

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. Možna izboljšava takega sistema predstavlja Glicko rating.

Key words: equiz, Glicko rating

1 UVOD

Ena izmed funkcionalnosti aplikacije eQuiz je rating uporabnikov in nalog preko Elo rating sistema. Uporabnik se sooči z nalogo in na ta način se istočasno izračuna rating študenta in rating naloge, kot bi igrala uporabnik in naloga šah. Uporabnik lahko na nalogo odgovori prav ali napačno, kar je ekvivalentno temu, da uporabnik nalogo premaga ali proti njej izgubi. Pojavijo pa se pomankljivosti takega rating sistema.

Če uporabljamo eQuiz za spremljanje ratinga uporabnikov in nalog v sklopu določenega predmeta, bo proti številu uporabnikov, ki bodo v tem letu aktivni, število nalog navadno mnogo večje. To pa pomeni, da bodo, dokler uporabniki ne rešujejo samih enakih nalog, ti prejeli posodobitev ratinga pogostejše kot posamezna naloga. Bolj aktivni uporabniki pa bodo nasploh dobili več posodobitev. Torej lahko ratingi, pripisani določeni nalogi ali uporabniku izvirajo iz več ali manj reševanj v določenem časovnem intervalu, kar naredi rating bolj ali manj verodostojen. Zaupanje v določen rating lahko tako iz posamezne številke raje razširimo na interval zaupanja.

2 GLICKO RATING

Za tak pristop je primeren Glicko rating, ki 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

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

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

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

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

Nov rating r' se izračuna po formuli

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

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 . s_1, s_2, \dots, s_m predstavljajo izide, ki so lahko 0 ali 1, za izgubo ali zmago.

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 posodobi kot

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

konstanta c je, poleg maksimalne deviacije, ki je v tem primeru 350, edini parameter, ki ga nastavlja administrator sistema. Več o tem v kontekstu equiza.

Matematične izpeljave formul so na voljo tukaj.

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.

Igralčevo moč tako najbolje 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 EQUITZ 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, kar pomeni, da so k izračunu ratinga nekoč lahko prispevali različni tipi ocenjevanj, trenutno pa se v rating štejejo samo Rating Quizi. Tako so bili pridobljeni podatki za 305 različnih uporabnikov in 1348 nalog, za nadaljnje izračune pa so bili uporabni podatki za 261 uporabnikov in 1299 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-f7f3-4aa9-8610-958584926d70 | 171 | 43 | 2019-06-17 14:39:53 | 2019-06-17 15:30:53 |

```
1 SELECT u.id, u.userId, u.rating, u.classExamId, a.startedAt, a.finishedAt
2 FROM computed_user_ratings u JOIN exam_attempts a ON u.classExamId = a.classExamId
3 WHERE u.classExamId=43;
```

| 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 |

```
1 SELECT e.id, e.exerciseId, e.ratingELO, e.classExamId, e.correct, a.startedAt, a.finishedAt
2 FROM computed_exercise_ratings e JOIN exam_attempts a ON e.classExamId = a.classExamId
3 WHERE e.classExamId=43;
```

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 število iger na katerih temelji trenutni rating (in število iger v turnamentu pri šahu), v trenutni implementaciji ratinga pa je privzet kot konstanten z vrednostjo 4,5.

