Rangiranje ekipa u sportu korištenjem PageRank algoritma

Marinov, Jure

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Jure Marinov

RANGIRANJE EKIPA U SPORTU KORIŠTENJEM PAGERANK ALGORITMA

ZAVRŠNI RAD

Varaždin, 2023.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Jure Marinov

Matični broj: 35918/07–R

Studij: Informacijski sustavi

RANGIRANJE EKIPA U SPORTU KORIŠTENJEM PAGERANK ALGORITMA

ZAVRŠNI RAD

Mentor/Mentorica:

Jelena Gusić Munđar, mag. math.

Varaždin, rujan 2023.

Jure Marinov

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

Ovaj završni rad bavi se primjenom PageRank algoritma za rangiranje nogometnih reprezentacija koje su sudjelovale na Svjetskom nogometnom prvenstvu u Katru 2022. Rad pruža dublji uvid u matematičke i teorijske osnove PageRank algoritma, često korištenog za rangiranje web stranica, i njegovu primjenu u sportskom kontekstu. U teorijskom dijelu rada detaljno su objašnjene metode i tehnike koje stoje iza PageRank algoritma, uključujući koncept Markovljevih lanaca, teorem Perron-Frobenius i njegovu primjenu. Također, razmatra se kako je PageRank algoritam evoluirao iz digitalne svijesti u ključan alat za rangiranje različitih entiteta. Praktični dio rada uključuje implementaciju PageRank algoritma u programskom jeziku Python i analizu podataka o rezultatima nogometnih utakmica na Svjetskom prvenstvu u Katru 2022. Rezultati su prezentirani u obliku rang liste reprezentacija, a analizirani su faktori koji utječu na konačno rangiranje. Iako je PageRank algoritam prvotno razvijen za rangiranje web stranica, ovaj rad pokazuje njegovu primjenjivost i u drugim područjima kao što je sportsko rangiranje. Zaključci rada ukazuju na važnost ovog algoritma u objektivnom ocjenjivanju uspjeha sportskih ekipa te mogućnosti za daljnja istraživanja i unaprjeđenja metoda rangiranja u sportskom svijetu.

Ključne riječi: PageRank algoritam, Perron- Frobeniusov teorem, Markovljevi lanci, metoda potencija, rangiranje ekipa, nogomet

Sadržaj

1. Uvod

U današnjem svijetu informacija koji se velikom brzinom širi, pronalaženje relevantnih i kvalitetnih izvora postalo je izazovna zadaća. Kako se Internet razvijao, tako su se i javljale potrebe za pronalaženjem novih rješenja koji bi nam pomogli navigirati kroz bezbrojne stranice i izdvojiti one koje najbliže odgovaraju našim željama pretraživanja.

Danas postoji 30 do 50 milijardi stranica indeksiranih preko Google-a, no „samo“ 200 milijuna aktivnih web stranica. Google-ovi osnivači, Larry Page i Sergey Brin, su 1998. osmislili PageRank algoritam. PageRank algoritam je izum koji je proizašao iz dubine digitalne svijesti, čija je svrha bila rangirati web stranice prema njihovoj važnosti. [1]

PageRank algoritam je brzo stekao popularnost zbog svoje iznimne sposobnosti da pruži korisnicima precizne i relevantne rezultate pretraživanja. U kombinaciji s drugim faktorima, kao što su sadržaj stranice i ključne riječi, PageRank je postao središnji dio Google-ovog pretraživačkog iskustva.

U teorijskom dijelu ovog rada detaljno će se objasniti metode i tehnike koje stoje iza PageRank algoritma, uključujući koncept Markovljevih lanaca, teorem Perron-Frobenius i njegova primjena. A nakon toga u praktičnom dijelu rada PageRank algoritam će biti implementiran u programskom jeziku Python i analizirat će se podatci o rezultatima nogometnih utakmica na Svjetskom prvenstvu u Katru 2022.

2. Metode i tehnike rada

U ovom radu će se koristiti PageRank algoritam za rangiranje nogometnih reprezentacija koje su sudjelovale na nogometnom prvenstvu u Katru 2022. PageRank je algoritam koji se koristi za ocjenjivanje važnosti web stranica na temelju strukture povezivanja između njih. Matematička pozadina PageRank algoritma će biti opisana kako bi se stekao temeljni uvid u način na koji se vrši rangiranje.

Za implementaciju PageRank algoritma i obradu podataka koristit ćemo programski jezik Python. Python nudi razne biblioteke i alate za manipulaciju podacima i matematičke operacije, kao što su NumPy, matplotlib.

Koristit ćemo Excel kao izvor podataka svih utakmica i statistike po reprezentacijama na Svjetskom nogometnom prvenstvu 2022. u Katru.

3. Metode i PageRank algoritam

PageRank algoritam je neizostavni dio Googleove metode rangiranja web stranica u rezultatima pretraživanja. Koristi mrežu veza između stranica kako bi odredio njihovu vrijednost. U ovom poglavlju ćemo najprije detaljno objasniti matematičku pozadinu iza PageRank algoritma. Objasnit ćemo dvije metode, metodu potencija i metodu koja se svodi na rješavanje sustava linearnih jednadžbi. Zatim ćemo implementirati PageRank algoritam na stvarnom primjeru rangiranja reprezentacija na Svjetskom prvenstvu u nogometu koje se održalo u Kataru 2022. godine.

3.1. Metoda potencija

Metoda potencija je jednostavan, ali učinkovit pristup za određivanje najveće (po apsolutnoj vrijednosti) svojstvene vrijednosti i pripadnog svojstvenog vektora matrice. Korištenjem ove metode, možemo dobiti vrijedne uvide u strukturu i ponašanje matrice, omogućujući nam da donesemo informirane odluke i riješimo različite probleme u različitim područjima.[2][3]

Metoda potencija je posebno korisna za računanje približne vrijednosti najveće svojstvene vrijednosti po apsolutnoj vrijednosti i odgovarajućeg svojstvenog vektora. Međutim, moguće je modificirati algoritam kako bismo dobili i druge svojstvene vrijednosti. Metoda potencija se fokusira na dominatnu svojstvenu vrijednost jer iterativno umnožavanje matrice sa vektorom u svakoj iteraciji pojačava utjecaj najveće svojstvene vrijednosti. Ostale svojstvene vrijednosti, koje su manje od dominante, postupno gube utjecaj tijekom iteracija. U nastavku ćemo objasiti algoritam metode potencija.[2][3]

3.1.1. Algoritam metode potencija

Metoda potencija je iterativna metoda koja se može koristiti kada:

- Martica A reda n ima n linearno neovisnih svojstvenih vektora
- Svojstvene vrijednosti mogu se poredati po apsolutnoj vrijednosti $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$

Svojstveni vektor kvadratne matrice A je nenulti vektor x takav da za određeni broj λ vrijedi sljedeće:

$$Ax = \lambda x, \qquad za\ x \neq 0$$

Pri čemu λ zovemo svojstvenom vrijednošću.[2][3]

Algoritam 1. pronalazi po modulu najveću svojstvenu vrijednost zadane matrice, te pripadni svojstveni vektor.

Ulaz: $A \in C^{n\times n}$ takva da vrijedi $|\lambda_1| > |\lambda_2|$

Izlaz: $s^{(k)}, \lambda^{(k)}$ takvi da vrijede $s^{(k)} \approx x_1$, $\lambda^{(k)} \approx \lambda_1$

1. Odaberite $z^{(0)} \in C^n$ takav da je $\left|\left|z^{(0)}\right|\right|_2 = 1$
2. $\boldsymbol{for}\ k = 1,2,3,\ldots\ \boldsymbol{do}$
3. $w^{(k)} = Az^{(k-1)}$
4. $z^{(k)} = \frac{w^{(k)}}{\left|\left|w^{(k)}\right|\right|_2}$
5. $\lambda^k = \langle z^{(k)}, Az^{(k)}\rangle$
6. $\boldsymbol{if}\ ||Az^{(x)} - \lambda^{(k)}z^{(k)}||/||\lambda^{(k)}|| < \mathrm{tol}\ \boldsymbol{break}$
7. $\boldsymbol{end\ for}$.[2]

Razmotrimo svojstva konvergencije metode potencija. Ako se matrica A može dijagonizirati tada postoje n nezavisna svojstvena vektora matrice A. Neka su $x_1,\ldots,x_n$ ti svojstveni vektori, tada $x_1,\ldots,x_n$ čine bazu prostora R^n. Stoga se početni vektor $q^{(0)}$ može zapisati kao:

$$q^{(0)} = a_1x_1 + a_2x_2 + \ldots a_nx_n$$

Gdje su $a_1, \ldots, a_n$ skalari te množenjem obje strane jednadžbe s matricom A^k dobivamo:

$$\begin{aligned} A^kq^{(0)} &= A^k(a_1x_1 + a_2x_2 + \cdots + a_nx_n) = a_1A^kx_1 + a_2A^kx_2 + \cdots + a_nA^kx_n \\ &= a_1\lambda_1^kx_1 + a_2\lambda_2^kx_2 + \ldots + a_n\lambda_n^kx_n = a_1\lambda_1^k\left(x_1 + \sum_{j=2}^{n}\frac{a_j}{a_1}\left(\frac{\lambda_j}{\lambda_1}\right)^k x_j\right) \end{aligned}$$

