

Sistem za preporuku poteza u pokeru

Članovi tima:

1. Stefan Đurica SV 35/2021
2. Nenad Berić SV 23/2021

Opis problema

Motivacija

Poker je igra koja kombinuje strategiju, psihologiju i statistiku. Donošenje optimalnih odluka u realnom vremenu zahteva iskustvo i analizu velikog broja faktora. Cilj ovog projekta je izgradnja sistema koji može da preporuči najbolji potez (call, raise, fold) na osnovu trenutne situacije u igri. Takav sistem može biti koristan:

- početnicima za učenje strategije (kao sto sam ja jer uvek all in-ujem u prvom potezu),
- analizi statističkih obrazaca,
- istraživanju donošenja odluka u pokeru.

Pregled problema

Trenutno postoje softveri i botovi za poker, ali većina je fokusirana na simulacije ili treniranje profesionalaca. Nedostaje jednostavan sistem zasnovan na pravilima koji:

- ima dovoljno brz i intuitivan user interface
- omogućava korisniku da vidi faktor rizika i potencijalni dobitak
- uzima u obzir prethodno ponašanje drugih igrača

Naš sistem će koristiti **sisteme bazirane na znanju** (rule-based) kako bi:

- preporučio optimalan potez u preflop fazi igre

- uključio pravila o verovatnoći pobeđe, pozicije igrača i rangiranju ruke
 - prilagodio igru u zavisnosti od prethodnog ponašanja ostalih igrača
-

Metodologija rada

Ulazi u sistem (Input)

- Broj igrača za stolom i pozicija svakog igrača
- Karte u ruci korisnika
- Trenutni ulog, pot i akcija prethodnih igrača

Izlazi iz sistema (Output)

- Preporučeni potez: fold, call/check, raise

Baza znanja

Sistem će koristiti pravila bazirana na:

- rangiranju ruku i šansi za pobeđu u preflop fazami igre
- tabelama verovatnoće (odnosno “hand strength tables”)
- poziciji igrača za stolom
- količini čipova u odnosu na protivnike, agresivnosti stola i veličini pota

Baza znanja se popunjava iz literature, poker strategija i statističkih simulacija.

Template za generisanje RAISE/CALL forward chaining pravila

Svrha

Ovaj template definiše generički okvir za kreiranje pravila u **forward chaining** logici koja

odlučuje da li igrač treba da **RAISE**, **CALL** ili eventualno izvrši drugu akciju u poker rundi. Pravila se generišu dinamički pomoću vrednosti iz CSV ili JSON podataka.

Ulazi (template header)

Svako pravilo definiše sledeće parametre:

- **ruleName** – jedinstveno ime pravila
 - **stakeLevel** – nivo uloga (**high_stakes** ili **normal_stakes**), koristi se kao **activation-group**
 - **salience** – prioritet pravila
 - **raiseThreshold** – operator za poređenje trenutnog raise-a sa pragom (>, <=, itd.)
 - **stakesOperator** – operator za poređenje raise-a sa minimalnim blind-ovima
 - **positionThreshold** – prag pozicije igrača za pravilo
 - **positionOperator** – operator za poređenje pozicije (>, <=, itd.)
 - **handCategory** – minimalna kategorija ruke potrebna da pravilo aktivira akciju
 - **requiresStrongEconomy** – da li igrač mora imati jaku ekonomiju
 - **requiresAggressiveTable** – da li pravilo zahteva agresivan sto
 - **action** – akcija koju sistem predlaže (**RAISE**, **CALL**, itd.)
 - **log** – tekst koji se štampa prilikom aktivacije pravila
-

Opis logike template-a

- Pravilo uvek počinje sa **\$r: Round()**, što predstavlja trenutnu rundu igre.
- **Raise i stake uslovi:**
 - **eval(\$r.getCurrentRaise() @{{raiseThreshold}} (min_bigger_raise * \$r.getPot()) && \$r.getCurrentRaise() @{{stakesOperator}} min_bigger_raise_blinds)**
 - Omogućava fleksibilnu proveru da li je trenutni raise dovoljno visok da se pravilo aktivira.
- **Provera jačine ruke:**
 - Ako je **handCategory** definisana, uključuje se evaluacija:
 - **eval(handEvalService.isHandAtLeast(\$r.getHand(), "handCategory"))**
 - Ovim se osigurava da pravilo važi samo za ruke određene kategorije (npr. Premium, Strong, Medium).
- **Pozicija igrača:**
 - Ako je definisan **positionThreshold**, uključuje se uslov:
 - **eval(\$r.getPlayerPosition() positionOperator positionThreshold)**
 - Ovo omogućava da se pravilo aktivira samo za dobre ili loše pozicije.