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

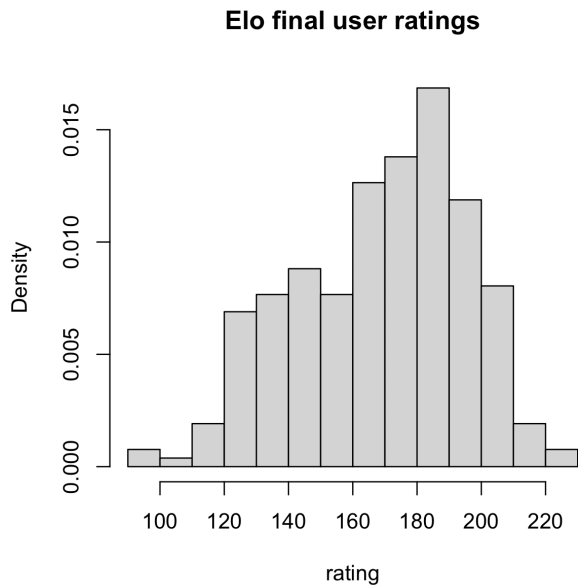


Figure 1.: Histogram končnih Elo ratingov uporabnikov

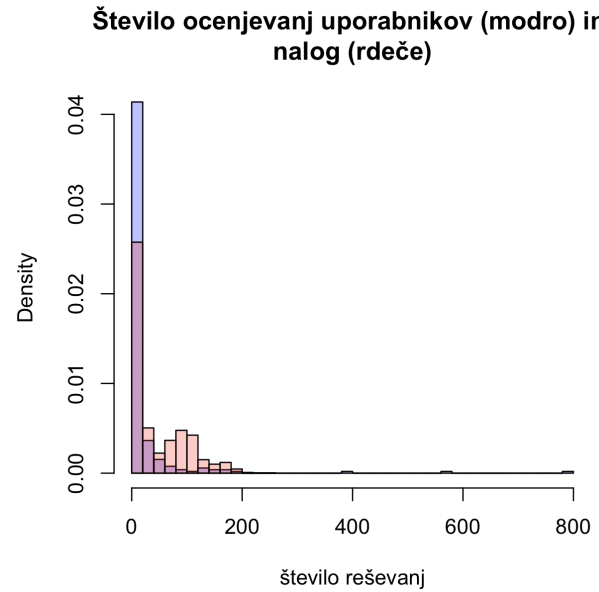


Figure 3.: Histogram reševanja za uporabnike

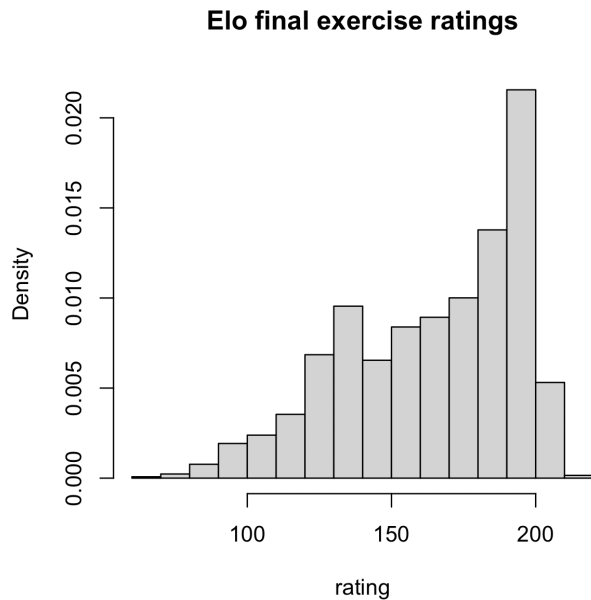


Figure 2.: Histogram končnih Elo ratingov nalog

Zaradi velikega števila nalog iz katerih se naključno generira kviz, so posamezne naloge povprečno manjkrat ocenjene kot uporabniki, poleg tega pa so po številu rešenih kvizov med uporabniki velike razlike.

Zaradi tega nekateri ratingi bazirajo na manj ali več ocenjevanjih v določenem časovnem intervalu in tako boljše ali slabše odražajo pravo sposobnost reševanja uporabnikov ali težavnost nalog. Zato je bil v nadaljevanju za enake podatke izračunan še Glicko rating.

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, 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. Pri trenutnem eQuiz osnovnem ratingu 200 ima večina ratingov po Glicko vrednost 0, zato je bil osnovni rating nastavljen na 1000;

Za upoštevanje časov reševanj nalog, je potreben še izračun konstante c kot parameter Glicko ratinga. Kot opisano v dokumentaciji 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.

