

# Glicko rating v aplikaciji eQuiz

**Matjaž Pogačnik**

*faculty of computer and information science  
večna pot 113  
1000 ljubljana*

**Abstract.** V spletni aplikaciji eQuiz se na podlagi reševanj nalog računa rating uporabnikov z uporabo Elo sistema, ki nam ne pove dosti o zanesljivosti izračunanega ratinga. Možno izboljšavo takega sistema predstavlja Glicko rating.

**Key words:** eQuiz, rating, Glicko, Elo

## 1 UVOD

Ena izmed funkcionalnosti aplikacije eQuiz je ocenjevanje uporabnikov in nalog preko Elo rating sistema. Uporabnik v kvizu premaguje naloge, na podlagi rezultatov pa se izračuna rating uporabnika in rating naloge, kot bi uporabnik igral proti nalogam šah. To nam lahko predstavlja dobro informacijo o uspešnosti uporabnikov, poleg tega pa zvemo nekaj več o morebitni težavnosti nalog iz strani uporabnikov. Za ta namen nam je na voljo več različnih rating sistemov. V nadaljnji raziskavi se primerja trenutno nameščen Elo rating sistem v eQuizu proti Glicko rating sistemu, ki uspešno rešuje probleme, ki se pojavijo v Elo rating sistemu. Osredotočili se bomo na problem zanesljivosti ratinga posameznega uporabnika.

Ker uporabniki poljubno tekom različnih časovnih intervalov rešujejo različno mnogo kvizov, bo njihov rating izračunan na podlagi različnih količin podatkov in bo posledično različno zanesljiv. Poleg tega je zbirka nalog že v osnovi lahko po velikosti precej drugačna od števila uporabnikov, zaradi česar bodo študenti in naloge prejeli precej drugače razporejena ocenjevanja. Glicko za ta namen uporabi pristop predstavitev ratinga kot intervala zaupanja, ki se z večjo aktivnostjo manjša, z neaktivnotjo pa večja.

Glicko rating sistem je leta 1995 objavil Mark Glickman, kot odziv na našteje pomankljivosti Elo sistema. Izpeljan je bil iz statističnega modela izidov pri šahu in s pomočjo matematičnih aproksimacij omogoča preproste izračune. Specifikacija sistema je javno dostopna [1], sistem pa je bil uporabljen v mnogih igrah kot so Pokémon Go, Lichess, Free Internet Chess Server, Chess.com, Online Go Server (OGS), Counter-Strike: Global Offensive, Quake Live, Team Fortress 2, Dota 2, Dota Underlords, Guild Wars 2 in tekmovanjih iz programiranja [4].

V članku je torej raziskan problem trenutnega rating sistema in obnašanje Glicko rating sistema na enakih podatkih. Skupaj s podatki je predstavljena metodologija izračunov (poglavje 3.1-3.2), v nadaljevanju pa je predstavljen Elo rating sistem in problematika (poglavje 3.3), ki jo rešujemo z Glicko ratingom (poglavje 3.4). V zaključku so predstavljene dodane funkcionalnosti (poglavje 4-5) pridobljene s takim sistemom.

## 2 GLICKO RATING

Glicko rating poleg ratinga za posameznega igralca (uporabnik ali naloga) vpeljuje še deviacijo RD, ki nam omogoča predstavitev ratinga posameznega igralca kot 95% interval zaupanja:

$$(r - 1.96RD, r + 1.96RD) \quad (1)$$

pri čemer se RD manjša ob vsaki posodobitvi ratinga, kjer se igralec sooči z drugim igralcem (ali več njih)

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2}\right)^{-1}} \quad (2)$$

kjer RD predstavlja deviacijo pred posodobitvijo,  $d^2$  pa je definiran kot

$$d^2 = \left( q^2 \sum_{j=1}^m (g(RD_j))^2 E(1 - E) \right)^{-1} \quad (3)$$

$$q = \frac{\ln 10}{400} = 0.0057565 \quad (4)$$

$$E = E(s|r, r_j, RD_j) = \frac{1}{1 + 10^{-g(RD_j)(r-r_j)/400}} \quad (5)$$

$$g(RD) = \frac{1}{\sqrt{1 + 3q^2 (RD^2) / \pi^2}} \quad (6)$$

Nov rating  $r'$  se izračuna po formuli

$$r' = r + \frac{q}{1/RD^2 + 1/d^2} \sum_{j=1}^m g(RD_j) (s_j - E) \quad (7)$$

Zgornje formule so posplošene za posodobitev ratinga  $r$  in deviacije  $RD$  igralca proti skupini  $m$  nasprotnikov z deviacijami  $RD_1, RD_2, \dots, RD_m$  in ratingi  $r_1, r_2, \dots, r_m$ . Ivide z vrednostmi 0 ali 1 predstavljajo  $s_1, s_2, \dots, s_m$ .

