

Automatizacija matematičkog modela u procjeni vjerojatnosti ishoda nogometnih utakmica primjenom Poissonovog modela

Ištvanović, Antonio

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, School of Applied Mathematics and Informatics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:126:543994>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-07**



mathos

Repository / Repozitorij:

[Repository of School of Applied Mathematics and Informatics](#)





SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET PRIMIJENJENE MATEMATIKE I INFORMATIKE

Sveučilišni prijediplomski studij Matematika i računarstvo

Automatizacija matematičkog modela u procjeni vjerojatnosti ishoda nogometnih utakmica primjenom Poissonovog modela

ZAVRŠNI RAD

Mentor:

**izv. prof. dr. sc.
Danijel Grahovac**

Student:

Antonio Ištvanović

Osijek, 2024

Sadržaj

1	Uvod	1
2	Osnovne definicije i Poissonova distribucija	3
2.1	Diskretna slučajna varijabla	3
2.2	Distribucija diskretne slučajne varijable	3
2.3	Grafički prikaz distribucije	4
2.4	Očekivanje i varijanca	5
2.5	Poissonova distribucija i njezina svojstva	6
2.6	Osnovni Poissonov model	8
3	Implementacija web aplikacije u Svelte frameworku	11
3.1	Dohvaćanje podataka	11
3.2	Obrada podataka i računanje	15
3.2.1	Funkcija za predviđanje ishoda utakmice	16
3.2.2	Funkcija za računanje prosječnog broja golova po momčadi	16
3.3	Dohvaćanje podataka na klijentu	17
3.4	Prikaz podataka na klijentu	19
3.5	Prikaz isplativih oklada	19
3.6	Arhitektura aplikacije	22
	Literatura	25
	Sažetak	27
	Summary	29
	Životopis	31

1 | Uvod

Kažemo da je nogomet najvažnija sporedna stvar na svijetu, jedan je od najpopularnijih sportova današnjice i neprestano privlači milijune gledatelja. Ljepota ovog sporta je u dinamičnosti i nepredvidivosti, što ga čini privlačnim ne samo kao igru nego i kao predmet izučavanja i predviđanja. Nogometne utakmice, od malih lokalnih dvoboja do raznih prvenstava poput Svjetskog ili Europskog nogometnog prvenstva, okupljaju ljubitelje ovog sporta svih generacija i društvenih slojeva. Primjerice, 2018. godine, kada je Hrvatska osvojila drugo mjesto na Svjetskom prvenstvu u Rusiji, euforija je zahvatila čitavu zemlju, a finalnu utakmicu između Hrvatske i Francuske pratilo je više od milijardu gledatelja diljem svijeta.

Upravo ta nepredvidivost i faktor iznenađenja čini nogomet zanimljivim ne samo za gledanje, već i za klađenje. Predviđanje ishoda nogometnih utakmica postalo je popularna aktivnost, posebno s rastom online kladionica i dostupnošću podataka. No, dok mnogi ljubitelji sporta pristupaju klađenju intuitivno, postoje sofisticirani matematički modeli koji omogućuju precizniju analizu i predviđanje rezultata od ljudske intuicije.

Jedan takav model temelji se na Poissonovoj distribuciji, koja ima svoje korijene u radu francuskog matematičara Siméona Denisa Poissona iz 19. stoljeća. Ovaj statistički model, izvorno razvijen za analizu rijetkih događaja, kao što su nezgode ili greške u proizvodnim procesima, danas se široko koristi u sportskoj analitici za predviđanje broja golova u nogometnim utakmicama. Poissonova distribucija omogućuje procjenu vjerojatnosti da će se određeni broj golova postići tijekom utakmice, uzimajući u obzir povijesne podatke i mogućnosti određene momčadi.

U ovom radu, istražiti ćemo primjenu Poissonovog modela za predviđanje ishoda nogometnih utakmica. U prvom dijelu, fokusirat ćemo se na teorijske osnove, definiciju Poissonove distribucije i njezinu primjenu u nogometu. U drugom dijelu, opisat ćemo razvoj web aplikacije koja koristi ovaj model za analizu ponuda sportskih kladionica. Aplikacija je razvijena u Svelte frameworku, dok su podaci pohranjeni i obrađeni pomoću Firebase platforme. Ova tehnička implementacija omogućuje automatizaciju procesa prikupljanja i analize podataka, što rezultira točnijim predviđanjima i potencijalno većim uspjesima u klađenju.

2 | Osnovne definicije i Poissonova distribucija

Događanja u sportu kao što su postignuti golovi u nogometnim utakmicama, mogu se smatrati slučajnim događanjima. Nogometna utakmica traje regularno 90 minuta, ono što možemo uočiti je da tada golove možemo rasporediti nasumično na vremenskom intervalu od 90 minuta jer bismo time obuhvatili cijeli regularni dio utakmice. Niz koji smo stvorili nasumičnim raspoređivanjem golova možemo gledati kao proces prebrojavanja nasumičnih događaja. Za modeliranje takvih podataka, gdje brojimo događaje unutar nekog vremenskog intervala može nam pomoći Poissonova distribucija. Dakle, ako se očekuje da će se nešto dogoditi 10 puta, mi uz pomoć Poissonove distribucije možemo izračunati vjerojatnost za svako moguće pojavljivanje tog događaja, a u našem slučaju gola. U ovom poglavlju pojasnit ćemo slučajnu vribablu, pojmove očekivanja i varijance kao i Poissonove distribucije i njezina svojstva.

2.1 Diskretna slučajna varijabla

Diskretan vjerojatnosni prostor karakteriziran je činjenicom da skup elementarnih događaja Ω ima konačno ili prebrojivo mnogo elemenata. U takvom prostoru, svaka funkcija $X : \Omega \rightarrow \mathbb{R}$ naziva se *diskretna slučajna varijabla*.

Definicija 1. (vidjeti [1, Definicija 2.1]) Neka je dan diskretan vjerojatnosni prostor $(\Omega, \mathcal{P}(\Omega), P)$. Svaka funkcija $X : \Omega \rightarrow \mathbb{R}$ naziva se *diskretna slučajna varijabla*.

Primjer koji ilustrira ovaj koncept je ispit koji se sastoji od 10 pitanja višestrukog izbora. U ovom slučaju, kandidat je prvenstveno zainteresiran za broj točnih odgovora, a ne za redoslijed točnih odgovora. Dakle, relevantan skup mogućih ishoda možemo definirati kao $\{0, 1, 2, \dots, 10\}$.