Ako je $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$ za λ_1 kažemo da je dominantna svojstvena vrijednost. U tom slučaju $\left(\frac{\lambda_j}{\lambda_1}\right)^k \to 0$ i onda ako je $a_1 \neq 0, A^kq^{(0)} \to a_1\lambda_1^k$. Metoda potencija normalizira produkte $Aq^{(k-1)}$ kako bi izbjegla prelijevanje ili podlijevanje vrijednosti, stoga konvergira prema vektoru x_1 (pretpostavljajući da ima jediničnu normu). [2][3]

Da bi metoda potencija konvergirala, ključno je da svojstvena vrijednost λ_1 bude najveća po apsolutnoj vrijednosti među svim svojstvenim vrijednostima matrice A. Osim toga, početni vektor $q^{(0)}$ mora imati nenultu komponentu u smjeru svojstvenog vektora x_1 koji je povezan s λ_1. Ovi uvjeti osiguravaju da metoda potencija konvergira prema dominantnom svojstvenom vektoru.[2][3]

Praktična korisnost metode potencija ovisi o omjeru $|\lambda 2|/|\lambda 1|$, koji određuje brzinu konvergencije. Potencijalni problem nedostatne usklađenosti $q^{(0)}$ s $x_1(a_1 = 0)$ nije značajan jer, kada se $q^{(0)}$ nasumično odabere, vjerojatnost za to je nula. Nadalje, prisutnost zaokruživanja pogrešaka tijekom iteracijskog procesa obično osigurava da će kasniji vektori $q^{(k)}$ imati komponentu u smjeru x_1. [2][3]

Kada se metoda potencija konvergira prema dominantnom svojstvenom vektoru nakon k iteracija, izraz $\left[q^{(k)}\right]^T A q^{(k)} \approx \left[q^{(k)}\right]^T \lambda q^{(k)} = \lambda\left[q^{(k)}\right]^T q^{(k)} = \lambda\left\|q^{(k)}\right\|^2 = \lambda$ $(\left\|q^{(k)}\right\|^2 = 1$ jer $q^{(k)}$ je normaliziran u svakoj iteraciji). [2][3]

Primijetimo da u svakoj iteraciji se provodi samo jedno množenje matrice s vektorom (složenost $O(n^2)$). Nikada ne koristimo množenje matrica, koje zahtijeva više operacija (složenost $O(n^3)$). Kada je matrica A rijetka (sadrži samo nekoliko nenultih elemenata), množenje matrice s vektorom se može efikasno izvršiti. Stoga je metoda potencija praktična čak i ako je n veličina matrice jako velika, kao što je slučaj s Googleovim PageRank algoritmom. [2][3]

Metoda potencija ne konvergira za rotacijske matrice jer imaju svojstvene vrijednosti koje su jednake po apsolutnoj vrijednosti ($|\lambda_1| = |\lambda_2|$). Ovo se može izraziti formulom:

$$|\lambda|^2\|x\|^2 = \|\lambda x\|^2 = \|Ux\|^2 = \|x^{\mathsf{T}} U^{\mathsf{T}} U x\|^2 = \|x^{\mathsf{T}} x\|^2 = \|x\|^2$$

gdje λ označava svojstvenu vrijednost, x označava svojstveni vektor, a U predstavlja rotacijsku matricu. Iz ove formule slijedi da je apsolutna vrijednost svojstvene vrijednosti uvijek jednaka 1. Kako rotacijske matrice obično imaju barem dvije različite svojstvene vrijednosti s apsolutnom vrijednošću 1, metoda potencija neće konvergirati jer neće postojati dominirajuća svojstvena vrijednost koju bi se moglo izdvojiti iteracijama. [2] [3]

3.2. Metoda koja se svodi na rješavanje sustava linearnih jednadžbi

Druga metoda za izračunavanje PageRank vrijednosti svodi se na rješavanje sustava linearnih jednadžbi. Ovaj pristup koristi matematičku reprezentaciju grafa web stranica i linearnu algebru za izračunavanje PageRanka. Matematički, PageRank se može izraziti kao rješenje sustava linearnih jednadžbi koje uključuju veze između web stranica.

3.2.1. Markovljevi lanci

Markovljevi lanci su slučajni procesi u kojima buduće stanje ovisi samo o trenutnom stanju. Nazvani su po matematičaru Andreju Markovu, koji je proučavao ove lance i objavio rezultate o njima 1906. godine. Markovljevi lanci su jedna vrsta stohastičkih procesa, što znači da se sastoje od skupa slučajnih varijabli. Markovljevi lanci imaju Markovljevo svojstvo, što znači da je buduće stanje uvjetovano samo trenutnim stanjem. Oni mogu imati diskretan skup stanja (u diskretnom vremenu) ili kontinuiran skup stanja (u kontinuiranom vremenu).[4]

Ovi lanci mijenjaju svoja stanja u diskretnim točkama, bilo da je to diskretno ili kontinuirano prema parametru. Markovljevi lanci se koriste ne samo u matematici, već imaju značajnu primjenu i u tehničkim i društvenim znanostima. Oni predstavljaju važan model za proučavanje slučajne evolucije u različitim područjima. Markovljev lanac je vrsta slučajnog procesa u kojem buduće stanje sustava ovisi samo o trenutnom stanju, a ne o prethodnim stanjima. Matematički se to izražava pomoću prijelaznih vrijednosti, koje predstavljaju vjerojatnost prijelaza iz jednog stanja u drugo. Prijelazne vrijednosti se obično prikazuju pomoću matrice prijelaznih vrijednosti, koja se označava kao $P = [p_{ij}]$. [5]

Matrica prijelaznih vrijednosti P je kvadratna matrica dimenzija $k \times k$, gdje je k broj mogućih stanja u Markovljevom lancu. Element p_{ij} matrice P predstavlja vjerojatnost prijelaza iz stanja j u stanje i. Ova matrica omogućuje modeliranje ponašanja Markovljevog lanca i predviđanje budućih stanja sustava. Analizom ove matrice moguće je izračunati stacionarnu raspodjelu vjerojatnosti, odnosno dugoročne vjerojatnosti za svako stanje u lancu.[5]

Kroz primjenu matrice prijelaznih vrijednosti i koncepta Markovljevog lanca, moguće je modelirati i analizirati širok spektar sustava i fenomena, poput vremenskih serija, financijskih tržišta, bioloških sustava i mnogih drugih. Markovljev lanac može biti predstavljen kao sustav s diskretnim skupom stanja, koji može biti konačan ili beskonačan. Diskretna stanja $(n \geq 0)$ $S = \{0, 1, 2, 3, \dots k\}$ koja su određena X_n. Ako je $X_n = i$ onda se Markovljev lanac nalazi u stanju i u n-tom koraku.[5]

Postoji opći zapis Markovljevog lanca:

Lanac $X_1, X_2, X_3 \cdots X_n$ je Markovljev, ako za sva stanja $i_1, i_2, i_3, \cdots i_n$ vrijedi:

$$P(X_n + 1 = j | X_n = i_n, \dots, X_0 = i_0) = P(X_n + 1 = j | X_n = i_n)$$

Ova formula označava Markovljevo svojstvo. U ovom svojstvu $n + 1$ je buduće stanje dok je n trenutno ili sadašnje stanje, a prošla stanja su označena $0, 1, 2, \dots n - 1$.[5]

Prijelazna vjerojatnost, označena kao $P(X_n + 1 = j | X_n = i)$, predstavlja vjerojatnost prijelaza iz jednog stanja i u drugo stanje j. Ova vjerojatnost ovisi o trenutnom stanju i i parametru n, koji predstavlja korak u Markovljevom lancu. Kako se vrijednost parametra n mijenja, mijenja se i vjerojatnost P, pri čemu P ovisi o vrijednosti parametra n. Ako Markovljev lanac ima m stanja, tada imamo $m \times m$ mogućih vjerojatnosti. To znači da za svako trenutno stanje imamo m mogućih budućih stanja. Matrica vjerojatnosti, $P = \left[p_{i_j}\right]$, koristi se kako bismo odredili vjerojatnosti stanja u n-tom koraku. Važno je da su sve vrijednosti $p_{i_j} \geq 0$, jer vjerojatnosti moraju biti nenegativne. Matrica vjerojatnosti omogućuje nam da analiziramo ponašanje Markovljevog lanca, predvidimo buduća stanja i izračunamo vjerojatnosti prijelaza između stanja. Ovisno o konkretnom problemu, matrica vjerojatnosti može biti definirana na različite načine, uzimajući u obzir specifičnosti sustava koji se modelira.[6]

3.2.2. Primjer raspodjele posjetitelja

Pretpostavimo da imamo dvije e-commerce stranice koje drže sav promet proizvoda X. Primjećujemo da 80% posjetitelja koji dolaze na e-commerce stranicu S ostaje na istoj stranici, dok se preostalih 20% preusmjerava na e-commerce stranicu T. S druge strane, 60% posjetitelja e-commerce stranice B odlazi na e-commerce stranicu A, dok 40% posjetitelja ostaje na stranici B.

Koristit ćemo S_k i T_k kako bismo označili broj posjetitelja obje stranice u danu k. Dakle, broj posjetitelja na stranici S koji ostaje je jednak 80% broja S_k i 40% T_k.

$$S_{k+1} = 0.8S_k + 0.4T_k$$

$$T_{k+1} = 0.2S_k + 0.6T_k$$

$$x_{k+1} = Ax_k = \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix} x_k$$

Ova matrica ima nekoliko specifičnih svojstava. Prvo, svaki element predstavlja vjerojatnost da se posjetitelj koji je došao na jednu lokacije ode na drugu lokaciju. Na primjer, postoji 80% vjerojatnost da će posjetitelj na stranici S ostati na stranici S, što objašnjava vrijednost 0.8 u gornjem lijevom kutu matrice. Dakle, vrijednosti u matrici su između 0 i 1, što označava vjerojatnost ostanka posjetitelja sa web stranice.