Igralčev  $RD$  se ne posodablja samo, ko se ta sooča, temveč tudi ob preteku določene časovne periode, kar predstavlja zniževanje verodostojnosti trenutnega ratinga, če igralec določen čas ne igra. Tako se njegov  $RD$  po preteku  $t$  ocenjevalnih obdobj posodobi kot

$$RD = \min \left( \sqrt{RD_{old}^2 + c^2 t}, 350 \right) \quad (8)$$

konstanta  $c$  je, poleg maksimalne deviacije, ki je v tem primeru 350, edini parameter, ki ga nastavlja administrator sistema glede na izbrano ocenjevalno obdobje (poglavje 3.4).

Za matematične izpeljave formul glej referenco [2].

Iz formul vidimo, da je  $r'$  odvisen od  $RD$  igralca in ni uravnotežen tako, kot je pri Elo sistemu, kjer se zmagovalcu zviša rating toliko, kot se poražencu zniža. Ker naj bi velikost  $RD$  odražala koliko informacije imamo o igralcu, se v iteraciji ocenjevanja, kjer ima en igralec velik  $RD$ , drug pa majhen, prvemu zviša veliko več, kot se drugemu zniža, saj vemo, da je rating prvega nezanesljiv, rating drugega pa je. Enako se v tem primeru ne more upravičeno znižati rating drugega igralca toliko, kot bi se, če bi bil rating njegovega soigralca zanesljiv.

Kot primer vzemimo dva igralca z ratingom 1700, kjer prvi premaga drugega. Pri Elo verziji, kot jo uporabljajo v Ameriški šahovski zvezi, bi prvi dobil 16 rating točk, drugi pa izgubil prav tako 16 točk. Če vemo, da se je na primer prvi igralec ravno vrnil v igro po nekaj letih neaktivnosti, drugi igralec pa igra vsak vikend, bo ocena drugega bolj zanesljiva kot ocena prvega. Po intuiciji bi se tako moral rating prvega igralca povišati za precej več kot 16 točk, saj ne moramo zaupati njegovem trenutnem ratingu pri 1700, poleg tega pa je zmaga proti enakem, vendar dokaj zanesljivem ratingu drugega igralca, dovolj dober dokaz, da je njegova prava moč višja od 1700. Obratno pa bi se rating drugega igralca moral znižati za manj kot 16 točk, saj je njegov rating že dokaj zanesljiv pri 1700 in izguba proti uporabniku, katerega ratingu ne moremo zaupati ne pove veliko o njegovi pravi moči.

Igralčevo moč tako najboljše povzamemo z intervalom zaupanja namesto ene vrednosti. Tako si interpretiramo interval, ki ga izračunamo z Glicko (95% interval zaupanja), kot interval možnih vrednosti za igralčevo moč, kjer je rating igralca samo ocena njegove nepoznane resnične moči.

### 3 IZRAČUNI IZ EQUIZ PODATKOV

#### 3.1 Pridobljeni podatki

Iz podatkovne baze eQuiza so bili pridobljeni trenutni ratingi študentov in nalog ter kronološki zapis reševanj quizov, sestavljenih iz različnih nalog. Podatek o reševanju posamezne naloge v quizu je bil izračunan na podlagi sprememb ratingov nalog po končanem quizu. V tabelo podatkov je bil tako dodan stolpec correct z vrednostjo 0 ali 1 za vsako nalogo, glede na celotno pravilnost naloge. Ta podatek je bil uporabljen tudi v kasnejšnjih izračunih Elo in Glicko ratingov.

Za izračun Glicko ratinga je, za razliko od Elo ratinga, potreben še čas reševanja. Zato so bile posodobitve ratingov združene s tabelo reševanj, ki vsebuje potrebne čase. Pri tem so se pojavila neskladja, kjer nekatere posodobitve ratingov, ki se nanašajo na nek identifikator kviza, niso bile nikdar zabeležene, nekatera zabeležena reševanja pa se ne nanašajo na noben rating. Poleg tega so bili za veljaven kviz vzeti vsi, ki so trajali vsaj 5 minut, da se izognemo praznim oddajam kvizov. Tako so bili pridobljeni podatki za 305 različnih uporabnikov in 1348 nalog, za nadaljnje izračune pa so bili uporabni podatki za 235 uporabnikov in 1246 nalog.

#### 3.2 Metoda izračuna ratinga

Za izračune ratingov so bile od najstarejšega ratinga uporabnika po času reševanj zaporedoma pregledane naloge iz ustreznega kviza, ki so bile glede na prejšnji najstarejši ali (osnovni rating) uporabnika in naloge ponovno ratane po izbranem rating algoritmu (Elo ali Glicko). Tako je bil po zaporednih nalogah glede na reševanje (stolpec correct) izračunan nov rating uporabnika in novi ratingi nalog, ki so bili sproti zapisani v tabele na mesta z ustreznimi časi in identifikatorji izpitov, za uporabo v naslednjih korakih algoritma. Ob koncu zabeleženih ratingov so bili tako izračunani ratingi za vse naloge in uporabnike.