2.2 Distribucija diskretne slučajne varijable

Distribucija diskretne slučajne varijable definira se pomoću skupa mogućih vrijednosti koje varijabla može poprimiti i pripadnih vjerojatnosti. Skup svih vrijednosti koje slučajna varijabla X može primiti označavamo s $R(X) = \{x_i : i \in I\}$, dok pripadne vjerojatnosti označavamo nizom $(p_i, i \in I)$ gdje vrijedi $p_i = P(X = x_i)$.

Primjer 1. (vidjeti [1, Primjer 2.3]) Kao primjer razmotrimo situaciju na skladištu koje sadrži četiri istovrsna proizvoda. Vjerojatnost prodaje jednog proizvoda u jednom danu iznosi $\frac{1}{2}$. Broj prodanih proizvoda u jednom danu tada možemo modelirati kao slučajnu varijablu X , koja može poprimiti vrijednosti od 0 do 4. Tablica distribucije za ovu varijablu je sljedeća:

$$X \sim \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ \frac{1}{16} & \frac{1}{4} & \frac{3}{8} & \frac{1}{4} & \frac{1}{16} \end{pmatrix}$$

Općenito, za diskretnu slučajnu varijablu možemo definirati dva bitna skupa brojeva: skup svih vrijednosti $R(X)$ koje slučajna varijabla X može primiti i niz pripadnih vjerojatnosti $(p_i, i \in I)$ za koji vrijedi $\sum_{i=1}^{\infty} p_i = 1$.

Primjer 2. (vidjeti [1, Primjer 2.4])

Primjer diskretne slučajne varijable je zbroj brojeva koji se pojave prilikom bacanja dviju pravilnih kockica. Distribucija ove slučajne varijable može se prikazati tablicom distribucije:

$$X \sim \begin{pmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \frac{1}{36} & \frac{2}{36} & \frac{3}{36} & \frac{4}{36} & \frac{5}{36} & \frac{6}{36} & \frac{5}{36} & \frac{4}{36} & \frac{3}{36} & \frac{2}{36} & \frac{1}{36} \end{pmatrix}$$

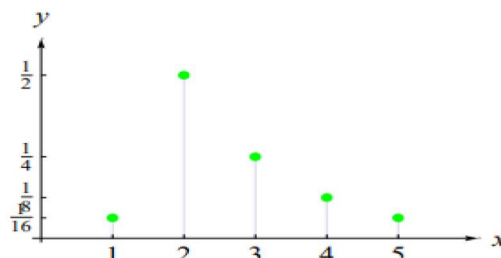
Ako nas zanima vjerojatnost događaja da je zbroj brojeva na kockicama manji od 6, koristimo tablicu distribucije:

$$P(X < 6) = \frac{1}{36} + \frac{2}{36} + \frac{3}{36} + \frac{4}{36} = \frac{5}{18}$$

2.3 Grafički prikaz distribucije

Distribuciju diskretne slučajne varijable možemo grafički prikazati pomoću grafa funkcije koja svakom elementu $x_i \in R(X)$ pridružuje pripadnu vjerojatnost p_i . Također, možemo koristiti stupčasti dijagram, gdje svaki stupac predstavlja vjerojatnost pripadnu određenoj vrijednosti x_i .

Na taj način, distribucija slučajne varijable postaje jasnija i preglednija za interpretaciju, omogućava bolje razumijevanje svojstava promatranog slučajnog pokusa.



Slika 2.1: Graf distribucije diskretne slučajne varijable iz primjera 1 (vidjeti [1, Slika 2.2])

2.4 Očekivanje i varijanca

Očekivanje i varijanca osnovne su numeričke karakteristike slučajne varijable, a pružaju važne informacije o njenom ponašanju. Očekivanje (ili matematičko očekivanje) predstavlja prosječnu vrijednost koju slučajna varijabla poprima u velikom broju ponavljanja eksperimenta, dok varijanca mjeri raspršenost vrijednosti slučajne varijable oko njenog očekivanja.

Očekivanje

Definicija 2. (vidjeti [1, Definicija 2.12]) Neka je $(\Omega, \mathcal{P}(\Omega), P)$ diskretan vjerojatnosni prostor i X slučajna varijabla na njemu. Ako red

$$\sum_{\omega \in \Omega} |X(\omega)|P(\{\omega\})$$

apsolutno konvergira, kažemo da X ima matematičko očekivanje, koje je definirano kao:

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} X(\omega)P(\{\omega\}).$$

Ova definicija često se koristi u teoriji, dok se u praksi očekivanje računa pomoću distribucije diskretne slučajne varijable:

$$\mathbb{E}[X] = \sum_{i \in \mathbb{N}} x_i p_i,$$

gdje x_i predstavlja moguće vrijednosti varijable X , a p_i pripadajuće vjerojatnosti.

Očekivanje ima nekoliko korisnih svojstava, među kojima su:

- **Linearnost:** Za slučajne varijable X i Y i konstante a i b vrijedi:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y].$$

- **Monotonost:** Ako za sve $\omega \in \Omega$ vrijedi $X(\omega) \leq Y(\omega)$, tada je $\mathbb{E}[X] \leq \mathbb{E}[Y]$.

- **Pozitivnost:** Ako je $X(\omega) \geq 0$ za sve $\omega \in \Omega$, tada je $\mathbb{E}[X] \geq 0$.

Varijanca

Varijanca mjeri koliko su vrijednosti slučajne varijable raspršene oko njenog očekivanja.

Definicija 3. (vidjeti [1, Definicija 2.13]) Neka je X slučajna varijabla s očekivanjem $\mathbb{E}[X]$. Varijanca od X je definirana kao:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2].$$

Varijanca se također može izraziti pomoću očekivanja od X i X^2 :

$$\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2.$$

Standardna devijacija, koja se označava sa σ_X , kvadratni je korjen iz varijance:

$$\sigma_X = \sqrt{\text{Var}(X)}.$$

Varijanca daje ključnu informaciju o disperziji podataka; što je varijanca veća, veća je i raspršenost vrijednosti varijable oko njenog očekivanja. Zajedno s očekivanjem, predstavlja ključne parametre za razumjevanje distribucije slučajnih varijabli i njihove primjene u statistici i vjerojatnosti.

2.5 Poissonova distribucija i njezina svojstva

Poissonovom distribucijom modeliramo slučajnu varijablu koja broji uspjehe u nekom intervalu vremena, površine, mase i slično. Poissonova distribucija može biti prikladna kada vrijedi sljedeće:

1. **Homogenost vjerojatnosti pojavljivanja uspjeha:** Vjerojatnost pojavljivanja uspjeha ne ovisi o tome u kojem će se jediničnom intervalu dogoditi u kontekstu modeliranja broja golova na nogometnoj utakmici. To znači da svaki napad na gol tijekom utakmice ima istu vjerojatnost rezultirati golom, bez obzira na to u kojem se trenutku utakmice napad dogodio.
2. **Neovisnost broja uspjeha u različitim intervalima:** Broj uspjeha u jednom intervalu neovisan je o broju uspjeha u bilo kojem drugom intervalu. U kontekstu nogometne utakmice, to znači da broj golova postignutih u jednom dijelu utakmice ne utječe na broj golova postignutih u drugom dijelu utakmice.
3. **Jednaka očekivana vrijednost uspjeha:** Očekivani broj uspjeha isti je za sve jedinične intervale koje odaberemo i dan je pozitivnim realnim brojem λ . U kontekstu nogometne utakmice, λ predstavlja očekivani broj golova po utakmici, i ovaj parametar ostaje konstantan kroz cijelu utakmicu.

