosti je [-0.037, 0.013].

- **AwayIn5**: Koeficijent za **AwayIn5** je 0.0096, što nije statistički značajno ($p = 0.441$). Ovo ukazuje na to da broj gostujućih utakmica u posljednjih 5 nema značajan utjecaj na zavisnu varijablu. Interval pouzdanosti je [-0.015, 0.034].

- **HomePts**: Koeficijent za **HomePts** je 0.0030, što nije statistički značajno ($p = 0.445$). Ovo implicira da bodovi domaćeg tima nemaju značajan utjecaj na zavisnu varijablu. Interval pouzdanosti je [-0.005, 0.011].

- **AwayPts**: Koeficijent za **AwayPts** je -0.0049, što nije statistički značajno ($p = 0.219$). Ovo sugerira da bodovi gostujućeg tima nemaju značajan utjecaj na zavisnu varijablu. Interval pouzdanosti je [-0.013, 0.003].

Komentar sažetka

Poissonov regresijski model pokazuje da među prediktorima, samo varijabla Home ima statistički značajan (Statistička značajnost ocjenjuje događa li se neki rezultat slučajno. Kad je rezultat statistički značajan, to znači da nije vjerojatno da se rezultat događa slučajno ili radi slučajne fluktuacije [4].) pozitivan učinak na zavisnu varijablu. Ostali prediktori, uključujući EloHost, EloGuest, HomeVal, AwayVal, HomeIn5, AwayIn5, HomePts i AwayPts, ne pokazuju značajne učinke na razini značajnosti od 0.05. Intervali pouzdanosti za ove koeficijente uključuju nulu, što dodatno sugerira nedostatak značajnog utjecaja.

3.0.2. Testiranje modela

Idući korak analize je testiranje treniranog modela. Kao podaci koji ulaze u testiranje uzete su sve utakmice HNL-a nakon zimske stanke, dakle od drugog mjeseca 2024. godine pa sve do kraja natjecanja (što je krajem petog mjeseca iste godine). Podaci za testiranje se također, kao i podaci za treniranje modela, nalaze u .csv datoteci radi lakšeg učitavanja podataka.

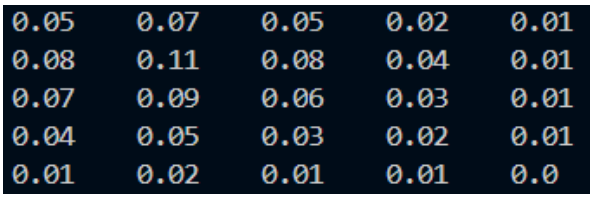
Način na koji predviđamo ishod svake pojedine utakmice prikazan je u kodu 3..1:

Listing 3..1: Predviđanje ishoda

```
for i in range(len(dates)):
    home_goals = trainingModel.predict(
        pd.DataFrame(
            data={"EloHost": eloHost[i], "EloGuest":eloGuest[i], "Home"
   : 1, "HomeVal":homeVal[i], "AwayVal":awayVal[i],
                  "HomeIn5":homeIn3[i], "AwayIn5":awayIn3[i], "HomePts"
   :homepts[i], "AwayPts":awaypts[i]},
            index=[1],
        )
    ).values[0]
```

```
away_goals = trainingModel.predict(
    pd.DataFrame(
        data={"EloHost": eloHost[i], "EloGuest":eloGuest[i], "Home"
: 0, "HomeVal":homeVal[i], "AwayVal":awayVal[i],
              "HomeIn5":homeIn3[i], "AwayIn5":awayIn3[i],"HomePts":
homepts[i], "AwayPts":awaypts[i]},
        index=[1],
    )
).values[0]

home_goals = round(home_goals, 2)
away_goals = round(away_goals, 2)
max_goals = 5

probabilities = [
    [poisson.pmf(i, team_avg) for i in range(0, max_goals)]
    for team_avg in [home_goals, away_goals]
]

matrix = np.outer(np.array(probabilities[0]), np.array(
probabilities[1]))
```

gdje kao parametre koristimo specifične vrijednosti stupaca za svaku utakmicu, a za domaću ekipu kao "Home" stavljamo vrijednost 1, dok za goste istu varijablu postavljamo na 0. S tim dobivamo predviđenu vrijednost broja golova za obje ekipe, tj. naše vrijednosti λ za obje ekipe što u principu predstavlja okviran broj golova koji se od obje ekipe očekuje da zabiju. Kao što je rečeno u poglavlju 1., nezavisnost golova nam ovdje omogućuje da izgeneriramo matricu konkretnog rezultata. Varijabla "max_goals" je postavljena na 5 što znači da ćemo u obzir uzeti vjerojatnosti za sve kombinacije rezultata od 0-0 do 4-4. Upravo su ti postotci zapisani u varijabli "matrix" koja je kod nas 2D polje. Njen ispis za utakmicu "Dinamo - Rudeš" prikazan je na slici 3.2.:



Slika 3.2. Matrica rezultata (Dinamo - Rudeš)

Matricu tumačimo na idući način: stupac nam označava gostujuću ekipu, a redak domaću ekipu. Da bi dobili postotak za određeni ishod broj iz određene ćelije matrice množimo sa brojem 100. Ovim putem tako možemo vidjeti da je vjerojatnost za 0-0 (prvi stupac i prvi redak matrice) 5%. Najveća šansa je za 1-1 (čak 11%), dok za izniman slučaj kao 4-4 je šansa kada se zaokruži na dvije decimale čista nula.
Iduće je pitanje kako dobiti vjerojatnost pobjede domaćina/gosta i vjerojatnost neriješenog rezultata. Iz dane matrice se to može iščitati na način da zbrojimo ćelije gornjeg trokuta matrice (dio iznad glavne dijagonale) te nam taj broj (pomnožen naravno sa 100) govori vjerojatnost pobjede gostujuće ekipe. Na analogan način možemo dobiti i vjerojatnost za pobjedu domaće ekipe na način da zbrojimo ćelije donjeg trokuta matrice i dobiveni zbroj pomnožimo sa 100. Preostaje nam vjerojatnost neriješenog susreta, a nju dobijemo kad prosumiramo elemente glavne dijagonale. Ovim putem dobivamo iduće vjerojatnosti:

- **Pobjeda domaćina (Dinamo)**: 41%
- **Neriješen ishod**: 24%
- **Pobjeda gosta (Rudeš)**: 31%

3.0.3. Usporedba dvaju modela

Nakon provedene statističke analize za obje vrste modela, smislili smo tri načina za provedbu analize točnosti. Za **prvi način**, odabrali smo klasičan kriterij točnosti gdje smo postotak točnosti očitali kao omjer točno predviđenih ishoda. Kao "točno" predviđen ishod uzimali smo onaj s najvećim postotkom. Na primjer, ako je model za pobjedu domaćina predvidio 45%, za neriješen ishod 25%, a za pobjedu gostujuće momčadi 30%, utakmica bi se smatrala točno predviđenom samo ako se stvarno dogodio pozitivan ishod za domaću momčad. Sveukupna točnost izračunavala se kao omjer točno predviđenih utakmica i ukupnog broja utakmica.

Za **drugi način** primijetili smo da oba modela kaskaju u točnom predviđanju neriješenih ishoda (skoro nikad ne predvide neriješen rezultat kao najvjerojatniji). U tom slučaju, odredili smo donju granicu vjerojatnosti za neriješeni rezultat. Ako bi model predvidio vjerojatnost veću od te granice, a u stvarnosti se dogodio neriješen ishod, ta bi utakmica pridonijela točno predviđenima. U slučaju da jedan od dva uvjeta nije ispunjen, model bi gledao predikciju kojoj pridjeljuje najveći postotak.

Konačno, za **treći način** odlučili smo uzeti nasumičnu vrijednost jedne od tri predviđene za svaku utakmicu. Ako je nasumično odabrana vrijednost bila korespondentna stvarnom ishodu, ta bi utakmica se ubrajala u točno predviđene. Za ovaj način odlučili smo provesti uzorkovanje s više pokretanja modela kako bismo uočili raspon, srednju vrijednost i odstupanja koje ovaj način pruža.

Primitivni model

Prvi način

Pokretanjem ovog modela i bilježenjem rezultata za prvi način predviđanja, postigli smo impresivnu točnost od **68.6%**. Ovo ukazuje na izvanrednu sposobnost modela da s visokim postotkom točnosti predvidi stvarni ishod, dajući najveću vrijednost upravo onom ishodu koji se i dogodi.