Primer zapisov izračunov ratingov za izpit z identifikatorjem 43:

id	userId	rating	classExamId	startedAt	finishedAt
2	7bf1181f-7f3-4aa9-9610-958584926d70	171	43	2019-06-17 14:39:53	2019-06-17 15:30:53

Figure 1.: Zapis za reševanje uporabnika

id	exerciseId	ratingELO	classExamId	correct	startedAt	finishedAt
13	1738	202	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
14	1765	202	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
15	1774	197	43	1	2019-06-17 14:39:53	2019-06-17 15:30:53
16	1785	202	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
17	1872	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
18	1922	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
19	1961	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
20	1990	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
21	2003	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
22	2014	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
23	2159	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
24	2201	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
25	2223	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
26	2229	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53
27	2233	201	43	0	2019-06-17 14:39:53	2019-06-17 15:30:53

Figure 2.: Zapisi za naloge na kvizu

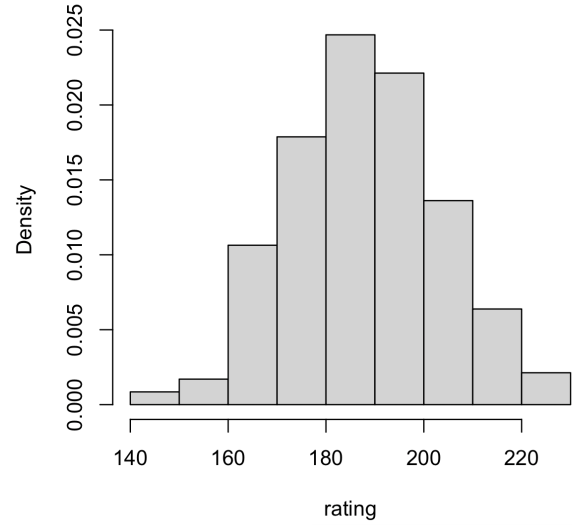


Figure 3.: Histogram končnih Elo ratingov uporabnikov

### 3.3 Trenutni rating sistem - Elo

Pri oddaji kviza so pridobljeni prejšnji rating posameznih nalog v kvizu in prejšnji rating uporabnika. Za naloge in uporabnika so potem zaporedoma po nalogah izračunani novi ratingi po naslednjih enačbah:

$$R'_A = R_A + K \cdot (S_A - E_A) \quad (9)$$

$$E_A = \frac{Q_A}{Q_A + Q_B} \quad (10)$$

Kjer so

$$Q_A = 10^{R_A/66} \quad Q_B = 10^{R_B/66} \quad (11)$$

$S_A$  predstavlja dejanski izid, v primeru eQuiza 0 ali 1,  $E_A$  pa pričakovan izid.

$K$  je po navadi izračunan glede na statistično analizo reševanja, na različne načine [3], v trenutni implementaciji ratinga pa je privzet kot konstanten z vrednostjo 4,5.

Po takem postopku je bil po zgoraj opisanih algoritmičnih izračunih Elo rating na filtriranih podatkih z osnovnim ratingom 200.

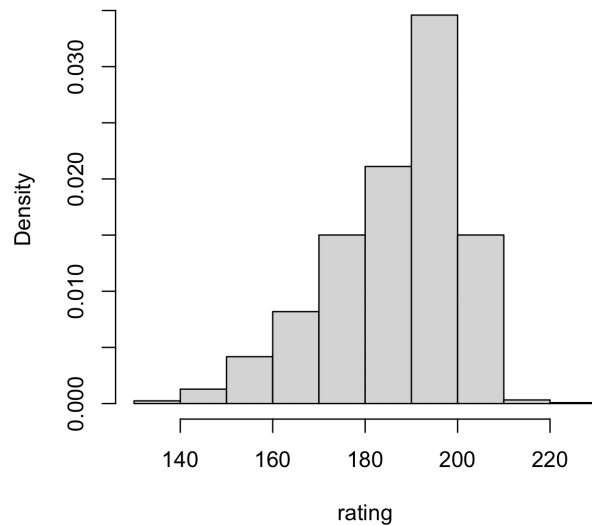


Figure 4.: Histogram končnih Elo ratingov nalog

Pri porazdelitvi ratingov uporabnikov na sliki 3, predvsem pa nalog na sliki 4 opazimo, da so tako uporabniki, kot tudi naloge v veliki meri ocenjeni pod osnovnim ratingom 200. Pri nalogah pa je porazdelitev celo bolj podobna eksponentni kot normalni.