Primjenom prethodno navedenih uvjeta na nogometnu utakmicu, možemo zaključiti da nogometna utakmica u pravilu ima prethodne karakteristike Poissonove distribucije.

Definicija 4. Poissonova distribucija (vidjeti [1, Definicija 2.5]). Slučajna varijabla X ima Poissonovu distribuciju s parametrom $\lambda > 0$ ako prima vrijednosti iz skupa $\{0, 1, 2, \dots\}$ s vjerojatnostima

$$p_i = P\{X = i\} = \frac{e^{-\lambda} \lambda^i}{i!}.$$

Tada pišemo $X \sim P(\lambda)$.

Na Slici 2.2 prikazana je Poissonova distribucija za različite vrijednosti parametra λ .

Provjerimo je li na taj način dobro definirana distribucija, tj. je li $\sum_{i=0}^{\infty} p_i = 1$. Vrijedi:

$$\sum_{i=0}^{\infty} \frac{e^{-\lambda} \lambda^i}{i!} = e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^i}{i!} = e^{-\lambda} e^{\lambda} = 1.$$

Jedinstveno svojstvo Poissonove distribucije je da su očekivanje i varijanca jednaki λ .

Očekivanje možemo izračunati kao:

$$E[X] = \sum_{k=0}^{\infty} k \cdot P(X = k) = \sum_{k=0}^{\infty} k \cdot \frac{e^{-\lambda} \lambda^k}{k!}.$$

Koristeći $k^2 = k(k-1) + k$, dobivamo:

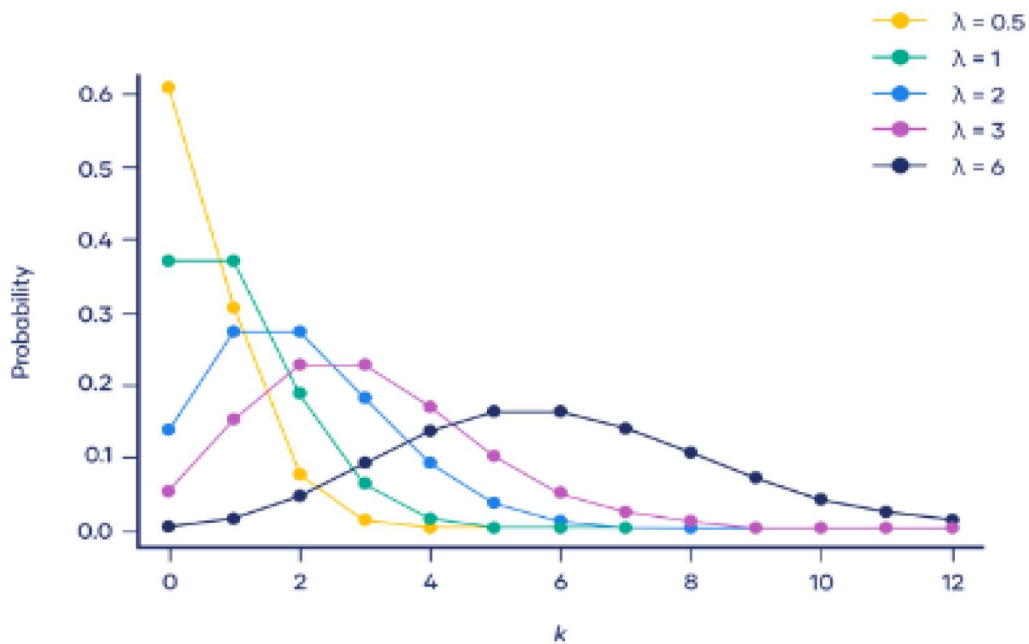
$$E[X^2] = e^{-\lambda} \left(\lambda + \sum_{k=2}^{\infty} \frac{(k-1)k\lambda^k}{k!} + \sum_{k=2}^{\infty} \frac{k\lambda^k}{k!} \right).$$

Prepravljajući izraze, dobivamo:

$$E[X^2] = e^{-\lambda} (\lambda e^{\lambda} + \lambda^2 e^{\lambda}) = \lambda + \lambda^2.$$

Sada možemo izračunati varijancu:

$$\text{Var}(X) = E[X^2] - (E[X])^2 = (\lambda + \lambda^2) - \lambda^2 = \lambda.$$



Slika 2.2: Poissonova distribucija za različite vrijednosti parametra λ (vidjeti [6])

2.6 Osnovni Poissonov model

Nakon što smo definirali Poissonovu distribuciju, možemo prijeći na izgradnju modela koji će predviđati ishode utakmica pomoću podataka koje nam daju kladionice i povijesnih podataka o odigranim dvobojima.

Model za predviđanje vjerojatnosti rezultata, odnosno ishoda utakmice koristi tri varijable (vidjeti [2]) :

- λ_{home} : prosječan broj golova domaćina,
- λ_{away} : prosječan broj golova gostujuće ekipe,
- k_{max} : maksimalan broj golova koji je fiksno postavljen na 10.

Povijesni podaci smatraju se mjerom napadačke i obrambene snage ekipa. Neka je:

1. Napadačka snaga domaćina:

$$AS_{\text{home}} = \frac{\text{Ukupan broj golova domaćina u sezoni}}{\text{Ukupan broj utakmica domaćina}}$$

2. Napadačka snaga gostiju:

$$AS_{\text{away}} = \frac{\text{Ukupan broj golova gostiju u sezoni}}{\text{Ukupan broj utakmica gostiju}}$$

3. Obrambena snaga domaćina:

$$DS_{\text{home}} = \frac{\text{Ukupan broj primljenih golova od strane domaćina}}{\text{Ukupan broj utakmica domaćina}}$$

4. Obrambena snaga gostiju:

$$DS_{\text{away}} = \frac{\text{Ukupan broj primljenih golova od strane gostiju}}{\text{Ukupan broj utakmica gostiju}}$$

Vjerojatnost da domaćin postigne k golova može se modelirati kao Poissonova distribucija s parametrom λ_{home} , gdje je:

$$\lambda_{\text{home}} = AS_{\text{home}} \times DS_{\text{away}}$$

Slično, vjerojatnost da gostujuća ekipa postigne k golova može se modelirati kao Poissonova distribucija s parametrom λ_{away} , gdje je:

$$\lambda_{\text{away}} = AS_{\text{away}} \times DS_{\text{home}}$$

Kako bi se ograničio broj golova u modelu, pretpostavlja se da se broj golova kreće između 0 i $k_{\text{max}} = 10$. Dakle, konačni model za vjerojatnost rezultata utakmice definira se kao:

$$P(\text{rezultat} = (k_{\text{home}}, k_{\text{away}})) = P(\text{domaćin postiže } k_{\text{home}} \text{ golova}) \\ \times P(\text{gost postiže } k_{\text{away}} \text{ golova})$$

gdje $k_{\text{home}}, k_{\text{away}} \in \{0, 1, 2, \dots, 10\}$.