Također, važno je napomenuti da svaki posjetitelj koji je došao na jednu stranicu može otići na drugu stranicu ukoliko mu nešto ne odgovara. Na primjer, kada se 80% posjetitelja ostane na stranici S to znači da 20% ode na stranicu T. Ova karakteristika osigurava da zbroj vjerojatnosti u svakom stupcu matrice bude jednak 1.

3.2.3. Perron-Frobeniusov teorem

Perron-Frobeniusov teorem, formuliran od strane matematičara Oskara Perrona i Georga Frobeniusa, značajno je otkriće u linearnoj algebri. Teorem tvrdi da za realnu kvadratnu matricu s pozitivnim elementima postoji jedinstvena najveća stvarna svojstvena vrijednost i odgovarajući svojstveni vektor može se odabrati da ima strogo pozitivne komponente. Ovaj teorem se također proširuje na određene klase nenegativnih matrica, pružajući važne uvide u njihova svojstva i ponašanje.[7]

U kontekstu problema rangiranja, Perron-Frobeniusov teorem igra ključnu ulogu jer nam pruža uvjete koji osiguravaju postojanje rješenja za problem. Ovi uvjeti su povezani s karakteristikama matrice A. Bitno je imati određene osobine koje nam omogućuju da, uz pretpostavke o međusobnim odnosima sudionika, pronađemo prikladno uređenje ili rangiranje unutar skupa. Perron-Frobeniusov teorem je klasično "strukturno složen" teorem koji uključuje tehničke definicije matrica i može se izraziti na različite, ali ekvivalentne načine. Jednostavan oblik teorema može se sažeti na sljedeći način:

„Teorem 2 (Perron-Frobenius) Ako matrica A ima nenegativne elemente, tada postoji svojstveni vektor r s nenegativnim elementima koji odgovara pozitivnoj svojstvenoj vrijednosti ρ. Nadalje, ako je matrica A neprekidna, svojstveni vektor r ima strogo pozitivne elemente, a odgovarajuća svojstvena vrijednost ρ je jedinstvena i jednostavna te je najveća svojstvena vrijednost matrice A prema apsolutnoj vrijednosti.“[8]

Vektor ili matrica koja sadrži samo nenegativne vrijednosti naziva se nenegativnim vektorom ili matricom (označavamo $v \geq 0$ da bismo naglasili da je v nenegativni vektor). Slično tome, vektor ili matrica s isključivo pozitivnim vrijednostima može se nazvati pozitivnim vektorom ili pozitivnom matricom (označavamo $v > 0$ da bismo naglasili da je v pozitivni vektor).[8]

Ponovimo da ove definicije omogućuju uspostavu djelomičnog redoslijeda među nenegativnim vektorima, gdje vrijedi da su, za dva vektora v i u, $v \geq u$ kada god je vektor dobiven oduzimanjem u od vektora veći ili jednak nula, a $v > u$ kada god je taj razlika strogo pozitivna. Treba napomenuti da taj redoslijed nije potpuni jer postoje parovi vektora koji nisu usporedivi, odnosno nije moguće utvrditi niti $v \geq u$ niti $u \geq$.[8]

Teorem se često iznosi u dvije tvrdnje, jednu za pozitivne matrice i drugu za nenegativne matrice. Prva forma potječe od Perrona, dok je Frobenius, nastojeći proširiti teorem na slučaj nenegativnih matrica, došao do druge forme, u kojoj je teza također istinita, ali uz dodatnu pretpostavku ireducibilnosti matrice.[8]

„Teorem 3 (Perronov teorem za pozitivne matrice) Neka je A matrica dimenzija $n \times n$ s pozitivnim elementima. Tada vrijede sljedeće tvrdnje.

(i) Postoji svojstvena vrijednost ρ (tako zvana Perron-Frobeniusova svojstvena vrijednost), koja je realna i pozitivna, i za bilo koju drugu svojstvenu vrijednost λ vrijedi $|\lambda| < \rho$.

(ii) ρ je jednostavna svojstvena vrijednost, odnosno jednostavni korijen karakterističnog polinoma. Drugim riječima, njezina algebraička multiplikacija je jedan. Kao posljedica toga, prostor svojstvenih vektora $V\rho$ povezan s ρ je jednodimenzionalan

(iii) Postoji pozitivan svojstveni vektor r povezan s ρ. Odnosno, postoji pozitivan lijevi svojstveni vektor s.

(iv) Nema drugih pozitivnih svojstvenih vektora matrice A, osim (pozitivnih) višekratnika vektora r (odnosno lijevih svojstvenih vektora osim (pozitivnih) višekratnika vektora s).

(v) $\lim_{k\to+\infty}\left(\frac{A}{\rho}\right)^k = rs^T$ pri čemu su desni i lijevi svojstveni vektori normalizirani tako da je $s^T r = 1$. Osim toga matrica rs^T je projekcija na prostor svojstvenih vektora $V\rho$, tako zvana Perronova projekcija.

(vi) Perron-Frobeniusova svojstvena vrijednost ρ zadovoljava nejednakosti $\min_i \sum_j a_{ij} \leq \rho \leq \max_i \sum_j a_{ij}$

“[8]

„Teorem 4 (Frobeniusov teorem za nenegativne ireducibilne matrice) Neka je A nenegativna ireducibilna matrica dimenzija n × n s periodom p i spektralnim radijusom ρ. Tada vrijede sljedeće tvrdnje:

(i) ρ je pozitivan i on je svojstvena vrijednost matrice A, nazvana Perron-Frobeniusova svojstvena vrijednost.

(ii) ρ je jednostavan. Svojstveni prostori desno i lijevo, povezani s ρ, su jednodimenzionalni.

(iii) Matrica A ima svojstveni vektor r i lijevi svojstveni vektor s povezane s ρ, čije su komponente pozitivne i jedini svojstveni vektori s pozitivnim komponentama su oni povezani s ρ.

(iv) Matrica A ima točno p (period) kompleksnih svojstvenih vrijednosti s modulom ρ. Svaka od njih je jednostavan korijen karakterističnog polinoma i proizvod je ρ s p-tim kompleksnim korijenom jedinice.

(v) $\lim_{k\to+\infty}\left(\frac{A}{\rho}\right)^k = rs^T$, gdje su desni i lijevi svojstveni vektori normalizirani tako da je $s^T r = 1$. Osim toga matrica rs^T je projekcija na prostor svojstvenih vektora $V\rho$, tako zvana Perronova projekcija.

(vi) Perron-Frobeniusova svojstvena vrijednost ρ zadovoljava nejednakosti $\min_i \sum_j a_{ij} \leq \rho \leq \max_i \sum_j a_{ij}$“[8]

3.2.4.Primjena Perron-Frobeniusa u PageRank algoritmu

Graf je skup povezanih čvorova te svaku mrežu možemo opisati uz pomoć grafa.Svaka web stranica predstavlja jedan čvor unutar grafa. Ukoliko postoji usmjereni brid odnosno veza između stranice X i stranice Y možemo reći da stranica X sadrži link koji nas odvodi na stranicu Y. Matrica susjedstva $\boldsymbol{A}$ ovog grafa još se zove web matrica. Matrica susjedstva je $\boldsymbol{n} \times \boldsymbol{n}$ matrica s elementima $\boldsymbol{A_{ij}} = \mathbf{1}$ ako postoji veza između tih stranica. Normalizirana web matrica, nazvana još Markovljeva matrica $\boldsymbol{A}$, dobiva se djeljenjem svakog stupca s ukupnim brojem 1 u tom stupcu. Recimo da imamo web graf s 4 stranice, označene s A, B, C. Matrica susjedstva bi izgledala ovako.[9]

$$\begin{matrix} \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \end{matrix}$$

Ukupni broj veza po retcima je 2, 1, 1. Zatim podijelimo svaki element u stupcu sa odgovarajućim brojem veza:

$$\begin{matrix} \mathbf{0/2} & \mathbf{1/1} & \mathbf{1/1} \\ \mathbf{1/2} & \mathbf{0/1} & \mathbf{0/1} \\ \mathbf{1/2} & \mathbf{0/1} & \mathbf{0/1} \end{matrix}$$

Konačno, dobijemo normaliziranu Markovljevu matricu (A) u kojoj su stupci normalizirani tako da zbroj elemenata u svakom stupcu bude 1:

$$\begin{matrix} \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0.5} & \mathbf{0} & \mathbf{0} \\ \mathbf{0.5} & \mathbf{0} & \mathbf{0} \end{matrix}$$

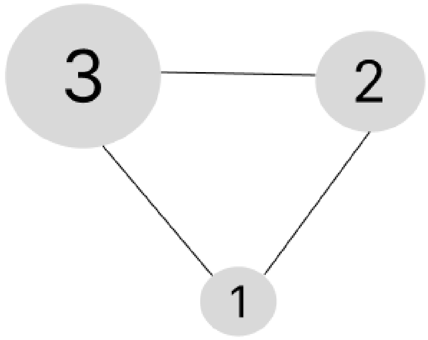
Ovo je rezultat normalizacije matrice za primjer s tri web stranice