Pri uporabi Elo ratinga, kot je implementiran sedaj, se v kontekstu eQuiza pojavijo očitne težave. Za iskanje vzroka bomo primerjali del podatkov, izračunanih v poglavju 3.4, s podatki trenutnega Elo ratinga uporabnika, kjer rating uporabnika že po prvem reševanju pade (slika 5a), pri Glicko pa pri istem reševanju naraste (slika 5b).

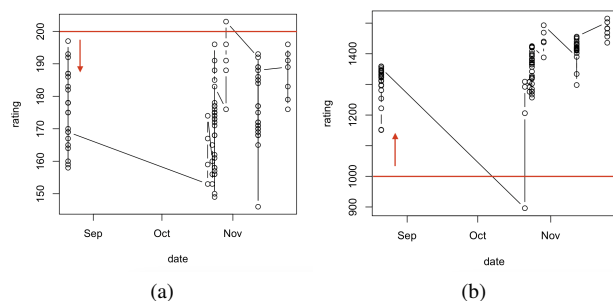


Figure 5.: Grafi posodobitev ratinga uporabnika za Elo (a) in Glicko (b)

Iz tabele reševanja prvega kviza (slika 6) vidimo, da je uporabnik pri pretežno pravilno rešenih nalogah (correct = 1) dosegel negativno posodobitev ratinga pri Elo. Opazimo tudi, da je pri Elo in Glicko razlika v ratingih nalog, kjer so pri Elo vse ocenjene pod base ratingom 200, kar bo pri ocenjevanju pomenilo, da bodo pravilno rešene naloge minimalno zvišale rating uporabnika, napačno rešene naloge pa ga močno znižale, saj je uporabnik napačno rešil "lahko" nalogo. Torej uporabnik izredno težko zviša svoj rating, kar je rezultat nizko ocenjenih nalog.

classExamId	exerciseld	ELO	Glicko	correct
7524	12132	194	824	1
7524	2518	151	807	1
7524	1960	115	1044	0
7524	1968	116	644	1
7524	2001	113	604	1
7524	7157	138	549	1
7524	2069	117	678	1
7524	2085	149	1100	1
7524	2096	148	800	1
7524	2130	121	1047	0

Figure 6.: Tabela nalog v kvizu

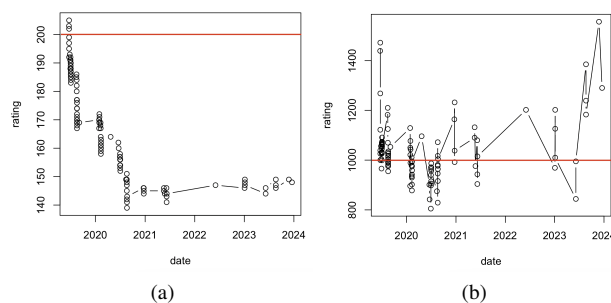


Figure 7.: Grafi posodobitev ratinga naloge za Elo (a) in Glicko (b)

Pri primerjavi ratingov za nalogo (slika 7), ki je pri Elo ocenjena pod osnovnim ratingom, pri Glicko pa nad, opazimo, da so pri trenutni implementaciji Elo, spremembe ratinga naloge majhne, rating pa vztrajno pada. V kolikor naloga nima možnosti zadostno zvišati svoj rating v posamezni iteraciji pa bo pri uporabi ratinga v reševanju kvizov, pričakovano, da bodo naloge ocenjene pod base ratingom, saj uporabniki kljub vsemu v povprečju rešijo več nalog prav kot napačno. V podatkih pridobljenih iz eQuiza so napačno rešene naloge proti pravilno rešenim v razmerju približno 1:2.

Odzivnost ratinga na posamično nalogo, oziroma omejitev spremembe ratinga določa pri Elo ratingu parameter K (K-faktor). Nizek K-faktor predstavlja nizko odzivnost ratinga, kar je v trenutni implementaciji očiten problem. S trenutnim K faktorjem in osnovnim ratingom se Elo izkaže za preveč togega, zato uporabniki ne uspejo pridobiti dovolj visokega ratinga, da ne bi bile vse naloge pretežno nizko ocenjene, reševanje dobrih študentov pa je torej navzgor omejeno zaradi nizkih ocen nalog. Ker ima med drugim posamezen kviz na študenta veliko večji vpliv kot na posamezno nalogo zaradi mnogih nalog v kvizu, ki tekmujejo proti študentu, bomo s pretežno nizko ocenjenimi nalogami poskrbeli, da bo študent večno nizko ocenjen. Posledično pa tudi naloge ne bodo imele, brez dovolj velikega K-faktorja možnosti, da se jim rating zviša.

Kot rešitev se torej ponudi ta, da K-faktor povečamo in s tem povečamo možnost nalogam in posledično uporabnikom, da si zvišajo rating, vendar bomo lahko s tem naredili rating nestabilen. Lahko bi uporabili spremenljiv K-faktor, za kar obstaja mnogo pristopov, najpogostejše pa se K-faktor spreminja za konstantno vrednost glede na rating razdeljen na intervale [3].