3 | Implementacija web aplikacije u Svelte frameworku

U ovom poglavlju opisat ćemo tehničku implementaciju web aplikacije koja koristi Poissonov model za analizu ponuda sportskih kladionica. Aplikacija je razvijena u Svelte frameworku, dok su podaci pohranjeni i obrađeni pomoću Firebase platforme. Projekt je podijeljen na dva glavna dijela, klijent i Firebase funkcije koje imaju ulogu servera. Podatke prikupljamo pomoću API-ja koji pruža podatke uživo o nogometnim utakmicama, dok povijesne podatke analiziramo iz CSV datoteke koja sadrži podatke o odigranim utakmicama. CSV datoteku povlačimo iz Firebase Storage-a.

3.1 Dohvaćanje podataka

Svi su korišteni podaci javno dostupni na internetskim stranicama koje se bave podacima o nogometnim utakmicama (API za utakmice i kvote, CSV datoteka o povijesnim podacima utakmica). Podaci o utakmicama dnevno se povlače s API-ja, zapisani su u listi JSON objekata, a povijesni podaci u CSV datoteci. Baza podataka za povijesne podatke učitava se tako što iz Firebase Storage povlačimo CSV, a onda pomoću CSV parsera tu datoteku čitamo.

```
1 import * as functions from 'firebase-functions';
2 import axios from 'axios';
3 import * as dotenv from 'dotenv';
4 import * as cors from 'cors';
5
6 dotenv.config();
7
8 const API_KEY = process.env.API_KEY;
9
10 const corsHandler = cors({ origin: true });
11
12 export const getOdds = functions.https.onRequest((req, res) => {
13   corsHandler(req, res, async () => {
14     const { sport } = req.query;
15     if (!sport) {
16       res.status(400).send('Missing sport parameter');
17       return;
18     }
19   })
20 }
```



```
20     try {
21         const response = await axios.get('https://api.the-odds-api.
com/v4/sports/${sport}/odds', {
22             params: {
23                 regions: 'eu',
24                 oddsFormat: 'decimal',
25                 apiKey: API_KEY
26             }
27         });
28
29         let oddsData = response.data;
30
31         // demo utakmica za Bundesligu
32         const demoMatch = {
33             id: 'demo-match-1',
34             sport_key: 'soccer_germany_bundesliga',
35             commence_time: '2023-01-01T12:00:00Z',
36             home_team: 'Bayern Munchen',
37             away_team: 'Borussia Dortmund',
38             bookmakers: [
39                 {
40                     key: 'demo_bookmaker',
41                     title: 'Demo Bookmaker',
42                     markets: [
43                         {
44                             key: 'h2h',
45                             outcomes: [
46                                 { name: 'Bayern Munchen', price: 1.73 },
47                                 { name: 'Draw', price: 3.10 },
48                                 { name: 'Borussia Dortmund', price: 5.50 }
49                             ]
50                         }
51                     ]
52                 }
53             ]
54         };
55
56         const demoMatch2 = {
57             id: 'demo-match-2',
58             sport_key: 'soccer_germany_bundesliga1',
59             commence_time: '2024-08-09T20:00:00Z',
60             home_team: 'FC Schalke 04',
61             away_team: 'Hamburger SV',
62             bookmakers: [
63                 {
64                     key: 'demo_bookmaker',
65                     title: 'Demo Bookmaker',
66                     markets: [
67                         {
68                             key: 'h2h',
69                             outcomes: [
70                                 { name: 'FC Schalke 04', price: 3.0 },
71                                 { name: 'Draw', price: 2.4 },
72                                 { name: 'Hamburger SV', price: 3.0 }
73                             ]
74                         }
75                     ]
76                 }
77             ]
78         };
79     }
80 }
```

```
75         ]
76     }
77 ]
78 };
79
80 oddsData.push(demoMatch, demoMatch2);
81
82 res.status(200).send(oddsData);
83 } catch (error) {
84     console.error('Error getting odds:', error);
85     res.status(500).send('Failed to fetch odds');
86 }
87 });
88 });
```

Ovdje definiramo Firebase funkciju koja služi za preuzimanje podataka uživo s API-ja za određeni sport i vraća te podatke kao HTTP odgovor.

```
1 import * as functions from 'firebase-functions';
2 import { admin } from '../init';
3 import * as csv from 'csv-parser';
4
5 const bucket = admin.storage().bucket();
6
7 export const getCSV = functions.https.onCall(async (data, context)
8     => {
9     const file = bucket.file('germany.csv');
10    let csvData: any[] = [];
11
12    return new Promise((resolve, reject) => {
13        file.createReadStream()
14            .pipe(csv())
15            .on('data', (row) => {
16                csvData.push(row);
17            })
18            .on('end', () => {
19                resolve({ csvData });
20            })
21            .on('error', (error) => {
22                reject(error);
23            });
24    });
```

Ovo je također Firebase funkcija, a služi za preuzimanje i čitanje CSV datoteke "germany.csv" iz Firebase Storage-a i vraća podatke kao JSON objekt.

