

1. Sadržaj predloga

Tim

Miloš Matunović RA124/2021

Zadatak

Tema projekta je izrada sistema za **federativno učenje** sa ciljem **predikcije pobjednika NBA turnira 2026. godine**.

Sistem spaja koncept federativnog mašinskog učenja i aktorskog radnog okruženja, tako da se treniranje modela obavlja distribuirano na više čvorova, dok aktorski sistem omogućava asinhronu komunikaciju, toleranciju na greške i skaliranje na više mašina.

Federativno učenje

- **Algoritam:** FedAvg kao osnovni, sa opcijom korišćenja FedProx varijante.
- **Skup podataka:** istorijski NBA mečevi i statistike, dostupni putem javnih izvora (Kaggle, Basketball-Reference, nba_api).
- **Distribucija:** svaki tim (čvor) trenira lokalni model na svojim podacima, a ažuriranja modela se dele i agregiraju kroz provider ili peer-to-peer režim.
- **Evaluacija:** log-loss i Brier score kao glavne metrike, kao i tačnost predikcije plej-of serija i poređenje sa centralizovanim baseline modelom.

Implementacija aktorskog radnog okruženja

Biće implementirani sledeći elementi:

- **Obavezni elementi:** aktori, asinhrono slanje i primanje poruka, sanduče (mailbox), promene ponašanja aktora, lifecycle događaji (kreiranje i gašenje), udaljena komunikacija.
- **Dodatne funkcionalnosti:**
 - supervizija aktora (strategije restartovanja),
 - perzistencija stanja (čuvanje modela i poruka),
 - klasterovanje u dva moda: provider (centralni agregator) i peer-to-peer (gossip protokol),
 - CRDT strukture (LWW-Map i PN-Counter) za održavanje konzistentnog stanja u distribuiranom okruženju.

Aktori

U sistemu učestvuju sledeće vrste aktora:

- **TeamNode:** trenira lokalni model i čuva rezultate.
- **Aggregator:** prikuplja i kombinuje lokalne modele.
- **Evaluator:** meri performanse i poredi sa baseline-om.

- **ScheduleWatcher:** pribavlja raspored i rezultate iz dostupnih izvora podataka.
- **CrdtReplicator:** zadužen za sinhronizaciju stanja korišćenjem CRDT struktura.

Poruke koje se razmenjuju:

- `TrainRequest(season_window, round_id)` – zahtev za pokretanje runde treniranja.
- `ModelUpdate(round_id, weights_digest, metrics)` – ažuriranje modela i prateće metrike.
- `CrdtMerge(delta)` – delimična sinhronizacija stanja.
- `EvalRequest(split_id) / EvalReport(metrics)` – evaluacija modela i izveštaj.
- `NodeJoin/Leave, HealthPing/HealthAck` – poruke za održavanje članstva i proveru dostupnosti.

Detalji implementacije

- **Jezik i okruženje:** Python (asyncio, multiprocessing).
- **Mašinsko učenje:** PyTorch i scikit-learn.
- **Federativni sloj:** sopstvena implementacija FedAvg/FedProx unutar aktorskog sistema.
- **Mrežna komunikacija:** gRPC ili WebSocket za udaljene poruke.
- **Perzistencija:** SQLite ili DuckDB za čuvanje lokalnih stanja i modela.
- **CRDT:** sopstvena implementacija jednostavnih struktura (LWW-Map, PN-Counter).
- **Testiranje i merenja:** pytest za jedinичne testove, skripte za merenje latencije i performansi.