- **Ekonomija i agresivnost stola:**
 - Ako **requiresStrongEconomy** == true → proverava se **\$r.getStrongEconomy()**
 - Ako **requiresAggressiveTable** == true → proverava se **handService.isTableAggressive()**
 - **Akcija i log:**
 - Kada su svi uslovi ispunjeni, **\$r.setSuggestedAction("action")** definiše predloženu akciju.
 - **System.out.println("Action: log")** omogućava praćenje aktiviranih pravila.
-

Forward Chaining template za FOLD

Svrha

Ovaj template definiše generički okvir za kreiranje pravila u **forward chaining** logici koja odlučuje kada igrač treba da **FOLD** ili da ne izvrši fold u poker rundi. Pravila se generišu dinamički pomoću vrednosti iz CSV ili JSON podataka.

Ulazi (template header)

Svako pravilo definiše sledeće parametre.

- **ruleName** – jedinstveno ime pravila
 - **stakeLevel** – nivo uloga ili kontekst (npr. **fold_decision**) koji se koristi kao **activation-group**
 - **salience** – prioritet pravila
 - **requiresPlayableHand** – da li pravilo zahteva da ruka bude **ispod praga Playable**
 - **requiresMediumHand** – da li pravilo zahteva da ruka **nije Medium**
 - **requiresBadPosition** – da li pravilo zahteva lošu poziciju igrača
 - **requiresBigRaise** – da li pravilo zahteva veliki raise od prethodnih igrača
 - **requiresWeakEconomy** – da li pravilo zahteva slabu ekonomiju igrača
 - **requiresPassiveTable** – da li pravilo zahteva pasivan sto
 - **action** – akcija koju sistem predlaže (**FOLD** ili **NO FOLD**)
 - **log** – tekst koji se štampa prilikom aktivacije pravila
-

Opis logike template-a

- Pravilo uvek počinje sa **\$r: Round()**, što predstavlja trenutnu rundu igre.
- **Provera ruke:**
 - Ako **requiresPlayableHand == false**, uključuje se evaluacija:
 - **eval(!handEvalService.isHandAtLeast(\$r.getHand(), "Playable"))**
 - Ako **requiresMediumHand == false**, uključuje se evaluacija:

- `eval(!handEvalService.isHandAtLeast($r.getHand(), "Medium"))`
 - **Pozicija i ulog:**
 - Ako `requiresBadPosition == true`, uključuje se:
 - `eval($r.getPlayerPosition() <= bad_position_treshold)`
 - Ako `requiresBigRaise == true`, uključuje se:
 - `eval($r.getCurrentRaise() > (min_bigger_raise * $r.getPot()) && $r.getCurrentRaise() >= min_bigger_raise_blinds)`
 - **Ekonomija i pasivnost stola:**
 - Ako `requiresWeakEconomy == true` → proverava se:
 - `eval($r.getStrongEconomy() == false)`
 - Ako `requiresPassiveTable == true` → proverava se:
 - `eval(!handService.isTableAggressive())`
 - **Akcija i log:**
 - Kada su svi uslovi ispunjeni, `$r.setSuggestedAction("action")` definiše predloženu akciju.
 - `System.out.println("Backward: log")` omogućava praćenje aktiviranih pravila.
-

CEP za detekciju agresivnosti stola

Ulazi

Sistem koristi događaje koji dolaze u toku igre:

- **RaiseEvent** – predstavlja akciju raisa od strane igrača. Sadrži:
 - **playerId** – identifikator igrača
 - **amount** – iznos raisa
 - **eventTime** – vreme kada je raise izveden
- **TableAggressivenessEvent** – signal koji označava da je sto agresivan. Sadrži:
 - **eventTime** – vreme nastanka događaja
 - Automatski ističe nakon 10 minuta (**@expires(10m)**)

Sistem prati događaje u realnom vremenu, koristeći **prozor od 10 minuta** za agregaciju akcija.

Opis pravila

- **Detect table aggression**
 - Sakuplja sve **RaiseEvent** događaje u poslednjih 10 minuta (**over window:time(10m)**).

- Evaluira veličinu kolekcije:
 - Ako je broj raise-ova ≥ 5 , pravilo se aktivira.
 - Kada se pravilo aktivira:
 - Štampa poruku u log: "Detected table aggression! Raises: N".
 - Ubacuje novi **TableAggressivenessEvent** u sesiju, označavajući sto kao agresivan.
-

Backward Chaining za evaluaciju jačine ruke

Ulazi

- **HandCategory** – veza između konkretne ruke i njene osnovne kategorije (npr. "AA" → "Premium").
- **CategoryStrongerThan** – hijerarhija kategorija koja definiše koja kategorija je jača od koje.
- **Round** – trenutna runda igre (nije direktno potrebna za ovaj query, ali može biti kontekst).