Prije korištenja podataka moramo pripaziti na formate. Bilo je potrebno napraviti funkciju koja normalizira imena timova u CSV i funkciju *preprocessData* u podacima povučenim s API-ja.

```
1 export function normalizeTeamName(name: string): string {
2   if (!name) {
3     console.error('Received undefined or empty team name');
4     return '';
5   }
6   return name
7     .replace(/^\\s+|\\s+$/g, '') // Trim leading/trailing
  whitespace
8     .replace(/\\s+/g, ' ') // Replace multiple spaces with a
  single space
9     .replace(/[~a-zA-Z0-9\\s]/g, '') // Remove non-alphanumeric
  characters
10 }
```

preporpcessData.ts je funkcija koja prima podatke iz CSV datoteke i s API-ja i vraća podatke u jednakom obliku.

```
1 import {normalizeTeamName} from "$lib/utils";
2
3 export function preprocessData(data: any[]) {
4   return data.map((row) => {
5     if (!row.home || !row.visitor) {
6       console.error('Home or away team is undefined:', row);
7       return null;
8     }
9
10    const goalsHome = parseInt(row.hgoal, 10);
11    const goalsAway = parseInt(row.vgoal, 10);
12
13    if (isNaN(goalsHome) || isNaN(goalsAway)) {
14      console.error('Invalid goals data:', row);
15      return null;
16    }
17
18    return {
19      date: new Date(row.Date),
20      season: row.Season,
21      home_team: normalizeTeamName(row.home),
22      away_team: normalizeTeamName(row.visitor),
23      goals_home: goalsHome,
24      goals_away: goalsAway,
25      tier: row.tier,
26      division: row.division
27    };
28  }).filter(row => row !== null);
29 }
```

3.2 Obrada podataka i računanje

Nakon što smo povukli podatke, slijedi njihova obrada i računanje vjerojatnosti ishoda utakmica. U ovom koraku koristimo Poissonov model za analizu ponuda sportskih kladionica.

```
1 import { create, all } from 'mathjs';
2
3 const math = create(all);
4
5 function poissonProbability(lambda: number, k: number): number {
6   return (Math.pow(lambda, k) * Math.exp(-lambda)) / math.factorial
7     (k);
8 }
9 export function poissonModel(avgGoalsHome: number, avgGoalsAway:
10   number, maxGoals: number = 10) {
11   const homeGoals = Array.from({ length: maxGoals }, (_, k) =>
12     poissonProbability(avgGoalsHome, k));
13   const awayGoals = Array.from({ length: maxGoals }, (_, k) =>
14     poissonProbability(avgGoalsAway, k));
15   return { homeGoals, awayGoals };
16 }
```

Koristili smo biblioteku *mathjs* koja sadrži različite funkcije. Funkcija Poissonove vjerojatnosti računa da će se dogoditi točno k događaja (u ovom slučaju golova), kada je prosječna stopa događaja λ .

Pri tome je:

- λ je prosječan broj događaja (golova)
- k je broj događaja za koje želimo izračunati vjerojatnost
- e je baza prirodnog logaritma (≈ 2.718)
- $k!$ je faktorijel broja k

Funkcija za kreiranje Poissonovog modela za golove je funkcija koja koristi Poissonovu distribuciju da bi generirala vjerojatnosti za postizanje različitog broja golova za domaći i gostujući tim. Ova funkcija kreira niz vjerojatnosti za sve moguće brojeve golova od 0 do „maxGoals“.

Pri tome je:

- „avgGoalsHome“ je prosječan broj golova koje domaći tim postiže
- „avgGoalsAway“ je prosječan broj golova koje gostujući tim postiže
- „maxGoals“ je maksimalan broj golova za koje želimo izračunati vjerojatnosti (zadana vrijednost je 10)

3.2.1 Funkcija za predviđanje ishoda utakmice

Funkcija *predictOutcome* računa vjerojatnosti ishoda utakmice (pobjeda domaćina, neriješeno, pobjeda gostujuće ekipe) koristeći unaprijed izračunate vjerojatnosti postizanja određenog broja golova za domaću i gostujuću ekipu.

```
1 export function predictOutcome(homeGoals: number[], awayGoals:
  number[]): { homeWin: number, draw: number, awayWin: number } {
2   let homeWin = 0;
3   let draw = 0;
4   let awayWin = 0;
5
6   for (let i = 0; i < homeGoals.length; i++) {
7     for (let j = 0; j < awayGoals.length; j++) {
8       const prob = homeGoals[i] * awayGoals[j];
9       if (i > j) homeWin += prob;
10      else if (i === j) draw += prob;
11      else awayWin += prob;
12    }
13  }
14
15  return { homeWin, draw, awayWin };
16 }
```

3.2.2 Funkcija za računanje prosječnog broja golova po momčadi

Funkcija *calculateTeamAverages* računa prosječan broj golova po momčadi na temelju povijesnih podataka o odigranim utakmicama iz CSV datoteke.

```
1 export function calculateTeamAverages(csvData, homeTeam: string,
  awayTeam: string) {
2   let homeGoalsFor = 0, homeGoalsAgainst = 0, homeGames = 0;
3   let awayGoalsFor = 0, awayGoalsAgainst = 0, awayGames = 0;
4
5   csvData.forEach(match => {
6     if (match.home_team === homeTeam) {
7       homeGoalsFor += match.goals_home;
8       homeGoalsAgainst += match.goals_away;
9       homeGames++;
10    }
11    if (match.away_team === awayTeam) {
12      awayGoalsFor += match.goals_away;
13      awayGoalsAgainst += match.goals_home;
14      awayGames++;
15    }
16  });
17
18  const avgGoalsHomeFor = homeGames > 0 ? homeGoalsFor /
  homeGames : 0;
19  const avgGoalsHomeAgainst = homeGames > 0 ? homeGoalsAgainst /
  homeGames : 0;
```

```
20     const avgGoalsAwayFor = awayGames > 0 ? awayGoalsFor /
    awayGames : 0;
21     const avgGoalsAwayAgainst = awayGames > 0 ? awayGoalsAgainst /
    awayGames : 0;
22
23     return {
24         avgGoalsHomeFor ,
25         avgGoalsHomeAgainst ,
26         avgGoalsAwayFor ,
27         avgGoalsAwayAgainst
28     };
29 }
```