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

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

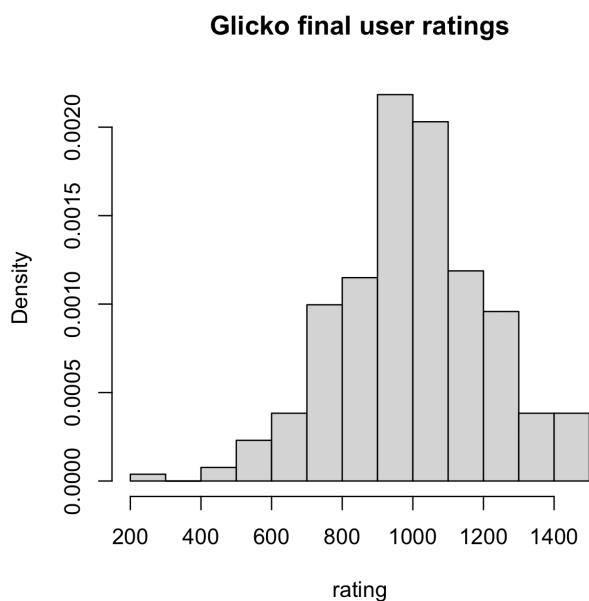


Figure 4.: Histogram končnih Glicko ratingov uporabnikov

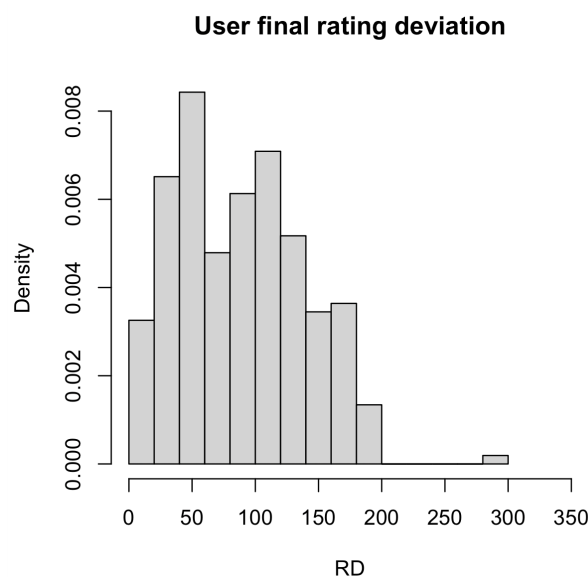


Figure 6.: Histogram končnih rating deviacij uporabnikov

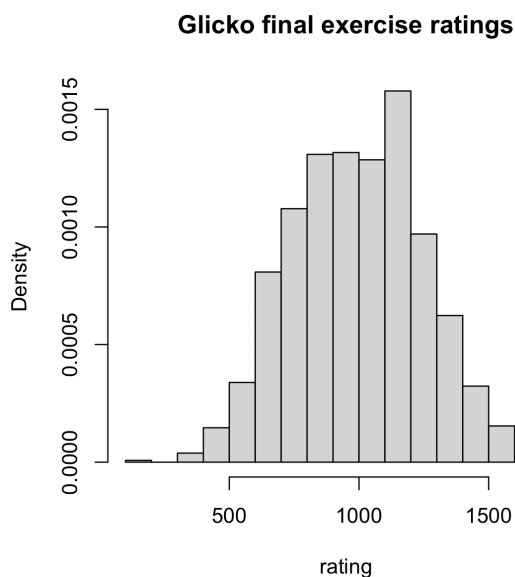


Figure 5.: Histogram končnih Glicko ratingov nalog

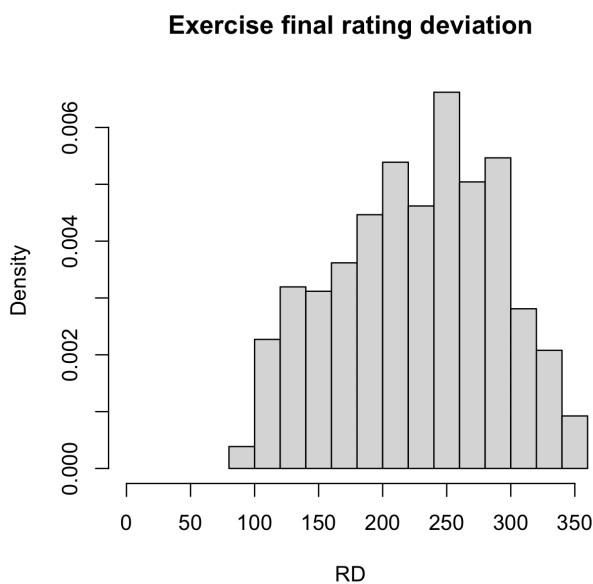


Figure 7.: Histogram končnih rating deviacij nalog

Opazimo, da je pri ratingu nalog RD povprečno večji kot pri ratingu uporabnikov, saj so bile naloge povprečno manjkrat ocenjene ali pa bolj na redko (med ocenjevanji je preteklo več časa)

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.