Po zvišanju trenutnega K-faktorja z 4,5 na 32 in premiku osnovnega ratinga na 1000, kar je pogost nabor parametrov za Elo [3], izgleda porazdelitev ratinga boljje (slika 8 in 9).

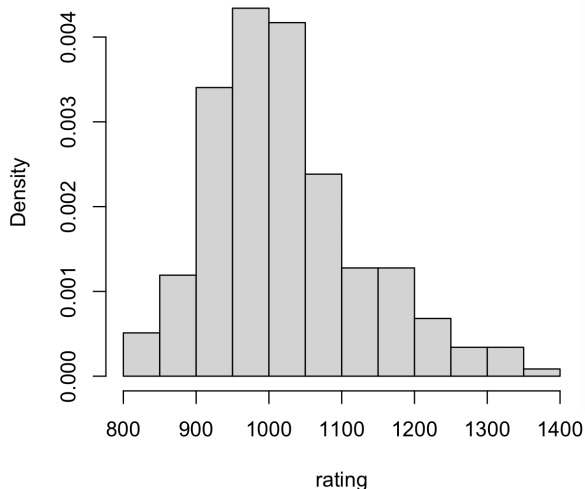


Figure 8.: Histogram končnih Elo ratingov nalog

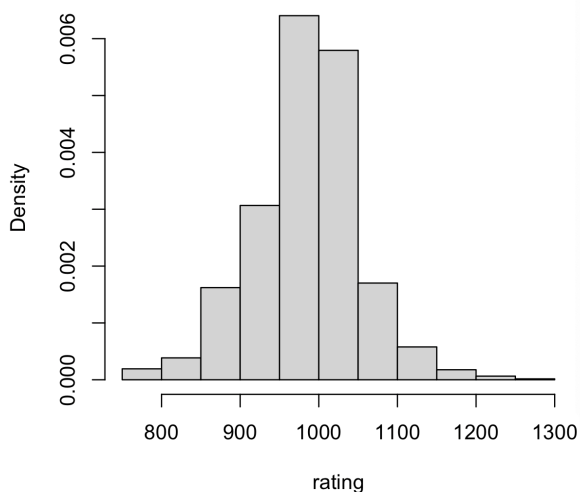


Figure 9.: Histogram končnih Elo ratingov nalog

Poglejmo še spremembe ratinga za konkretnega uporabnika.

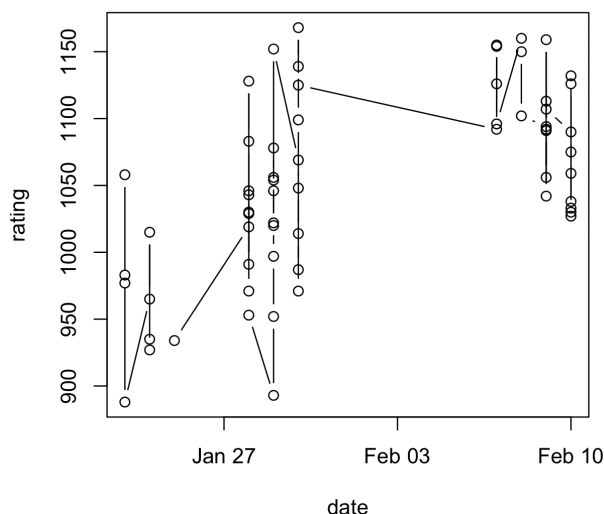


Figure 10.: Graf posodbitev ratinga uporabnika pri Elo

Na sliki 10 opazimo problem zvišanega K-faktorja, kar naredi Elo rating uporabnika zelo variabilen. Problem pa se da rešiti z Glicko ratingom. Glicko rating varianco v ratingu prilagaja glede na estimirano zanesljivost ratinga, kar lahko predstavlja rešitev, saj Glicko upošteva, da je rating nalog lahko razmeroma nezanesljiv, saj so ocenjevale bolj na redko kot uporabnik in tako omogoča nalogam, da se bolj razpršijo zaradi večjih prilagotiev ratinga glede na igralca, dokler njihov rating ne postane bolj zanesljiv zaradi kasnejših ocenjevanj. Če je naloga ocenjevana bolj pogosto, bo z manjšanjem deviacije prišla do natančnejšega ratinga.

To predstavlja tudi odgovor na vprašanje o zanesljivosti ratinga, če vemo, da so uporabniki kviz reševali različno mnogokrat v različno dolgih intervalih in tako njihov rating lahko različno zanesljivo odraža pravo sposobnost reševanja uporabnikov ali težavnost nalog. V nadaljevanju je predstavljen izračun Glicko ratinga za enake podatke.