3.3 Dohvaćanje podataka na klijentu

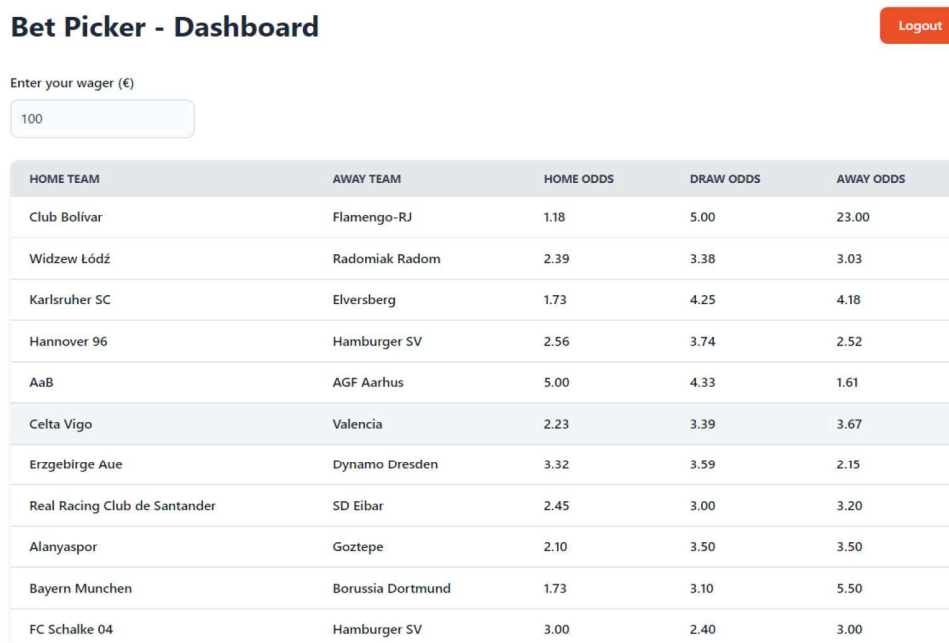
api.ts sadrži dvije funkcije koje dohvaćaju podatke iz Firebase funkcija. Prva funkcija *getOdds* dohvaća podatke o kvotama i protivnicima s API-ja, a druga funkcija *getCSV* dohvaća povijesne podatke iz CSV datoteke. Dakle, ove funkcije su pozivi prema backendu koji je hostan u Google Cloud-u putem Firebase-a, što znači da se funkcije izvršavaju na zahtjev klijenta i mogu uključivati razne operacije, poput dohvaćanja podataka iz baza ili generiranja izvještaja.

```
1 import axios from 'axios';
2 import { httpsCallable } from "firebase/functions";
3 import { firebaseFunctions } from "$lib/firebase";
4 import { normalizeTeamName } from "$lib/utils";
5
6 const PROJECT_ID = 'bet-picker';
7
8 export async function fetchOdds(sport: string) {
9     try {
10         console.log('Fetching odds for sport: ${sport}');
11
12         const response = await axios.get('http://localhost:5001/${
    PROJECT_ID}/us-central1/api-getOdds', {
13             params: { sport }
14         });
15
16         console.log('Response:', response.data);
17
18         const odds = response.data.map((odd: any) => {
19             const bookmaker = odd.bookmakers?.[0]; // provjerava postoji
    li prvi bookmaker
20             if (!bookmaker) return { home_team: odd.home_team, away_team:
    odd.away_team, home_odds: 'N/A', draw_odds: 'N/A', away_odds: '
    N/A' };
21
22             const homeOdds = bookmaker.markets?.[0]?.outcomes.find((o:
    any) => normalizeTeamName(o.name) === normalizeTeamName(odd.
    home_team))?.price;
```

```
23     const drawOdds = bookmaker.markets?.[0]?.outcomes.find((o:
any) => o.name.toLowerCase() === 'draw')?.price;
24     const awayOdds = bookmaker.markets?.[0]?.outcomes.find((o:
any) => normalizeTeamName(o.name) === normalizeTeamName(odd.
away_team))?.price;
25
26     console.log('Processed Odds:', {
27       home_team: odd.home_team,
28       away_team: odd.away_team,
29       homeOdds,
30       drawOdds,
31       awayOdds
32     });
33
34     return {
35       home_team: odd.home_team,
36       away_team: odd.away_team,
37       home_odds: homeOdds ? parseFloat(homeOdds).toFixed(2) :
null,
38       draw_odds: drawOdds ? parseFloat(drawOdds).toFixed(2) :
null,
39       away_odds: awayOdds ? parseFloat(awayOdds).toFixed(2) :
null
40     };
41   });
42
43   console.log('Final Odds:', odds);
44   return odds;
45 } catch (error) {
46   console.error('Error fetching odds:', error);
47   throw error;
48 }
49 }
50 }
51
52 export const fetchCSVData = async () => {
53   const getCSV = httpsCallable(firebaseFunctions, 'api-getCSV');
54   try {
55     const result = await getCSV();
56     const csvData = result.data.csvData;
57     console.log('Fetched CSV Data:', csvData);
58     return csvData;
59   } catch (error) {
60     console.error('Error fetching CSV data:', error);
61     throw error;
62   }
63 };
```

3.4 Prikaz podataka na klijentu

Podatke o utakmicama koje se učitavaju dnevno smo vizualizirali unutar jedne tablice na početnoj stranici, koristeći Flowbite Svelte paket komponenti, a za Bundesligu ako postoje isplative oklade prikazat će se unutar kartica iz istog paketa komponenti. Također korisnik može u polje za unos uplate unijeti iznos koji bi eventualno htio uplatiti i vidjeti koliko bi imao profit po utakmici i ukupno.



HOME TEAM	AWAY TEAM	HOME ODDS	DRAW ODDS	AWAY ODDS
Club Bolívar	Flamengo-RJ	1.18	5.00	23.00
Widzew Łódź	Radomiak Radom	2.39	3.38	3.03
Karlsruher SC	Elversberg	1.73	4.25	4.18
Hannover 96	Hamburger SV	2.56	3.74	2.52
AaB	AGF Aarhus	5.00	4.33	1.61
Celta Vigo	Valencia	2.23	3.39	3.67
Erzgebirge Aue	Dynamo Dresden	3.32	3.59	2.15
Real Racing Club de Santander	SD Eibar	2.45	3.00	3.20
Alanyaspor	Goztepe	2.10	3.50	3.50
Bayern Munchen	Borussia Dortmund	1.73	3.10	5.50
FC Schalke 04	Hamburger SV	3.00	2.40	3.00

Slika 3.1: Prikaz podataka s API-ja u tablici

3.5 Prikaz isplativih oklada

Oklade koje je Poissonov model prepoznao kao isplative prikazane su unutar kartica ispod tablice s podacima. Isplative utakmice su one za koje je model prepoznao da je implicirana vjerojatnost manja od one koju je model izračunao. Način na koji to računamo je korištenjem Poissonovog modela koji s dvije pomoćne funkcije *calculateTeamAverages* i *predictOutcome* računa prosjeke golova koji su potrebni parametri za model, izračunom vjerojatnosti i usporedbom impliciranih vjerojatnosti možemo zaključiti je li određena utakmica isplativa oklada. Implementacija modela i pomoćnih funkcija opisane su u poglavlju 3.2. U kodu se implicirane vjerojatnosti računaju koristeći:

$$\text{Implicirana vjerojatnost} = \frac{1}{\text{kvota}}$$

Implementacija izgleda na sljedeći način:


```

1 // implicirane vjerojatnosti iz kvota
2 const impliedProbHome = odd.home_odds > 0 ? 1 / odd.home_odds :
  0;
3 const impliedProbDraw = odd.draw_odds > 0 ? 1 / odd.draw_odds :
  0;
4 const impliedProbAway = odd.away_odds > 0 ? 1 / odd.away_odds :
  0;

```

Korištenje modela za procjenu vjerojatnosti:

```

1 // Poissonov model
2 const { homeGoals, awayGoals } = poissonModel(avgGoalsHome,
  avgGoalsAway);
3 const { homeWin, draw, awayWin } = predictOutcome(homeGoals,
  awayGoals);

```

Isplativa oklada identificira se usporedbom vjerojatnosti koju predviđa model s impliciranom vjerojatnosti iz kvota. Ako je vjerojatnost predviđena modelom veća od implicirane vjerojatnosti (uz malu marginu od 0.1%), tada se ta oklada smatra isplativom.

```

1 // usporedi procjene s impliciranim vjerojatnostima
2 const isValueBetHome = homeWin > impliedProbHome * 1.001;
3 const isValueBetDraw = draw > impliedProbDraw * 1.001;
4 const isValueBetAway = awayWin > impliedProbAway * 1.001;

```

Profit se računa na temelju vjerojatnosti ishoda koje su predviđene Poissonovim modelom i usporedbe s impliciranim vjerojatnostima iz kvota.