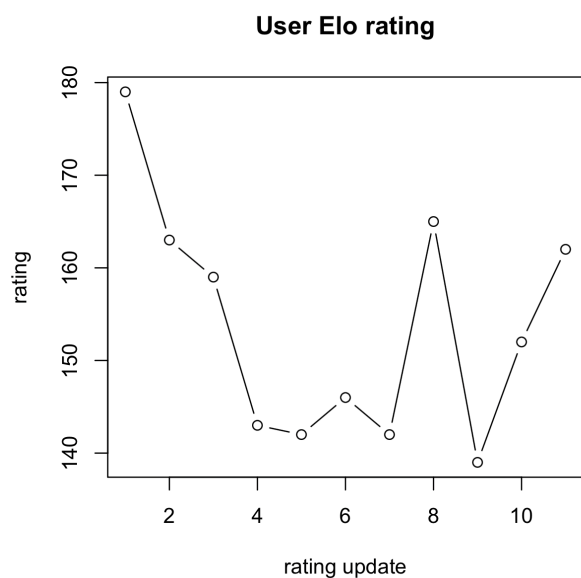


Figure 8.: Primer posodobitev ratinga za uporabnika pri Elo sistemu

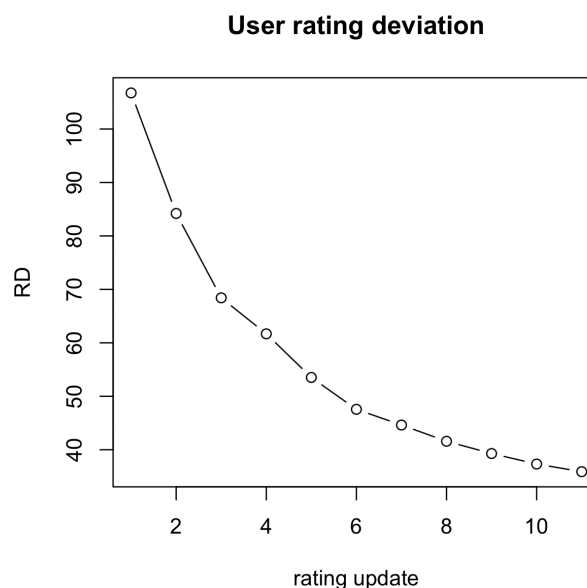


Figure 10.: Posodobitev RD za uporabnika

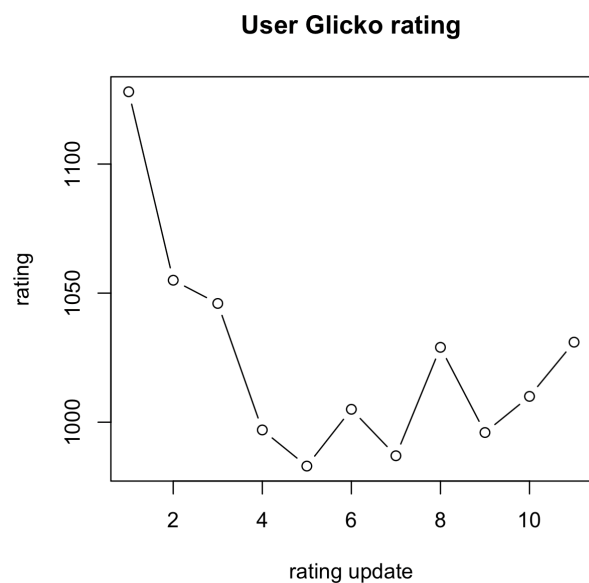


Figure 9.: Posodobitev ratinga za istega uporabnika pri Glicko sistemu

Za primer je bil vzet uporabnik, ki je vse ratinge pridobil v roku štirih dni., zato so spremembe v ratingu čedalje manjše, saj se njegov RD hitro manjša.

Če vzamemo uporabnika z ratingi razporejenimi čez večji časovni interval, spremembe ratinga in RD nista več nujno majhna. Skoke opazimo pri prvih ocenjevanjih po velikem časovnem intervalu brez posodobitev.

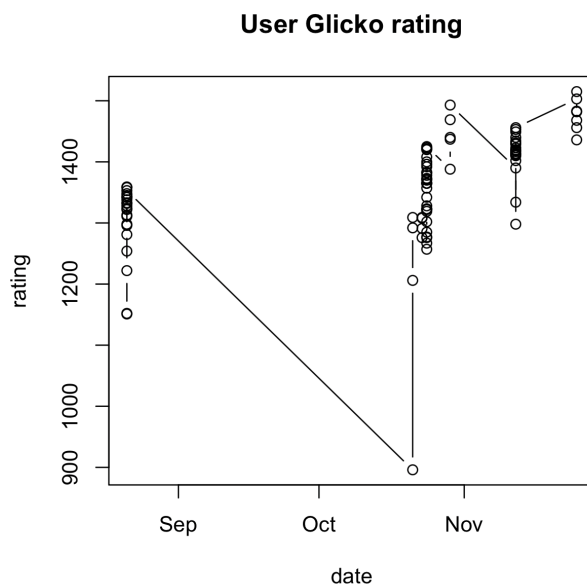


Figure 11.: Posodobitve ratinga skozi večji časovni interval