„Googleova matrica je matrica $\boldsymbol{G} = \boldsymbol{dA} + (\mathbf{1} - \boldsymbol{d})\boldsymbol{E}$, gdje $\boldsymbol{je}\ \mathbf{0} < \boldsymbol{d} < \mathbf{1}$ parametar koji se naziva faktor prigušenja, a $\boldsymbol{A}$ je Markovljeva matrica dobivena iz matrice susjedstva skaliranjem redaka kako bi postali stohastičke matrice. Ovo je $\boldsymbol{n} \times \boldsymbol{n}$ Markovljeva matrica sa svojstvenom vrijednošću 1.“[9]

„Perron-Frobeniusov svojstveni vektor v, skaliran tako da je najveća vrijednost 10, naziva se Pageov rang za faktor prigušenja d.“[9]

Jednadžba za PageRank algoritam je:

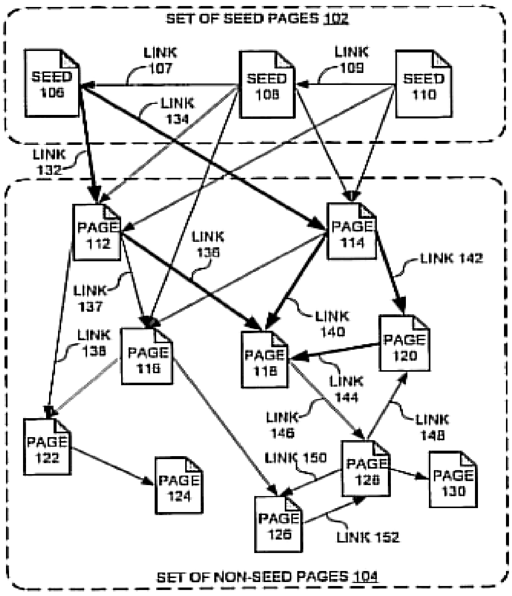
$$[\boldsymbol{dA} + (\mathbf{1} - \boldsymbol{d})\boldsymbol{E}]\boldsymbol{v} = \boldsymbol{v}$$



Slika 1: Graf web stranica (izrađena je od strane autora)

3.3. Google PageRank algoritam

PageRank algoritam su osmislili Larry Page i Sergey Brin. Njih dvojica dok su bili na Stanfordu su napisali rad „The PageRank Citation Ranking: Bringing Order to the Web.“ Rad je objavljen 1999. godine te demonstrira jednostavan algoritam za ocjenjivanje ranga stranica na webu.[10]



Slika 2: Link Graph Structure of Web Pages (Izvor: Dixon Jones, "Google PageRank Explained for SEO Beginners," May 16, 2023. Dostupno na: https://www.searchenginejournal.com/google-pagerank/483521/. Pristupljeno 23.6.2023.)

Algoritam za rangiranje web stranica slijedi niz koraka kako bi procijenio i dodijelio važnost različitim stranicama unutar mreže. Početno, algoritam prima skup međusobno povezanih web stranica. Zatim, dolazi faza u kojoj se dobiva skup „seed“ stranica. Ove sjemenske stranice su posebne jer sadrže izlazne veze prema stranicama u prvobitnom skupu. [10]

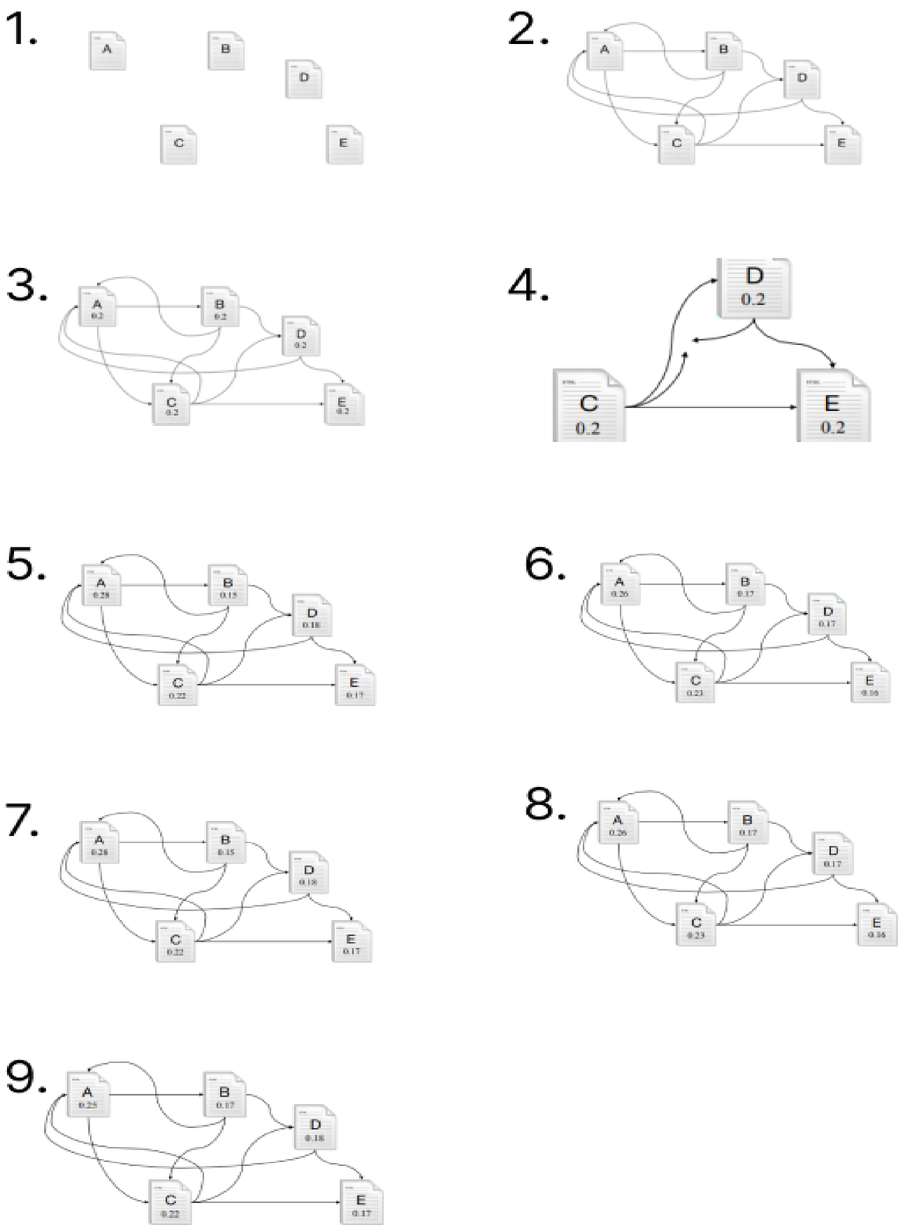
Nakon toga, svakoj vezi unutar mreže dodjeljuju se duljine temeljem svojstava povezanih stranica. Ova dodjela duljina temelji se na analizi karakteristika pojedinih stranica i njihovih veza. Dalje, algoritam izračunava najkraće udaljenosti od svake „seed“ stranice prema svakoj stranici unutar početnog skupa. Ove najkraće udaljenosti temelje se na duljinama veza koje su prethodno dodijeljene. [10]

Koristeći ove najkraće udaljenosti, svakoj stranici dodjeljuju se rangirajući bodovi. Ti rangirajući bodovi odražavaju važnost svake stranice unutar mreže, pri čemu se stranice koje su bliže „seed“ stranicama obično ocjenjuju višim bodovima. Na kraju, koristeći dobivene rangirajuće bodove, generira se konačno rangiranje svih stranica unutar mreže. Ovaj završni korak rezultira poretkom stranica prema njihovoj procijenjenoj važnosti. [10]

Ovaj algoritam pruža sistematičan način procjene važnosti i rangiranja web stranica unutar mreže temeljem njihovih veza i udaljenosti od početnih sjemenskih stranica. [10]

Algoritam PageRank dodjeljuje svakoj stranici ocjenu njezine važnosti koristeći rekurzivnu definiciju, pri čemu stranica postaje važna ako joj važne stranice daju linkove. Ova rekurzivna priroda znači da važnost jedne stranice ovisi o važnosti stranica koje joj daju linkove. Korisna usporedba za razumijevanje PageRank algoritma je zamisliti nasumičnog internetskog posjetitelja koji slijedi linkove s jedne stranice na drugu. PageRank određene stranice otprilike odgovara vjerojatnosti da će nasumični posjetitelj završiti na toj stranici. Budući da važne stranice obično imaju više dolaznih linkova, nasumični posjetitelj ima veću vjerojatnost da će završiti na takvim stranicama. Ponašanje nasumičnog posjetitelja predstavlja primjer Markovljevog procesa, koji je vrsta nasumičnog evolucijskog procesa koji ovisi samo o trenutnom stanju sustava, a ne o njegovoj povijesti.[10]

Googleov nasumični surfer predstavlja primjer Markovljevog procesa u kojem sustav koristi vjerojatnosti za prelazak iz jednog stanja (web stranice) u drugo. Ova vjerojatnost se temelji na broju veza koje povezuju web stranice i određuje kako nasumični surfer putuje kroz mrežu web stranica. [10]



Slika 3. Algoritam za inicijalno rangiranje web stranica (Izvor: Eric Roberts, „The Google PageRank Algorithm“, November 9, 2016. Dostupno na: https://web.stanford.edu/class/cs54n/handouts/24-GooglePageRankAlgorithm.pdf. Pristupljeno: 23.6.2023)