1. Za svaku pojedinu isplativu okladu:

- Računa se kao:

$$\text{Profit} = (\text{modelna vjerojatnost} \times \text{kvota} \times \text{uplata}) - \text{uplata}$$

2. Ukupni profit:

- Zbroje se svi pojedinačni profiti za svaku isplativu okladu.

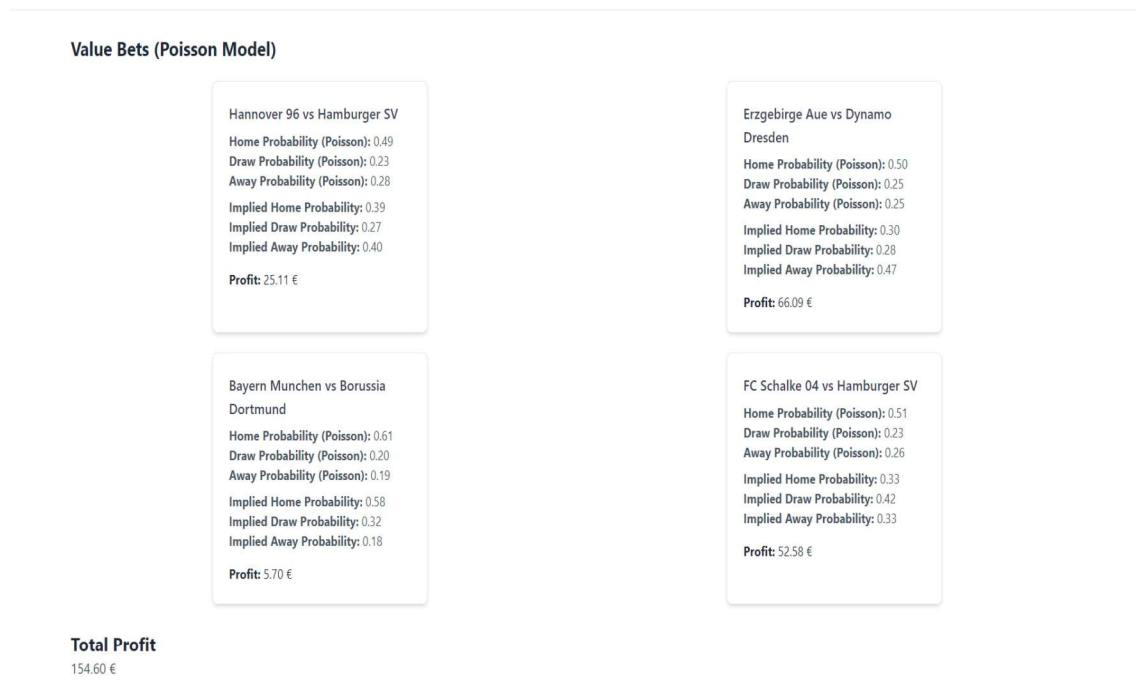
Ako je, primjerice, oklada na pobjedu domaćina prepoznata kao isplativa, tada:

$$\text{Profit} = (\text{vjerojatnost pobjede domaćina prema modelu} \times \\ \text{kvota na pobjedu domaćina} \times \text{uplata}) - \text{uplata}$$

Na sličan način računa se profit za isplative oklade na neriješeno ili pobjedu gosta.

Ako su sve tri vjerojatnosti (za pobjedu domaćina, neriješeno i pobjedu gosta) prepoznate kao isplative, profit za tu utakmicu bit će zbroj svih pojedinačnih profita.

Svaka kartica sadrži informacije o utakmici, vjerojatnosti ishoda, profit po utakmici za određenu uplatu, ovisno o tome koliko korisnik unese, a ispod kartica je prikazan ukupni profit ako bi se korisnik odlučio uplatiti.



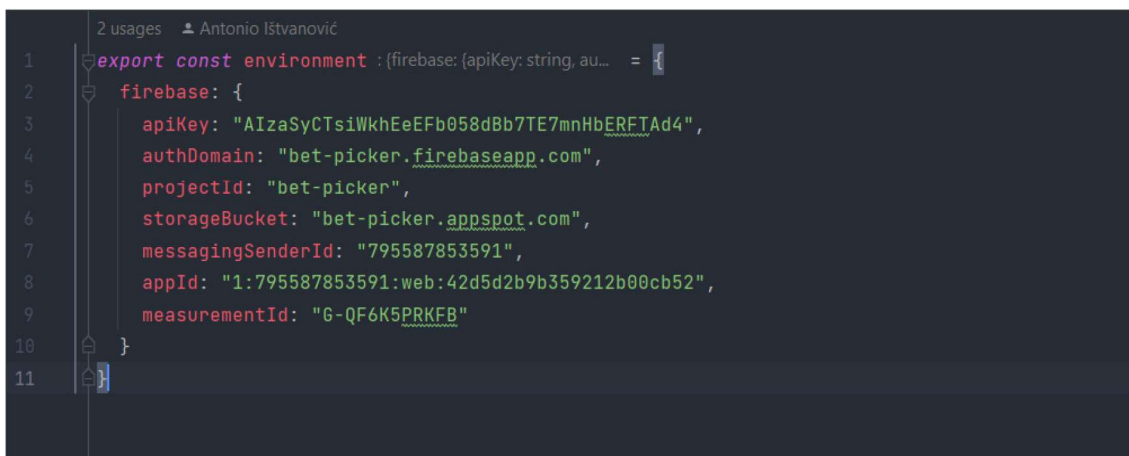
Slika 3.2: Prikaz vjerojatnosti prema modelu i impliciranih vrijednosti

3.6 Arhitektura aplikacije

Aplikacija se sastoji od dva glavna direktorija **Client** i **Functions**.