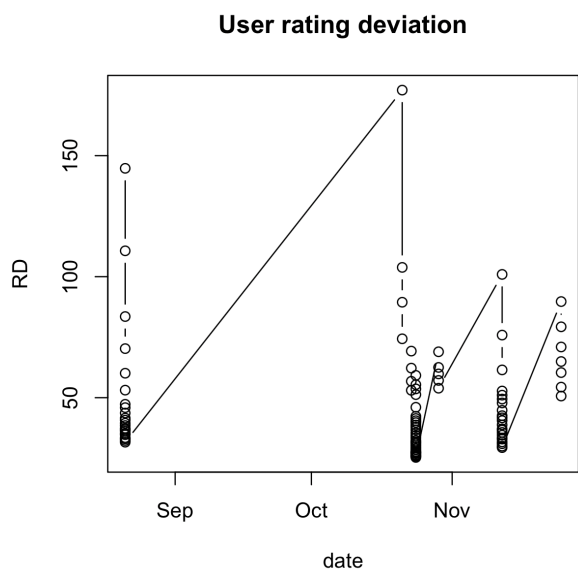


Figure 12.: Posodobitve deviacije skozi večji časovni interval

4 GLICKO ZA SPREMLJANJE ŠTUDENTOV

Z vpeljavo RD pridobimo v primeru študentov informacijo o sprotne delu, saj bo RD manjši za študente, ki so pogosteje 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. Za primer, pri predmetu Verjetnost in statistika je možno predmet 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 nalogo 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 do sedaj. Implementacija Glicko ratinga za eQuiz je na voljo skupaj z vsemi programi uporabljenimi pri navedenih izračunih na GitHub-u

Za Glicko rating pa obstaja tudi bolj kompleksna izboljšana različica Glicko-2, ki bi lahko bila bolj primerna za dejansko implementacijo v aplikaciji.