### 3.4 Glicko rating na pridobljenih podatkih

Po enakem postopku kot Elo rating je bil zaporedno izračunan Glicko rating. Kot opisano v dokumentaciji Glicko ratinga [1], pa so bile naloge, ki spadajo v isto iteracijo ocenjevanja obravnavane vsaka posebej, kot več iger proti enako ocenjenemu igralcu (uporabniku) in ne zaporedno z vmesnimi posodobitvami ratingov, kot v trenutnem rating sistemu. Osnovni rating je bil nastavljen na 1000;

Za upoštevanje časov reševanj nalog, je potreben še izračun konstante  $c$  kot parameter Glicko ratinga. Kot opisano v dokumentaciji [1] lahko  $c$  izračunamo glede na to, po kolikšnem času želimo, da RD igralca pade na maksimalno vrednost, v tem primeru 350. Torej če v kontekstu ocenjevanj študentov privzamemo, da bo po preteklem času enega semestra, četudi ima študent na začetku minimalen RD, postal njegov rating nezanesljiv in pade njegov RD na 350. Za tak čas je bilo vzetih 120 dni, da pade RD iz povprečne vrednosti 50 na 350.

$$350 = \sqrt{50^2 + 120 \cdot c^2} \quad (12)$$

$$c \approx 32 \quad (13)$$

Za obdobje ocenjevanja (rating period) je bil vzeti 1 dan, kar pomeni, da se ocenjevanjem znotraj istega dne ne bo deviacija zviševala zaradi neaktivnosti. Tako obdobje lahko nastavimo poljubno, dokumentacija Glicko ratinga [1] pa priporoča, da povprečno na rating obdobje pride okrog 5-10 ocenjevanj, zato da je rating v posameznem obdobju dovolj natančen, hkrati pa so reševanja razdeljena v dovolj obdobjih, da lahko upoštevamo obdobja, ko bi uporabnik lahko reševal, pa ni. V našem primeru bo posamično ocenjevalno obdobje, kjer je uporabnik reševal, vedno dovolj natančno, saj se uporabnik ne more soočiti z manj kot minimalnim številom nalog na kviz in se bo tako uporabnik vedno v kvizu pomeril proti vsaj 5 nalogam. Reševanje torej ni enakomerno razporejeno ampak poteka v skupinah (posamezni kvizi).

Na slikah 11, 12, 13 in 14 so histogrami izračunov končnih ratingov in deviacij uporabnikov in nalog po Glicko rating sistemu.