Drugi način

Budući da model rijetko pridijeli najveću vrijednost neriješenom ishodu, morali smo ga prilagoditi kako bismo vidjeli na kojoj vrijednosti točnost predviđanja raste. Postavili smo prag na 32%, te ako je predviđena šansa za neriješen ishod veća od tog praga i stvarno se dogodi, smatrali smo ga točno predviđenim. Na ovaj način, točnost modela porasla je na 69.77%. Daljnjim spuštanjem granice primijetili smo porast točnosti, no ispod 19% točnost je ostala nepromijenjena. Ono što smo još uočili prilagodbom ove granice jest da je za donju granicu 19% ujedno i uvelike porasla točnost na iznos od čak 79.07%, a sam broj utakmica gdje se predviđa neriješen ishod sa ovom donjom granicom iznosi 12 utakmica od ukupnih 14 (dok je npr. kod donje granice postavljene na 33% taj broj iznosio samo 4). Uz to, točnost predviđanja ostalih dvaju ishoda iznosi 65.12%. Ono što ovdje ne smijemo zanemariti jest broj krivo predviđenih ishoda za spuštenu donju granicu jer, naime, svaki put kada model pridijeli vjerojatnost neriješenog ishoda veću od 19%, a taj se ishod u stvarnosti nije realizirao te uz to još nije korektno predvidio ni ostala dva moguća ishoda, dobivamo brojku od 16 krivo predviđenih utakmica (od ukupno 86 na kojima testiramo), dok ta brojka za donju granicu od 33% iznosi samo 3. To je nezanemariva cijena koju ovim putem moramo platiti.

Treći način

Uzorkovanjem slučajno odabranih vrijednosti dobili smo donju granicu od 51.16%, gornju granicu od 65.12%, te srednju vrijednost uzorka od 57.27%. Ovi rezultati pokazuju da ovaj način predviđanja osigurava konzistentne rezultate s točnošću značajno većom od nasumičnog odabira jednog od tri ishoda, te unatoč svojoj slučajnosti, pruža pouzdane i stabilne rezultate.

Regresijski model

Prvi način

Ovaj sofisticiraniji model, primijenjen na naš "klasičan" način predviđanja, daje nižu točnost u usporedbi s prethodnim modelom. Točnost predviđanja iznosi 42.02%. Iako je ovo niže od očekivanog, rezultat je ipak bolji od nasumičnog odabira jednog od tri ishoda, što

ga čini relevantnim i vrijednim daljnjeg istraživanja.

Drugi način

Za ovaj način predviđanja, postavili smo uvjet da model predvidi vjerojatnost neriješenog ishoda veću od 29%, pod uvjetom da se takav ishod stvarno dogodi. U tom slučaju, model postiže točnost od 44.19%. Daljnjim snižavanjem praga primijetili smo porast točnosti sve dok nismo došli do donje granice od 24% (gdje dobivamo čak 13 točno predviđenih od ukupno 14 neriješenih ishoda), kod koje točnost stagnira uzastopnim njenim smanjivanjem dok nam točnost predviđanja ostalih ishoda u ovom slučaju iznosi i dalje 43.02%. Ovo ukazuje na to da ovaj model predviđa neriješene ishode još slabije od prethodnog modela.

Treći način

Uzorkovanjem ovim načinom dobili smo niz vrijednosti točno predviđenih utakmica. Donja granica iznosila je 33.72%, dok je gornja granica bila 52.33%. Ovi rezultati ukazuju na određena odstupanja. Srednja vrijednost uzorka iznosila je 42.73%, što nam govori da je prethodni model trenutno pouzdaniji u predviđanju nogometnih utakmica.

4. Zaključak i budući rad

Konačno, nakon raznih isprobavanja i igranja s našim modelima, možemo doći do nekih zaključaka. Konkretno rečeno, Poissonova regresija nam daje solidne predikcije s obzirom na to da je njezina točnost značajno veća od nasumičnog pogađanja jednog od tri moguća ishoda nogometne utakmice. Nažalost, područje koje uvelike zapostavlja jest predikcija neriješenog ishoda utakmice.

Elo sustav nije mogao doći do izražaja zbog velike prednosti koju Poissonova regresija daje domaćim ekipama u odnosu na gostujuće. Statistička analiza pokazala je da još neke varijable mogu igrati važnu ulogu. Uz sofisticiranije korekcije i načine rangiranja, kao i pridjeljivanja bodova u Elo sustavu, u budućnosti se nadamo ostvariti precizniji i sofisticiraniji model koji će biti konkurentniji u davanju rezultata.