Algoritam za inicijalno rangiranje web stranica započinje s prvim korakom u kojem se formira početni skup stranica (prikazano na slici pod brojem 1). Ovaj skup može biti proizvoljan ili unaprijed definiran. Sljedeći korak uključuje "pretragu" weba kako bi se utvrdila struktura veza između stranica (prikazano na slici pod brojem 2). Ovaj proces, često nazvan "crawling", omogućuje identificiranje veza između različitih web stranica.[11]

Nakon što su struktura veza i mreža stranica utvrđene, svakoj pojedinačnoj stranici dodjeljuje se početna vrijednost ranga (prikazano na slici pod brojem 3). Ova početna vrijednost obično je 1 podijeljeno s ukupnim brojem stranica u mreži (označeno s N). Na taj način, sve stranice počinju s jednakim inicijalnim rangom.[11]

Postupno ažuriranje rangiranja svake stranice obuhvaća sumiranje težina svih stranica koje imaju veze prema njoj, pri čemu se ta težina dijeli brojem veza koje potječu iz stranice koja je uputila link (prikazano na slici 4). U ovom trenutačnom primjeru, stranica E ima dva dolazna linka, odnosno linka koji vode prema njoj. Prvi dolazi sa stranice C, a drugi sa stranice D. Konkretno, stranica C doprinosi 1/3 svojeg trenutnog rangiranja stranici E, budući da postoji još dva linka koja također dolaze sa stranice C. Slično, stranica D daje doprinos od 1/2 svog trenutnog rangiranja stranici E. Na temelju ovih doprinosa, izračunava se novo rangiranje stranice E:

$$PR(E) = \frac{PR(C)}{3} + \frac{PR(D)}{2} = \frac{0.2}{3} + \frac{0.2}{2} = 0.17$$

Ako stranica (kao što je E u trenutnom primjeru) nema izlaznih veza, njezin rang će se jednakomjerno redistribuirati među ostalim stranicama u grafu. U ovom grafu, 1/4 ukupnog ranga stranice E je raspodijeljeno među stranicama A, B, C i D. Ovaj model temelji se na ideji da će korisnici nastaviti pretraživati ako dođu do "slijepe ulice" (prikazano na slici pod brojem [11]

Kako bismo dalje razumjeli PageRank algoritam, primijetimo sljedeće korake u njegovom procesu. Prvo, nakon što smo uspostavili inicijalne rangove svih stranica u grafu, slijedi primjena redistribucije. Ova faza uključuje redistribuiranje dijela rangova stranica koje nemaju izlazne veze među ostale stranice u grafu. Na primjer, ako stranica E u trenutačnom primjeru nema izlaznih veza, njen rang se ravnomjerno raspoređuje među ostalim stranicama u grafu, poput stranica A, B, C i D (prikazano na slici pod brojem 7).[11]

Nakon redistribucije, prelazimo na ponavljanje postupka dok se rangovi stranica ne stabiliziraju. Ovaj korak uključuje iterativno ažuriranje rangova stranica. Za svaku stranicu, njen novi rang izračunava se zbrajanjem udjela rangova svih stranica koje linkaju prema njoj, podijeljenih s brojem izlaznih veza tih referentnih stranica. Ovaj postupak osigurava da stranice koje su povezane s više drugih stranica dobiju veći rang, što odražava njihovu važnost i utjecaj unutar mreže (prikazano na slici pod brojem 8).[11]

Napokon, u praksi se dodaje faktor prigušenja kako bi se modeliralo ponašanje korisnika koji postupno prestaju s pretraživanjem. Ovaj faktor uzima u obzir da korisnici neće nastaviti beskonačno klikati kroz veze, već će se vjerojatno zaustaviti u nekom trenutku. Damping faktor djeluje kao način smanjenja težine pri prijenosu rangova stranica i time pridonosi realnijem modeliranju korisničkog ponašanja prilikom pretraživanja i kretanja po mreži (prikazano na slici pod brojem 9). [11]

4. Implementacija PageRank algortima

U narednom praktičnom dijelu istraživanja, primijenit ćemo PageRank algoritam kako bismo rangirali ekipe sudionike Svjetskog nogometnog prvenstva koje se održalo u Kataru 2022. godine. Kroz četiri ključna koraka, istražit ćemo rezultate utakmica, izraditi matricu povezanosti ekipa, primijeniti PageRank algoritam na tu matricu te na temelju interpretacije rezultata rangirati sudionike prvenstva. Ovaj praktični primjer pružit će uvid u upotrebu PageRank algoritma u sportskom kontekstu kako bi se objektivno procijenio uspjeh i rang timova.

4.1. Svjetsko nogometno prvenstvo u Katru 2022.

Svjetsko nogometno prvenstvo u Kataru 2022. predstavlja 22. izdanje ovog globalnog sportskog spektakla. Uzbudljivo natjecanje privuklo je 32 reprezentacije koje su se otisnule u borbu za titulu svjetskog prvaka. Ovaj put, na svjetskom prvenstvu u Kataru, posebno se istaknulo sudjelovanje čak 36 glavnih sudaca i 69 pomoćnih sudaca, te 24 VAR (Video Assistant Referee) suca, koji su zajedno doprinijeli pravednosti i točnosti suđenja. [12]

Jedno od najvažnijih obilježja ovog prvenstva je uvođenje povijesne promjene – prvi puta će sutkinje sudjelovati kao suci na prvenstvu za muškarce, čime se postavlja novi standard za rodnu ravnopravnost i inkluzivnost u svijetu sporta.[12]

Svjetsko prvenstvo 2022. godine održalo se na osam zapanjujućih stadiona diljem Katara, uključujući Stadion Lusail Iconic, Stadion Al Bayt, Stadion 974, Stadion Al Thumama, Stadion Education City, Stadion Ahmad bin Ali, Međunarodni stadion Khalifa i Stadion Al Janoub. [12]

Natjecanje je započelo s fazom grupne igre, gdje su 32 reprezentacije bile podijeljene u 8 skupina po 4 tima. Prve dvije reprezentacije iz svake grupe kvalificirale su se za osminu finala, a zatim su se putem četvrtfinala i polufinala probijale do velikog finala. Ovo natjecanje nije samo prilika za pokazivanje sportske izvrsnosti, već i za slavlje kulture, zajedništva i strasti prema nogometu na svjetskoj razini.[12]

4.2. Prikupljanje podataka o rezultatima utakmica

Podaci o rezultatima utakmica su prikupljeni putem SofaScore mobilne aplikacije, koja je služila kao izvor detaljnih informacija o svim odigranim utakmicama tijekom Svjetskog prvenstva. Nakon prikupljanja tih informacija, daljnji korak u analizi bio je stvaranje Excel datoteke kako bi se podaci mogli strukturirano obrađivati.

Excel datoteka koja je izrađena sadrži listu rezultata utakmica. . Ovaj popis uključuje podatke o timovima koji su igrali, rezultatu utakmice (pobjeda, poraz, neriješeno), penalima te fazi u kojoj je ta utakmica odigrana.

Ovi podaci iz Excel datoteke su esencijalni za implementaciju PageRank algoritma. Oni čine temelj matrice povezanosti ekipa i pružaju kvantitativne informacije o uspjesima svake reprezentacije tijekom turnira.

Korištenje Excel datoteke omogućilo je praktično organiziranje i manipuliranje ovim podacima kako bi se mogli uvesti u PageRank algoritam. Ovaj pristup je omogućio analitičko dublje razumijevanje uspjeha timova na temelju njihove igre i rezultata, pridonoseći sveobuhvatnom i informativnom istraživanju natjecanja.

Tablica 1. Rezultati utakmica (vlastita izrada)

Team 1	Team 2	Goals t	Goals t	Penalie	Penalti	Phase	MP
Qatar	Ecuador	0	2			Group	1
England	Iran	6	2			Group	1
Senegal	Netherlan	0	2			Group	1
United Sta	Wales	1	1			Group	1
Argentina	Saudi Arab	1	2			Group	1
Denmark	Tunisia	0	0			Group	1
Mexico	Poland	0	0			Group	1
France	Australia	4	1			Group	1
Marroco	Croatia	0	0			Group	2
Germany	Japan	1	2			Group	1
Spain	Costa Rica	7	0			Group	1
Belgium	Canada	1	0			Group	1
Switzerlan	Cameroon	1	0			Group	1
Uruguay	Korea Rep	0	0			Group	1
Portugal	Ghana	3	2			Group	1
Brazil	Serbia	2	0			Group	1
Wales	Iran	0	2			Group	1
Qatar	Senegal	1	3			Group	1
Netherlan	Ecuador	1	1			Group	1
England	United Sta	0	0			Group	1
Tunisia	Australia	0	1			Group	1
Poland	Saudi Arab	2	0			Group	1
France	Denmark	2	1			Group	1
Argentina	Mexico	2	0			Group	1
Japan	Costa Rica	0	1			Group	1
Belgium	Marroco	0	2			Group	1
Croatia	Canada	4	1			Group	1
Spain	Germany	1	1			Group	1
Cameroon	Serbia	3	3			Group	1
Korea Rep	Ghana	2	3			Group	1
Brazil	Switzerlan	1	0			Group	1
Portugal	Uruguay	2	0			Group	1
Netherlan	Qatar	2	0			Group	1
Ecuador	Senegal	1	2			Group	1
Wales	England	0	3			Group	1
Iran	United Sta	0	1			Group	1
Australia	Denmark	1	0			Group	1
Tunisia	France	1	0			Group	1
Poland	Argentina	0	2			Group	1
Saudi Arab	Mexico	1	2			Group	1
Croatia	Belgium	0	0			Group	1
Canada	Marroco	1	2			Group	1
Japan	Spain	2	1			Group	1
Costa Rica	Germany	2	4			Group	1
Ghana	Uruguay	0	2			Group	1
Korea Rep	Portugal	2	1			Group	1
Serbia	Switzerlan	2	3			Group	1
Cameroon	Brazil	1	0			Group	1
Netherlan	United Sta	3	1			Round of :	1
Argentina	Australia	2	1			Round of :	1
France	Poland	3	1			Round of :	1
England	Senegal	3	0			Round of :	1
Japan	Croatia	1	1	1	3	Round of :	1
Brazil	Korea Rep	4	1			Round of :	1
Marroco	Spain	0	0	3	0	Round of :	1
Portugal	Switzerlan	6	1			Round of :	1
Croatia	Brazil	1	1	4	2	Quarter-fi	1
Netherlan	Argentina	2	2	3	4	Quarter-fi	1
Marroco	Portugal	1	0			Quarter-fi	1
England	France	1	2			Quarter-fi	1
Argentina	Croatia	3	0			Semi-final	1
France	Marroco	2	0			Semi-final	1
Croatia	Marroco	2	1			Third plac	2
Argentina	France	3	3	4	2	Final	1