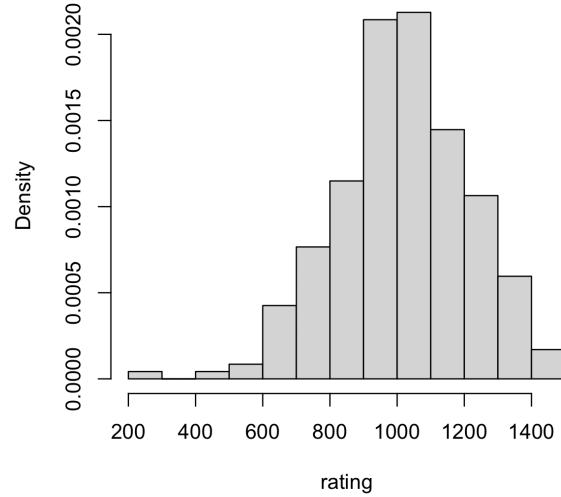


Figure 11.: Histogram končnih Glicko ratingov uporabnikov

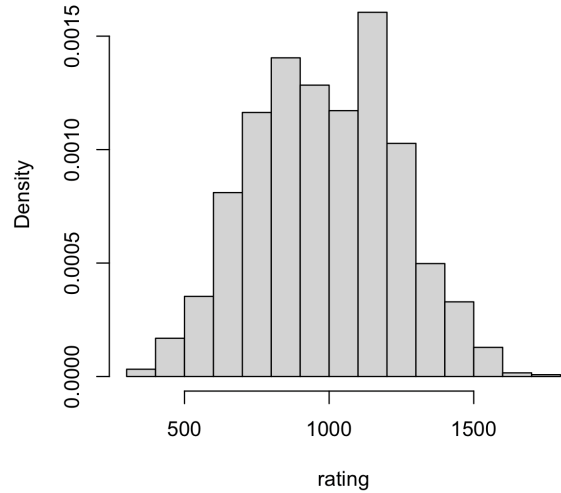


Figure 12.: Histogram končnih Glicko ratingov nalog

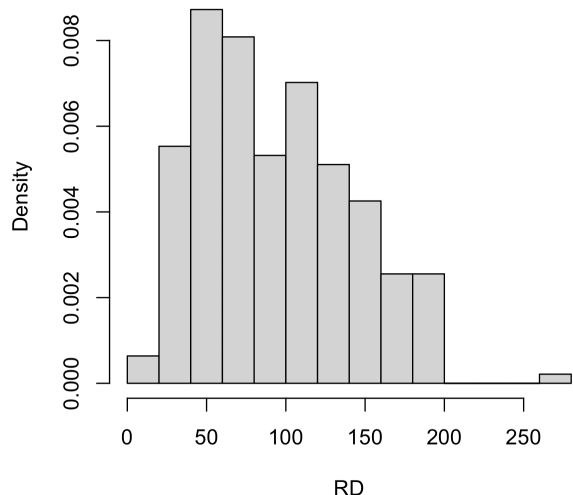


Figure 13.: Histogram končnih rating deviacij uporabnikov

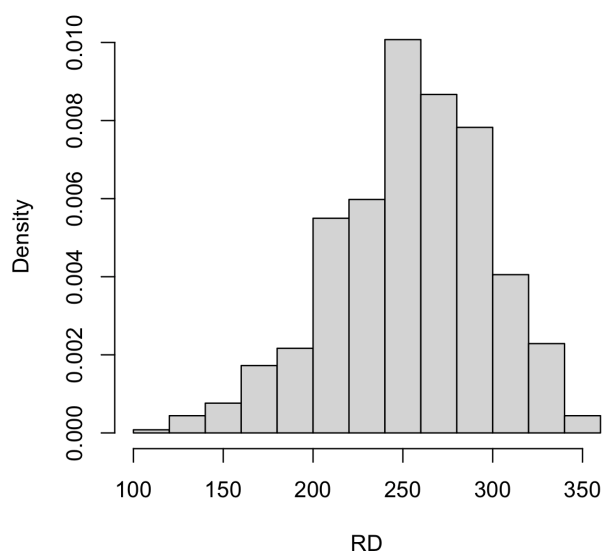


Figure 14.: Histogram končnih rating deviacij nalog

Opazimo, da je pri ratingu nalog RD povprečno večji (slika 14) kot pri ratingu uporabnikov (slika 13), saj so naloge povprečno ocenjene bolj na redko (ocenjevanja nalog so razpršena bolj enakomerno kot reševanja uporabnikov).

Ker smo za enoto vzeli dneve moramo za razlike v datumih ocenjevanj upoštevati razliko v dnevih, kjer naj bi se njegov RD stalno spreminjal. Dovolj pa je, da RD posodobimo samo pred novim ocenjevanjem ali pa ob vpogledu v rating uporabnika, tako da upoštevamo čas od njegovega zadnjega zabeleženega ocenjevanja. Za prikaz statistike glicko ratinga v tem primeru končni

izračun deviacije ni bil izveden, saj so bila ocenjevanja izvedena v zelo različnih obdobjih od 2019 do 2023 in bi tako velik del študentov imel maksimalen RD. Za njihove ratinge to nima vpliva, saj se je med njihovimi ocenjevanji RD prilagajal tudi po času. Tako lahko za posameznega študenta izračunamo tudi današnji RD po formuli za RD po preteklem času (8).

Zdaj pogledimo kako se rating istega uporabnika kot v primeru pri Elo (sliki 10), posodablja pri Glicko. Kjer so ratingi razporejeni čez razmeroma kratko obdobje, pričakujemo, da bo rating temu primerno zanesljiv in bo tako rating postopoma konvergiraj, oziroma se bo interval zaupanja za rating zmanjšal (slika 15 in 16).