Sistem ne ocenjuje igru po stepenu agresivnosti ili poziciji, već proverava hijerarhijsku jačinu ruke u odnosu na zadatu minimalnu kategoriju.

Opis pravila i query-ja

- **isCategoryStrongerOrEqual(\$catA, \$catB)**
 - Rekurzivni query koji proverava da li je kategorija **\$catA** jača ili jednaka **\$catB**.
 - Osnovni slučajevi:
 1. **\$catA** je ista kao **\$catB**.
 2. **\$catA** je direktno jača od **\$catB**.
 - Rekurzivni slučaj: postoji neka intermedijarna kategorija **\$intermediate** tako da **\$catA > \$intermediate** i **\$intermediate >= \$catB**.
 - **isHandAtLeastCategory(\$hand, \$requiredCategory)**
 - Glavni query za evaluaciju ruke.
 - Koristi sledeće korake:
 1. Pronađe kategoriju ruke **\$hand** pomoću **HandCategory**.
 2. Poziva **isCategoryStrongerOrEqual** da proveri da li je kategorija ruke jednaka ili jača od zadate **\$requiredCategory**.
-

Output

Za svaki query sistem vraća:

- TRUE – ruka zadovoljava minimalnu kategoriju ili je jača.
 - FALSE – ruka ne zadovoljava minimalnu kategoriju.
-

Forward chaining – dobra pozicija (nastavak od Fold logike)

Ulazi u sistem:

- Ruka je već $\geq 50\%$ (nije fold-ovana u prethodnoj fazi)
 - Pozicija: da li je u *drugih 50% igrača iza blindova* (dobra pozicija)
 - Pot Raised: da li je raise $> 50\%$ pota ili ≥ 3 big blinda (preflop)
 - H%: jačina ruke
 - More Chips: da li igrač ima jaku ekonomiju
-

Primer rezonovanja – Forward Chaining za RAISE/CALL

Ulazi

U ovom test-scenariju sistem koristi unapred definisane runde (**Round**) koje predstavljaju tipične situacije u **preflop** fazi igre.

Za svaku rundu definišu se sledeći parametri:

- **Hand (ruka):** karte koje igrač poseduje (npr. "AA", "AKo", "QJs", "72o")
 - **Player Position:** pozicija igrača u krugu (veća vrednost = kasnija pozicija = bolja pozicija)
 - **Players:** redosled igrača u partiji (koristi se za izračunavanje pozicije heroja)
 - **Chips:** količina čipova svakog igrača (određuje da li igrač ima *jaku* ili *slabu ekonomiju*)
 - **Current Raise:** trenutni iznos raisa (određuje nivo uloga — *high stakes* ili *normal stakes*)
 - **Pot:** trenutna veličina pota
 - **Big Blind:** osnovni ulog koji definiše minimalne granice raisa
 - **Table Aggressiveness:** ponašanje ostalih igrača
-

Opis slučajeva (Round scenarija)

High Stakes situacije (agresivan sto, veliki ulozi)

- **R1 – Premium hand:**
 - Loša pozicija, slabija ekonomija, ali vrlo jaka ruka.
 - Sistem prepoznaje *High Stakes Premium Hand* → **RAISE**.
 - **R2 – Dobra pozicija, strong hand:**
 - Kasna pozicija kompenzuje nedostatak ekonomije.
 - Aktivira se pravilo *High Stakes Good Position Strong Hand* → **RAISE**.
 - **R3 – Dobra pozicija, medium hand + strong economy:**
 - Srednja ruka, ali igrač ima mnogo čipova.
 - Pravilo *High Stakes Good Position Medium Hand Economy* → **RAISE**.
 - **R4 – Dobra pozicija, medium hand + agresivan sto:**
 - Agresivnost stola pojačava odluku za raise.
 - Pravilo *High Stakes Good Position Medium Hand AggTable* → **RAISE**.
 - **R5 – Loša pozicija, strong hand + strong economy:**
 - Loša pozicija, ali ruka i čipovi su jaki.
 - Pravilo *High Stakes Bad Position* → **RAISE**.
 - **R6 – Loša pozicija, medium hand, slaba ekonomija:**
 - Nijedno pravilo za raise nije ispunjeno.
 - Aktivira se fallback *High Stakes Call Fallback* → **CALL**.
 - **R7 – Dobra pozicija, playable hand, slaba ekonomija, pasivan sto:**
 - Nedovoljno jaka ruka, pasivan sto.
 - Takođe pada na fallback → **CALL**.
-