Tablica 2.Prikaz statistike po državama (vlastita izrada)

Rk	Squad	MP	W	D	L	GF	GA
1	ar Argenti	7	4	2	1	15	8
2	fr France	7	5	1	1	16	8
3	hr Croatia	7	2	4	1	8	7
4	ma Moroc	7	3	2	2	6	5
QF	nl Netherl	5	3	2	0	10	4
QF	eng Englar	5	3	1	1	13	4
QF	br Brazil	5	3	1	1	8	3
QF	pt Portuga	5	3	0	2	12	6
im	jp Japan	4	2	1	1	5	4
R16	sn Senega	4	2	0	2	5	7
R16	au Austral	4	2	0	2	4	6
R16	ch Switzer	4	2	0	2	5	9
R16	es Spain	4	1	2	1	9	3
R16	us United	4	1	2	1	3	4
R16	pl Poland	4	1	1	2	3	5
R16	kr Korea R	4	1	1	2	5	8
GR	de Germa	3	1	1	1	6	5
GR	ec Ecuado	3	1	1	1	4	3
GR	cm Camer	3	1	1	1	4	4
GR	uy Urugua	3	1	1	1	2	2
GR	tn Tunisia	3	1	1	1	1	1
GR	mx Mexic	3	1	1	1	2	3
GR	be Belgiur	3	1	1	1	1	2
GR	gh Ghana	3	1	0	2	5	7
GR	sa Saudi A	3	1	0	2	3	5
GR	ir IR Iran	3	1	0	2	4	7
GR	cr Costa R	3	1	0	2	3	11
GR	dk Denma	3	0	1	2	1	3
GR	rs Serbia	3	0	1	2	5	8
GR	wls Wales	3	0	1	2	1	6
GR	ca Canada	3	0	0	3	2	7
GR	qa Qatar	3	0	0	3	1	7

4.3. Težine temeljene na pobjedama i fazi odigravanja

Kako bi važnost utakmice utjecala na poredak, morali smo dodatne težine za faze odigravanja utakmica.

```
phase_weights = {
    "Round of 16": 0.1,
    "Quarter-final": 0.2,
    "Semi-final": 0.3,
    "Third place": 0.4,
    "Final": 0.5
}
```

Unutar grupne faze natjecanja, implementirali smo sustav dodjeljivanja težina kako bismo odrazili važnost svake utakmice i pravilno izračunali bodove za timove. U grupnoj fazi, timovi su zarađivali bodove prema standardnom sustavu bodovanja za nogometne utakmice: pobjeda donosi 3 boda, neriješeno 1 bod, a poraz 0 bodova. Ovaj sustav osmišljen je kako bi pravilno vrednovao uspjehe timova unutar njihovih grupa i omogućio napredovanje najboljim ekipama. Međutim, kako bi se odrazila važnost utakmica u kasnijim fazama natjecanja, dodali smo dodatnu težinu ("phase-weight"). Ovo povećava bodove koje timovi zarađuju tijekom natjecanja u fazama koje slijede grupnu fazu. To znači da su utakmice u kasnijim fazama, poput osmine finala, četvrtfinala i dalje, nagrađene s dodatnim bodovima, što potencijalno može značiti više za ukupni poredak timova.

4.4. Izrada matrice povezanosti ekipa

Matricu povezanosti smo izgenerirali uz pomoć koda:

```
teams_column = df_results[["Team 1", "Team 2"]]
unique_teams = pd.unique(teams_column.values.ravel())
num_teams = len(unique_teams)
team_dictionary = {team: i for i, team in enumerate(unique_teams)}
adjacency_matrix = np.zeros((num_teams, num_teams), dtype=int)

for i in range(0, len(df_results)):
team1 = df_results.iloc[i, 0]
team2 = df_results.iloc[i, 1]
goals1 = df_results.iloc[i, 2]
goals2 = df_results.iloc[i, 3]
```

```
penalties1 = df_results.iloc[i, 4]
penalties2 = df_results.iloc[i, 5]phase = df_results.iloc[i, 6]

if goals1 == goals2 and phase == "Group":
adjacency_matrix[team_dictionary[team1]][team_dictionary[team2]]= goals1
adjacency_matrix[team_dictionary[team2]][team_dictionary[team1]]= goals2

if goals1 == goals2 and phase != "Group":
goals1 = penalties1
goals2 = penalties2

adjacency_matrix[team_dictionary[team1]][team_dictionary[team2]]= goals1
adjacency_matrix[team_dictionary[team2]][team_dictionary[team1]]= goals2
pagerank_scores = pagerank(normalized_matrix)
```

Prvo se izdvajaju samo stupci "Team 1" i "Team 2" iz okvira podataka (DataFrame) nazvanog df_results, koji sadrži informacije o rezultatima sportskih utakmica. Ova dva stupca sadrže imena timova koji su sudjelovali u svakoj utakmici. Zatim se stvaraju dvije liste, unique_teams i team_dictionary, koje će se koristiti za praćenje jedinstvenih timova i njihovih indeksa u matrici povezanosti. unique_teams sadrži jedinstvena imena timova, dok team_dictionary mapira ime tima na odgovarajući indeks u matrici povezanosti. Nakon toga se inicijalizira matrica adjacency_matrix koja će se koristiti za pohranu rezultata utakmica između timova. Matrica se inicijalizira nulama. Slijedi petlja koja prolazi kroz svaki redak u okviru podataka df_results. U svakom koraku petlje se izvlače informacije o timovima, broju golova, broju penala i fazi natjecanja za trenutnu utakmicu. Prvi if uvjet provjerava jesu li brojevi golova za oba tima jednaki i je li utakmica u fazi "Group". Ako je to slučaj, rezultati se upisuju u matricu povezanosti za oba tima, pri čemu se broj golova koristi kao vrijednost u matrici. Sljedeći if uvjet obrađuje situaciju u kojoj su brojevi golova jednaki, ali utakmica nije u fazi "Group". U tom slučaju, rezultati penala zamjenjuju brojeve golova. Nakon toga se upisuju rezultati utakmice u matricu povezanosti za oba tima, uzimajući u obzir zamjene koje su se dogodile u prethodnom if uvjetu. Matrica povezanosti će nam poslužiti za dalje za izradu grafa kao i za PageRank algoritam.

Tablica 3.Prikaz statistike po državama (vlastita izrada)

	Katar	Ecuador	England	Iran	Senegal	Netherland
Katar	0	0	0	0	1	0
Ecuador	2	0	0	0	1	1
England	0	0	0	6	3	0
Iran	0	0	2	0	0	0
Senegal	3	2	0	0	0	0
Netherland	2	1	0	0	2	0
nited State	0	0	0	1	0	1
Wales	0	0	0	0	0	0
Argentina	0	0	0	0	0	4
audi Arabi	0	0	0	0	0	0
Denmark	0	0	0	0	0	0
Tunisia	0	0	0	0	0	0
Mexico	0	0	0	0	0	0
Poland	0	0	0	0	0	0
France	0	0	2	0	0	0
Australia	0	0	0	0	0	0
Marroco	0	0	0	0	0	0
Croatia	0	0	0	0	0	0
Germany	0	0	0	0	0	0
Japan	0	0	0	0	0	0
Spain	0	0	0	0	0	0
Costa Rica	0	0	0	0	0	0
Belgium	0	0	0	0	0	0
Canada	0	0	0	0	0	0
Switzerland	0	0	0	0	0	0
Cameroon	0	0	0	0	0	0
Uruguay	0	0	0	0	0	0
orea Repub	0	0	0	0	0	0
Portugal	0	0	0	0	0	0
Ghana	0	0	0	0	0	0
Brazil	0	0	0	0	0	0
Serbia	0	0	0	0	0	0

4.5. Primjena PageRank algoritma

Nakon što smo izradili matricu povezanosti potrebno je izraditi normalizaciju te matrice.

```
row_sums = adjacency_matrix.sum(axis=1)
normalized_matrix = adjacency_matrix / row_sums[:, np.newaxis]
```

Tablica 4. Isječak normalizirane matrice povezanosti (vlastita izrada)