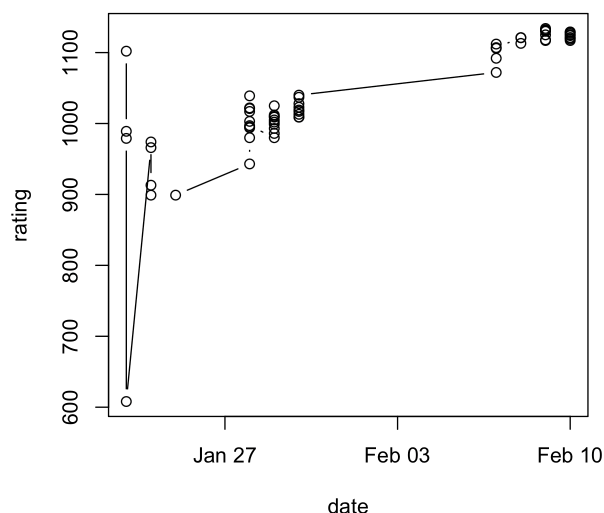


Figure 15.: Graf posodobitev ratinga za uporabnika pri Glicko

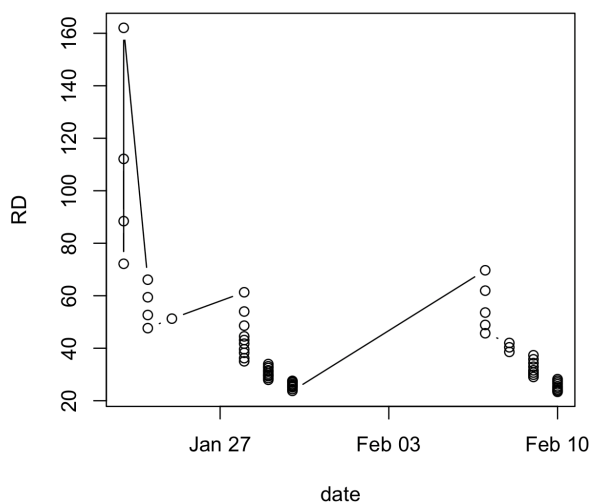


Figure 16.: Graf posodobitev RD za uporabnika

## 4 GLICKO ZA SPREMLJANJE ŠTUDENTOV

Z vpeljavo RD pridobimo v primeru študentov informacijo o sprotne delu, saj bo RD manjši za študente, ki so pogostejše ocenjeni. Če bi k ratingu pripomogli tudi, na primer redni kolokviji, bi reševanje dodatnih naključnih reševanj kvizov/nalog bilo razvidno iz manjšega RD, kar kaže na več sprotne delu in obratno. Predmet verjetnost in statistika, iz katerega imamo naloge na voljo tudi na eQuizu, se da na primer opraviti v celoti s sprotnimi preverjanji, kjer je snov enakomerno razdeljena na 5 kolokvijev skozi celoten semester. Torej bo študent, ki opravi samo eno preverjanje od petih ustrezno imel veliko deviacijo, saj imamo informacijo o njegovem znanju čedalje bolj nezanesljivo, ker vemo čedalje manj, koliko študent dejansko ve o statistiki, odkar je reševal tisti kolokvij.

Za nasprotnika uporabnika – nalogo, pa deviacija predstavlja kdaj je bila naloga nazadnje ocenjena, torej koliko je njena ocena zanesljiva. V primeru uporabnika, ki izbira naloge iz zbirke na eQuizu, bi lahko naloge z veliko deviacijo identificirali kot nepriljubljene (malo uporabnikov se je lotilo naloge).

### 4.1 Igrifikacija

V kontekstu igrifikacije eQuiza, lahko ratinge, predvsem pa RD preslikamo v razrede iz katerih lahko bolj prijazno uporabniku spremljamo njegovo aktivnost na eQuizu. Iz statistike končnih RD uporabnikov lahko uporabimo kvartile in razdelimo ratinge na 4 dele:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
7.127	48.392	88.107	91.466	127.271	275.376

Tako lahko razporedimo uporabnike glede na njihove RD:

$RD \leq 48,392$  : visoka aktivnost

$48,392 < RD \leq 88,107$  : srednje visoka aktivnost

$88,107 < RD \leq 127.271$  : srednje nizka aktivnost

$127.271 < RD$  : nizka aktivnost

## 5 ZAKJUČEK

Glicko rating nam doda novo informacijo glede vloženega dela uporabnika. Dobimo informacijo o količini reševanj v določenem časovnem intervalu, posledično pa tudi zanesljivost ratinga izračunanega z Glicko, kjer je zanesljivost upoštevana tudi med samim računanjem. Poleg sprotne delu pa dobimo informacijo tudi o reševanju nalog, njihovi težavnosti in priljubljenosti. Ker pri tem poskušamo zanesljivost zvečati, lahko z vpeljavo parametriziranih nalog, ki jih tako večkrat uporabimo na preverjanjih, zvišamo zanesljivost njihovih ratingov. Če v rating vključimo še podatke iz izpitov in skupnih preverjanj, pa bo posledično rating še zanesljivejši, saj bo v isti iteraciji naloge reševalo veliko učencev.

Kot je do sedaj algoritem operiral na trenutni podatkovni bazi eQuiza, lahko podoben algoritem uporabimo v nadaljevanju, za beleženje ratingov na enak način, kot je implementiran trenutno. Implementacija Glicko ratinga za eQuiz je na voljo skupaj z vsemi programi uporabljenimi pri navedenih izračunih na repozitoriju Glicko eQuiz.

Za Glicko rating pa obstaja tudi bolj kompleksna izboljšana različica Glicko-2, ki upošteva tudi konsistentnost reševanja in bi s tem lahko odpravila določene neprilakovane tehnične težave pri spletnem reševanju.

## REFERENCES

- [1] *The glicko system*. <http://www.glicko.net/glicko/glicko.pdf>.
- [2] *Parameter estimation in large dynamic paired comparison experiments*. <http://www.glicko.net/research/glicko.pdf>.
- [3] Wikimedia projects, Contributors to: *Elo rating system*, Jan 2024. [https://en.wikipedia.org/wiki/Elo\\_rating\\_system](https://en.wikipedia.org/wiki/Elo_rating_system).
- [4] Wikimedia projects, Contributors to: *Glicko rating system*, Jan 2024. [https://en.wikipedia.org/wiki/Glicko\\_rating\\_system](https://en.wikipedia.org/wiki/Glicko_rating_system).