Normal Stakes situacije

- **R8 – Premium hand:**
 - Klasičan slučaj jake ruke pri normalnom ulogu.
 - Pravilo *Normal Stakes Premium Hand* → **RAISE**.
- **R9 – Dobra pozicija, medium hand:**
 - Pozicija dozvoljava aktivniji pristup.
 - Pravilo *Normal Stakes Good Position Medium Hand* → **RAISE**.
- **R10 – Dobra pozicija, playable hand + strong economy:**
 - Igriva ruka uz dobru ekonomiju.
 - Pravilo *Normal Stakes Good Position Economy* → **RAISE**.
- **R11 – Dobra pozicija, playable hand + agresivan sto:**

- Aktivira *Normal Stakes Good Position AggTablePlayable* → **RAISE**.
- **R12 – Loša pozicija, strong hand:**
 - Ruka dovoljna jaka da prevaziđe lošu poziciju.
 - Pravilo *Normal Stakes Bad Position Strong Hand* → **RAISE**.
- **R13 – Loša pozicija, medium hand + strong economy:**
 - Igrač ima čipove i može sebi da dozvoli rizik.
 - Pravilo *Normal Stakes Bad Position MediumHandEconomy* → **RAISE**.
- **R14 – Loša pozicija, weak hand:**
 - Nema jake ruke ni ekonomije.
 - Pravilo *Normal Stakes Call Fallback* → **CALL**.
- **R15 – Dobra pozicija, weak hand, pasivan sto:**
 - Ni pozicija ni kontekst ne opravdavaju agresiju.
 - Takođe *Normal Stakes Call Fallback* → **CALL**.

Output

Za svaku rundu sistem generiše preporuku:

- **RAISE** – povećati ulog.
- **CALL** – ispratiti trenutni ulog.

Primer rezonovanja – Forward Chaining za FOLD odluke

Ulazi

U ovom test-scenariju sistem koristi unapred definisane runde (*Round*), identično kao prethodni forward chaining.

Cilj je da se proveri ponašanje sistema pri odlučivanju da li igrač treba da **FOLD-uje (odustane)** od igre, u zavisnosti od snage ruke, pozicije, visine raisa i ekonomskih faktora.

Opis slučajeva (Round scenarija)

- **F1 – Ruka ispod praga Playable:**
 - Ruka ispod osnovnog praga, dobra pozicija, jaka ekonomija.
 - Ruka ne dostiže prag "Playable"
 - Sistem prepoznaje **Fold - Below Medium Hand** → **FOLD**.

F1 – Ruka ispod praga Playable:

- Ruka ispod osnovnog praga jačine.
- Igrač ima dobru poziciju i jaku ekonomiju.
- Ruka ne dostiže prag "Playable".
- Aktivira se pravilo *Fold - Below Medium Hand* → **FOLD**.

F2 – Veliki raise protiv barem medium ruke:

- Ruka je barem medium kategorije.
- Igrač u lošoj poziciji, suočen sa velikim raise-om.
- Jaka ekonomija ne kompenzuje slabost ruke.
- Aktivira se pravilo *Fold - Big Raise vs Not Medium Hand* → **FOLD**.

F3 – Loša pozicija i ne-medium ruka:

- Ruka iznad medium praga.
- Pozicija loša, ekonomija slaba.
- Trenutni raise mali, ali ruka ne opravdava nastavak.
- Aktivira se pravilo *Fold - Bad Position vs Not Medium Hand* → **FOLD**.

F4 – Barem medium ruka uz slabost:

- Ruka ispod medium praga.
- Dobra pozicija, ali slaba ekonomija i pasivan sto.
- Nema podrške iz ekonomije ni konteksta stola.
- Aktivira se pravilo *Fold - Below Medium Hand with Weakness* → **FOLD**.

F5 – Kontrolni slučaj (nema fold uslova):

- Ruka dostiže prag medium jačine.
- Nijedan fold uslov nije ispunjen.
- Sistem ne prepoznaje potrebu za fold-om.
- Aktivira se pravilo *Fold - Default No Fold* → **NO FOLD**.

Output

Za svaku rundu sistem generiše preporuku:

- **FOLD** – odustati od igre.
 - **NO FOLD** – nastaviti igru, jer nijedan od uslova za fold nije ispunjen.
-