	Katar	Ecuador	England	Iran	Senegal	letherland	nited State	Wales	Argentina
Katar	0	0	0	0	1	0	0	0	0
Ecuador	0.5	0	0	0	0.25	0.25	0	0	0
England	0	0	0	0.461538	0.230769	0	0	0.230769	0
Iran	0	0	0.5	0	0	0	0	0.5	0
Senegal	0.6	0.4	0	0	0	0	0	0	0
letherland	0.181818	0.090909	0	0	0.181818	0	0.272727	0	0.272727
nited State	0	0	0	0.333333	0	0.333333	0	0.333333	0
Wales	0	0	0	0	0	0	1	0	0
Argentina	0	0	0	0	0	0.222222	0	0	0
audi Arabi	0	0	0	0	0	0	0	0	0.666667
Denmark	0	0	0	0	0	0	0	0	0
Tunisia	0	0	0	0	0	0	0	0	0

Normalizacija matrice se računa tako što svaku ćeliju podijelimo s zbrojem cijelog reda. Sljedeće potrebno je napisati funkciju za PageRank:

```
def pagerank(normalized_matrix, num_iterations=1000, damping_factor=0.85):
    num_teams = normalized_matrix.shape[0]
    initial_rank = np.ones(num_teams) / num_teams
    damped_transition_matrix = damping_factor * normalized_matrix + (1 -
damping_factor) / num_teams
    print("Initial Rank:", initial_rank)
    print("Google matrix:", damped_transition_matrix)

    rank = initial_rank.copy()
    for iteration in range(num_iterations):
        rank = damped_transition_matrix @ rank  # Matrično množenje

    return rank
```

U ovom dijelu koda implementiran je algoritam PageRank, koji se koristi za rangiranje čvorova u usmjerenom grafu, a često se primjenjuje u kontekstu rangiranja web stranica na osnovu njihove povezanosti. Funkcija pagerank prihvaća normaliziranu matricu povezanosti kao ulaz, zajedno s opcionalnim parametrima za broj iteracija i faktor prigušenja. Evo detaljnog opisa ovog dijela koda. U početku se funkcija pagerank poziva s tri glavna parametra:

normalized_matrix, koja predstavlja normaliziranu matricu povezanosti između čvorova, num_iterations (zadana vrijednost je 1000), što predstavlja broj iteracija algoritma PageRank, i damping_factor (zadana vrijednost je 0.85), što predstavlja faktor prigušenja koji kontrolira koliko se vjerojatnosti prenosi iz jednog čvora u drugi. Prvo se dobiva broj timova (čvorova) u normaliziranoj matrici kako bi se inicijalizirala početna rang lista. Inicijalna rang lista nazvana initial_rank se postavlja tako da svaki tim ima početni rang jednako raspoređen, tj. svaki tim ima inicijalni rang jednak obrnutom broju timova. Zatim se stvara prigušena tranzicijska matrica damped_transition_matrix koristeći faktor prigušenja i normaliziranu matricu povezanosti. Ova matrica kombinira dvije komponente: prigušenje faktora kojim se smanjuje vjerojatnost prijelaza iz jednog čvora u drugi i doprinos koji dolazi od jednakog raspodjele vjerojatnosti na sve čvorove. Slijedi izlazna poruka koja ispisuje početnu rang listu (initial_rank) i prigušenu tranzicijsku matricu (damped_transition_matrix) za potrebe provjere i praćenja. Dalje se izvodi glavna petlja koja se ponavlja num_iterations puta, pri čemu se u svakoj iteraciji izračunava nova rang lista. To se postiže matričnim množenjem prigušene tranzicijske matrice s trenutnom rang listom, što simulira proces širenja vjerojatnosti kroz graf. Na kraju funkcija vraća konačnu rang listu, koja će sadržavati rangiranje čvorova (timova) na temelju algoritma PageRank.

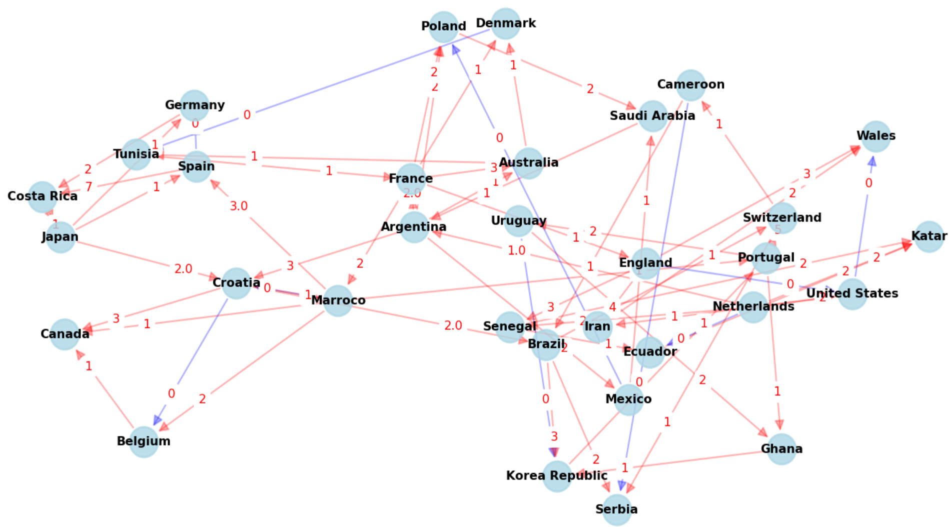
```
final_scores = {team: pagerank_scores[team_dictionary[team]] * weight for
team, weight in team_weights.items()}

sorted_teams = sorted(final_scores.items(),key=lambda x: x[1], reverse=True)

for team, weight in sorted_teams:
    print(f"Tim: {team}, Ukupna težina: {weight}")
```

Na kraju PageRank vrijednosti množimo s zadanim težinama i ispisujemo te vrijednosti.

4.6. Usmjereni graf

```
G = nx.DiGraph()

teams = set(df['Team 1']).union(set(df['Team 2']))
G.add_nodes_from(teams)

for index, row in df.iterrows():
    team1 = row['Team 1']
    team2 = row['Team 2']
    goals1 = row['Goals team 1']
```

```
    goals2 = row['Goals team 2']

    if goals1 > goals2:
        G.add_edge(team1, team2, weight=abs(goals1 - goals2))
    elif goals2 > goals1:
        G.add_edge(team2, team1, weight=abs(goals2 - goals1))
    else:
        if row['Phase'] == 'Group':
            G.add_edge(team1, team2, weight=0)
        else:
            G.add_edge(team1,  team2,  weight=abs(row['Penalties  team  1']  -
row['Penalties team 2']))

pos = nx.spring_layout(G, seed=42, k=1)
edge_labels = {(u, v): d['weight'] for u, v, d in G.edges(data=True)}
node_labels = {node: node for node in G.nodes()}

plt.figure(figsize=(15, 15))
node_sizes = [1000 for node in G.nodes()]
node_colors = ['lightblue' for node in G.nodes()]
edge_colors  =  ['red'  if  G.edges[e]['weight']  >  0  else  'blue'  for  e  in
G.edges()]

nx.draw_networkx_nodes(G,pos,   node_size=node_sizes,node_color=node_colors,
alpha=0.8)
nx.draw_networkx_edges(G,    pos,width=1.0,alpha=0.5,edge_color=edge_colors,
arrows=True)
nx.draw_networkx_labels(G,pos,font_size=8,font_color="black"font_weight="bo
ld")
nx.draw_networkx_edge_labels(G,  pos,  edge_labels=edge_labels,  font_size=8,
font_color="red",label_pos=0.3, rotate=False, bbox=dict(alpha=0.5))

plt.title("Usmjereni  graf  s  apsolutnom  razlikom  golova  i  strelicama  prema
pobjedniku", fontsize=16)
plt.axis("off")
plt.show()
```



Slika 4. Graf povezanosti (vlastita izrada, 10.8.2023)

U ovom dijelu koda stvoren je grafički prikaz rezultata sportskih utakmica koristeći usmjereni graf. Ovaj graf vizualizira pobjednike utakmica, pri čemu su strelice usmjerene prema pobjedniku, a težine na strelicama predstavljaju apsolutnu razliku u golovima ili broju penala između timova. Evo detaljnog opisa ovog dijela koda: Prvo se inicijalizira usmjereni graf G koristeći nx.DiGraph() iz NetworkX biblioteke. Ovaj graf će sadržavati čvorove koji predstavljaju timove iz sportskih utakmica. Zatim se stvara skup teams koji sadrži sve jedinstvene timove koji su sudjelovali u utakmicama. Čvorovi se dodaju u graf G koristeći G.add_nodes_from(teams), pri čemu svaki tim postaje čvor u grafu. Slijedi petlja koja prolazi kroz svaki redak u okviru podataka df, koji sadrži informacije o rezultatima utakmica. Za svaki redak se izvlače informacije o timovima, broju golova i broju penala. U ovisnosti o rezultatu utakmice (netko je pobijedio ili je bilo neriješeno), stvaraju se usmjereni bridovi u grafu. Ako je tim 1 pobijedio (više golova ili penala), brid se stvara između tima 1 i tima 2, pri čemu je težina brida jednaka apsolutnoj razlici u golovima ili penala. Ako je rezultat neriješen i utakmica je u fazi "Group", težina brida postavlja se na 0 kako bi se označilo da su timovi igrali neriješeno u grupnoj fazi. Ako je rezultat neriješen u nekoj drugoj fazi, težina brida se postavlja na apsolutnu razliku u broju penala. Nakon što su čvorovi i bridovi dodani u graf, definira se pozicija čvorova u grafu pomoću nx.spring_layout. Također se pripremaju oznake za čvorove i bridove.Nakon svih pripremnih koraka, slijedi crtanje grafa. Čvorovi su prikazani kao krugovi s određenim bojama, a bridovi su usmjereni i imaju boju ovisno o težini. Oznake čvorova i bridova dodane su na graf kako bi se jasno vidjele informacije o timovima i rezultatima.