Client

- Ovaj dio projekta se nalazi u mapi `client` i predstavlja frontend dio aplikacije. Frontend dio je pisan koristeći Svelte-Kit i TypeScript. Također, u aplikaciji smo koristili TailwindCSS za stiliziranje. Kod koji služi za funkcionalnost na frontendu se nalazi u `client/src/lib`, dok su rute poput `SignIn page` i `Dashboard page` smještene u `client/src/routes`. U `client`u je također postavljena konfiguracija i okruženje za Firebase.

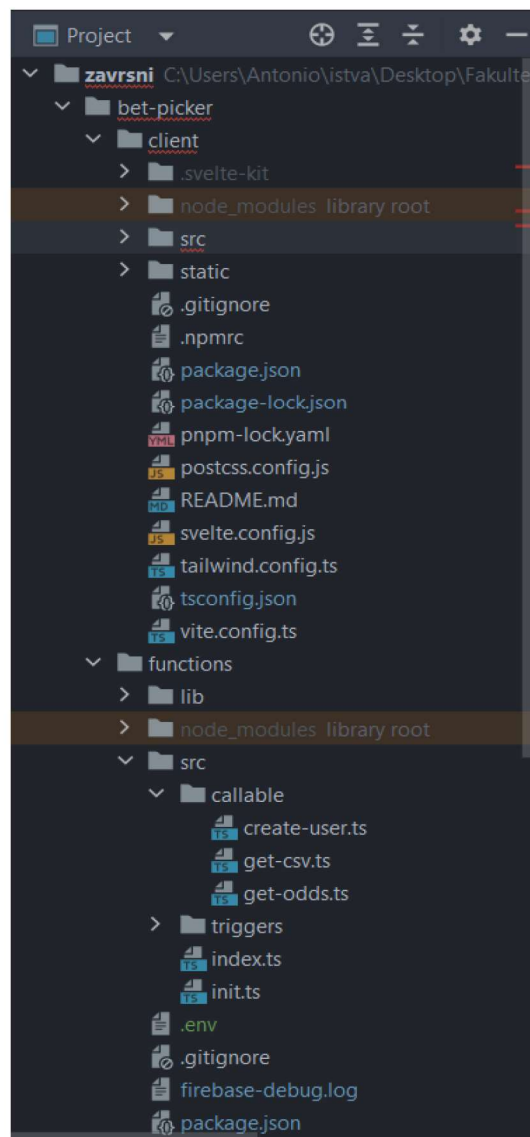
A screenshot of a code editor showing a TypeScript configuration for Firebase. The code is as follows:

```
1 export const environment : {firebase: {apiKey: string, au... = {
2   firebase: {
3     apiKey: "AIzaSyCTsiWkhEeEFb058dBb7TE7mnHbERFTAd4",
4     authDomain: "bet-picker.firebaseio.com",
5     projectId: "bet-picker",
6     storageBucket: "bet-picker.appspot.com",
7     messagingSenderId: "795587853591",
8     appId: "1:795587853591:web:42d5d2b9b359212b00cb52",
9     measurementId: "G-QF6K5PRKFB"
10  }
11 }
```

Slika 3.3: Podaci s Firebase projekta

Functions

- Služi za backend funkcionalnosti. U `functions/src` imamo dva direktorija `callable` i `triggers`. Direktorij `callable` sadrži sve funkcije koje pozivamo na frontend dijelu, odnosno u `clientu`, dok `triggers` direktorij sadrži sve funkcije koje se okidaju nekom akcijom koju korisnik izvede. U `functions/src/callable` imamo datoteke: `get-odds.ts`, `create-user.ts` i `get-csv.ts`, dok u `functions/src/triggers` imamo `delete-user.ts`.



Slika 3.4: Struktura Svelte projekta

Literatura

- [1] M. BENŠIĆ, N. SUVAK, *Uvod u vjerojatnost i statistiku*, Sveučilište J.J. Strossmayera, Odjel za matematiku, Osijek, 2014.
- [2] M. J. MAHER, *Modelling association football scores*, Statist. Neerland. 36, 109–1188 (1982).
- [3] A. A. ALZAID, M. A. OMAIR, *On the Poisson difference distribution inference and applications*, Bull. Malays. Math. Sci. Soc. (2), Vol. 33 (2010), 17–45.
- [4] *Poisson Distribution in Sports Betting*, dostupno na:
<https://medium.com/@xgscoreio2022/poisson-distribution-in-sports-betting-c93d1102e9a5>
- [5] *Sports Betting: Poisson Distribution*, dostupno na:
<https://itmb.medium.com/sports-betting-poisson-distribution-4946333c663a>
- [6] *Poisson Distribution*, dostupno na:
<https://www.scribbr.com/statistics/poisson-distribution>

Sažetak

U ovom radu bavili smo se statističkim pristupom koji se koristi za predviđanje vjerojatnosti određenih ishoda pod nazivom Poissonov model. Rad je podijeljen u dva dijela. Prvi dio analizira teorijske aspekte Poissonove distribucije, uključujući njenu primjenu u predviđanju rezultata nogometnih utakmica. Drugi dio sastoji se od implementacije Poissonovog modela unutar web aplikacije i primjene istog na ponude sportskih kladionica.

Ključne riječi

Poissonova distribucija, vjerojatnost, Poissonov model, sportsko klađenje, Svelte, Firebase

Automation of a Mathematical Model in Estimating the Probability of Football Match Outcomes Using the Poisson Model

Summary

In this paper, we have discussed the Poisson model, a statistical approach used to predict the probability of certain outcomes. The paper is divided into two main sections. The first section provides a detailed analysis of the theoretical aspects of the Poisson distribution, including its application in predicting football match outcomes. The second section focuses on the implementation of the Poisson model within a web application and its application to sports betting odds. The integration of these theoretical and practical components demonstrates the model's effectiveness in real-world scenarios, particularly in the context of sports analytics.

Keywords

Poisson distribution, probability, Poisson model, sport's betting, Svelte, Firebase

Životopis

Rođen sam 24.06.2002. u Osijeku. Obrazovanje započinjem u Osnovnoj školi "*Antun Gustav Matoš*" Vinkovci, a nakon toga sam upisao Opću gimnaziju "*Matije Antuna Reljkovića*" u Vinkovcima koju završavam 2021. godine. Te iste godine upisujem prijediplomski sveučilišni studij Matematike i računarstva na Fakultetu primijenjene matematike i informatike Sveučilišta J.J. Strossmayera u Osijeku. U akademskoj godini 2022./2023. sudjelovao sam u RZC PAN HR projektu kao asistent na vježbama i predavanjima za nastavnike osnovnih i srednjih škola, održao sam nekoliko predavanja vezanih za tehnologije koje se planiraju uvesti u osnovnoškolsko i srednjoškolsko obrazovanje.