Daljnji koraci uključuju integraciju dodatnih faktora kao što su forma igrača, vremenski uvjeti, povijesni susreti između ekipa, te možda i korištenje naprednijih tehnika strojnog učenja. Također, razmatranje drugih metoda poput Bayesove statistike ili kombinacije s drugim modelima moglo bi poboljšati točnost naših predikcija. Na kraju, vjerujemo da će kontinuirano prilagođavanje i evaluacija modela omogućiti značajno unapređenje u točnosti predikcija nogometnih utakmica.

Literatura

[1] Alan J. Lee, “Modeling scores in the premier league: Is manchester united really the best?” 1997., pristupljeno: 2024-04-13. [Mrežno]. Adresa: https://www.math.ntnu.no/emner/TMA4315/2017h/Lee1997.pdf

[2] Lars Schiefler. (2015) Football club elo ratings. Pristupljeno: 2024-04-10. [Mrežno]. Adresa: http://clubelo.com/

[3] Wikipedia contributors. (2024) Poisson regression. Pristupljeno: 2024-06-11. [Mrežno]. Adresa: https://en.wikipedia.org/wiki/Poisson_regression

[4] IBM. (2024) Statistička značajnost. Pristupljeno: 2024-06-13. [Mrežno]. Adresa: https://www.ibm.com/docs/hr/cognos-analytics/11.1.0?topic=dashboards-statistical-terms

Sažetak

Model za predviđanje ishoda nogometnih utakmica utemeljen na Elo sustavu rangiranja

Nino Ćurko

Ovim radom cilj je demonstrirati programskim kodom i alatima kako modelirati ishode nogometnih utakmica. Za rangiranje pojedinih timova odlučili smo se za sustav koji je uvelike primjenjivan u šahu: Elo sustav rangiranja. Ovim putem smo modelirali jedan jednostavan model na temelju bodova domaće i gostujuće ekipe. Uz primjenu Elo sustava rangiranja, kako bi modelirali sofisticiraniji model, odlučili smo se za Poissonovu regresiju koja se danas smatra prihvatljivom u području nogometa pošto nam pruža očekivani broj golova svake pojedine ekipe.

Primjenom Elo sustava rangiranja u Poissonovu regresiju, uspjeli smo postići određenu razinu točnosti predviđanja, koja je značajno veća od slučajnog biranja jednog od tri ishoda nogometne utakmice (pobjeda domaćina, neriješeno, pobjeda gostiju). Također smo analizirali različite faktore koji mogu utjecati na ishode utakmica, kao što su trenutna forma ekipe, povijesni rezultati i statistički podaci o postignutim i primljenim golovima.

Na kraju analize primijetili smo da model ima određene nedostatke, posebno u predviđanju neriješenih ishoda utakmica. Ovo ukazuje na potrebu za daljnjim unapređenjem modela, možda uključivanjem dodatnih varijabli ili korištenjem naprednijih statističkih tehnika. Unatoč tome, kombinacija Elo sustava i Poissonove regresije pokazala je značajan potencijal za predikciju nogometnih rezultata i može poslužiti kao dobra osnova za buduća istraživanja i poboljšanja u ovom području.

Abstract

Model for prediction of outcomes of football matches based on Elo rating system

Nino Ćurko

The aim of this work is to demonstrate, using programming code and tools, how to model the outcomes of football matches. For ranking individual teams, we chose a system widely used in chess: the Elo rating system. Using this system, we modeled a simple initial model based on the points of the home and away teams. To create a more sophisticated model, we decided to use Poisson regression, which is considered appropriate in the context of football as it provides the expected number of goals for each team.

By incorporating the Elo rating system into Poisson regression, we achieved a certain level of prediction accuracy, significantly higher than random guessing one of the three possible outcomes of a football match (home win, draw, away win). We also analyzed various factors that can influence match outcomes, such as current team form, historical results, and statistical data on goals scored and conceded.

In the final analysis, we observed that the model has certain shortcomings, particularly in predicting drawn matches. This indicates the need for further model improvements, possibly by including additional variables or using more advanced statistical techniques. Despite these issues, the combination of the Elo system and Poisson regression showed significant potential for predicting football results and can serve as a solid foundation for future research and improvements in this field.