4.7. Interpretacija rezultata i rangiranje ekipa

Analiza rezultata i rangiranje ekipa na temelju grafičke vizualizacije usmjerenog grafa s apsolutnom razlikom golova ili broja penala pruža nam uvid u relativne uspjehe različitih timova u sportskom natjecanju. Ukupni poredak ekipa, koji je izračunan na temelju težina bridova u grafu, pruža nam jasnu predodžbu o tome koje su ekipe ostvarile bolje rezultate. Prema dobivenim rezultatima, najbolje plasirane ekipe su Argentina, Francuska, Hrvatska i Maroko, redom, s najvišim ukupnim težinama. Argentina se ističe kao vodeći tim s ukupnom težinom od 0.596875. Ovo su četiri tima koja su se istaknula u natjecanju i postigla najbolje rezultate. Međutim, važno je napomenuti da su se dogodila odstupanja u poredku ekipa u usporedbi s stvarnim rezultatima. Naime, odstupanja se primjećuju u 23 slučaja od ukupno 32, što iznosi 71.87% odstupanja u poredku. Ova odstupanja uglavnom su sitna i događaju se zbog specifičnosti algoritma koji je generirao rezultate. Na primjer, zamjena mjesta između Portugala i Brazila, koji su imali iste vrijednosti ukupnih težina, može se smatrati slučajnim odstupanjem. Prva četiri mjesta u ukupnom poretku su ostala nepromijenjena i odražavaju stvarne uspjehe tih timova. S druge strane, odstupanja koja se događaju između 5. i 8. mjesta, kao i od 12. do 32. mjesta, sugeriraju da su neki timovi imali slične rezultate i težine, što je dovelo do manjih varijacija u njihovom poretku. U konačnici, ovi rezultati i odstupanja u rangiranju ekipa pružaju nam važan uvid u relativne uspjehe timova u natjecanju, ali također nas podsjećaju na važnost toga da algoritmi za rangiranje mogu imati svoje specifičnosti i ograničenja koja mogu utjecati na konačne rezultate. Stoga je važno koristiti takve analize kao osnovu za dublju analizu i donošenje informiranih odluka.

```
Tim: Argentina, Ukupna težina: 0.596875
Tim: France, Ukupna težina: 0.4875
Tim: Croatia, Ukupna težina: 0.459375
Tim: Marroco, Ukupna težina: 0.41562499999999997
Tim: England, Ukupna težina: 0.31562500000000016
Tim: Netherlands, Ukupna težina: 0.31562500000000004
Tim: Portugal, Ukupna težina: 0.28437499999999993
Tim: Brazil, Ukupna težina: 0.28437499999999993
Tim: Japan, Ukupna težina: 0.1875
Tim: Senegal, Ukupna težina: 0.1875
Tim: Australia, Ukupna težina: 0.1875
Tim: Switzerland, Ukupna težina: 0.18749999999999994
Tim: United States, Ukupna težina: 0.15625000000000006
Tim: Korea Republic, Ukupna težina: 0.12500000000000003
Tim: Ecuador, Ukupna težina: 0.125
Tim: Tunisia, Ukupna težina: 0.125
Tim: Mexico, Ukupna težina: 0.125
Tim: Poland, Ukupna težina: 0.125
Tim: Spain, Ukupna težina: 0.125
Tim: Belgium, Ukupna težina: 0.125
Tim: Germany, Ukupna težina: 0.125
Tim: Uruguay, Ukupna težina: 0.12499999999999997
Tim: Cameroon, Ukupna težina: 0.12499999999999997
Tim: Iran, Ukupna težina: 0.09375000000000004
Tim: Saudi Arabia, Ukupna težina: 0.09375
Tim: Costa Rica, Ukupna težina: 0.09375
Tim: Ghana, Ukupna težina: 0.09375
Tim: Wales, Ukupna težina: 0.03125000000000001
Tim: Denmark, Ukupna težina: 0.03125
Tim: Serbia, Ukupna težina: 0.031249999999999993
Tim: Katar, Ukupna težina: 0.0
Tim: Canada, Ukupna težina: 0.0
```

Slika 5. Vrijednosti PageRank algoritma (vlastita izrada, 10.8.2023.)

5. Zaključak

U ovom istraživanju smo se usredotočili na primjenu PageRank algoritma za rangiranje nogometnih reprezentacija koje su sudjelovale na Svjetskom nogometnom prvenstvu u Katru 2022. godine. PageRank algoritam, koji je prvobitno razvijen za rangiranje web stranica na temelju njihove povezanosti, pokazao se iznimno korisnim alatom i u drugim kontekstima, u ovom slučaju za ocjenjivanje uspjeha i važnosti nogometnih timova.

Ovaj rad započeo je teoretskim pregledom metoda koje su ključne za razumijevanje PageRank algoritma, uključujući metodu potencija, rješavanje sustava linearnih jednadžbi, Markovljeve lance i Perron-Frobeniusov teorem. Detaljno smo analizirali Google PageRank algoritam korak po korak kako bismo stekli temeljni uvid u njegovu matematičku pozadinu.

U praktičnom dijelu rada, implementirali smo PageRank algoritam koristeći programski jezik Python. Python je odabran zbog svoje fleksibilnosti i bogatih biblioteka, poput NumPy, koje su nam pomogle u manipulaciji podacima i matematičkim operacijama. Za izvor podataka koristili smo Excel tablicu koja je sadržavala informacije o svim utakmicama i statistici reprezentacija na Svjetskom nogometnom prvenstvu 2022. u Katru.

Rezultati našeg istraživanja su omogućili rangiranje nogometnih reprezentacija na temelju njihovih uspjeha u turniru. Pomoću PageRank algoritma, dobili smo ukupni poredak ekipa, pri čemu su se istaknule Argentina, Francuska, Hrvatska i Maroko kao najbolje rangirane reprezentacije.

Popis literature

[1] TechJury, "How Many Websites Are There? (Updated for 2021)," TechJury, dostupno na https://techjury.net/blog/how-many-websites-are-there/#:~:text=There%20are%2030%20to%2050%20billion%20web%20pages%20indexed%20through%20Google. Pristupljeno: 8.5.2023.

[2] I. Barunčić, "Iterativne metode i primjene," Repozitorij Sveučilište u Osijeku, 2018, dostupno na https://repozitorij.mathos.hr/islandora/object/mathos%3A239/datastream/PDF/view. Pristupljeno: 8.5.2023.

[3] A. Hajba, M. Marušić, B. Tadić, "Iterativne metode i primjene," Matematički Vesnik, vol. 39, br. 2, str. 178-195, 2007. [Online]. Dostupno na: http://e.math.hr/Vol39/Hajba_et_al. Pristupljeno 8.5.2023.

[4] W. Li, C. Zhang, "Markov Chain Analysis," in International Encyclopedia of Human Geography, 2009. [Online]. Dostupno: https://www.sciencedirect.com/topics/social-sciences/markovchain#:~:text=Markov%20Chain%20Analysis&text=Human%20Geography%2C%202009-,A%20Markov%20chain%20is%20a%20process%20that%20consists%20of%20a,one%20to%20study%20this%20process. Pristupljeno: 12.5.2023.

[5] "Markovljevi lanci," Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet. Dostupno na: https://www.grad.unizg.hr/_download/repository/Markovljevi_lanci.pdf. Pristupljeno: 12.5.2023.

[6] Z. Vondraček, "Poglavlje 2 – Primjeri Markovljevih lanaca," Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet. Dostupno na: http://web.math.pmf.unizg.hr/~vondra/ml-p2.pdf. Pristupljeno 13.5.2023.

[7] P. Huston, "What is the Perron-Frobenius theorem?," July 27, 2018. [Online]. Dostupno: https://math.osu.edu/sites/math.osu.edu/files/What_is_2018_Perron-Frobenius_Theorem.pdf. Pristupljeno: 20.6.2023.

[8] A. Peretti, "The importance of Perron-Frobenius Theorem in ranking problems," Working Paper Series. Department of Economics, University of Verona, prosinac 2014. Dostupno: http://dse.univr.it/home/workingpapers/wp2014n26.pdf.

[9] O. Knill, "Math 19b: Linear Algebra with Probability. Lecture 34: Perron Frobenius theorem," Harvard University. Dostupno na: https://people.math.harvard.edu/~knill/teaching/math19b_2011/handouts/lecture34.pdf. Pristupljeno 22.6.2023.

[10] Dixon Jones, "Google PageRank Explained for SEO Beginners," May 16, 2023. Dostupno na: https://www.searchenginejournal.com/google-pagerank/483521/. Pristupljeno 1.7.2023.

[11] E. Roberts, "The Google PageRank Algorithm," Handout #24, CS 54N, Stanford University, November 9, 2016. [Online]. Dostupno na: https://web.stanford.edu/class/cs54n/handouts/24-GooglePageRankAlgorithm.pdf. Pristupljeno 3.7.2023.

[12] FIFA. (Nedatirano). "FIFA World Cup Qatar 2022™." FIFA+ [Online]. Dostupno na: https://www.fifa.com/fifaplus/en/tournaments/mens/worldcup/qatar2022. Pristupljeno 6.8.2023.

Popis slika

Popis tablica