

Projektni seminar

Razvoj eksperimentalnega sistema za simulacijo sodelovanja LLM-jev na napovednih trgih

Uporabniška dokumentacija

Avtor: Marjan Meglen

Mentor: dr. Aleksandar Tošić

Somentor: dr. Niki Hrovatin

UP FAMNIT, Koper

Maj 2026

Kazalo

1	Namen dokumenta	2
2	Predpogoji	2
3	Osnovni potek uporabe	2
4	Priprava vhodnih podatkov	2
5	Izbira trga	3
6	Izbira konfiguracije	3
7	Zagon testnega eksperimenta	4
8	Zagon z LLM modelom	4
9	Pregled rezultatov	4
10	Analiza rezultatov	5
11	Pogoste napake	6
12	Priporočila za uporabo	6

1 Namen dokumenta

Ta dokument opisuje uporabo sistema za izvajanje LLM napovedi na napovednih trgih. Namenjen je uporabniku, ki želi pripraviti podatke, izvesti eksperiment, preveriti rezultate in nato uporabiti analitični del projekta.

2 Predpogoji

Za uporabo sistema je potrebno:

- nameščen Python 3,
- dostop do projektne mape,
- pripravljene vhodne JSON datoteke,
- po potrebi dostop do Ollama-kompatibilnega LLM strežnika,
- pri uporabi zaščitenega API strežnika nastavljena spremenljivka `HIVECORE_API_KEY` ali datoteka `.env`.

Za hiter preizkus ni potreben dejanski LLM strežnik, saj sistem podpira ponudnika `mock`.

3 Osnovni potek uporabe

Tipičen potek dela je:

1. priprava ali preverjanje vhodnih podatkov,
2. izbira trga in konfiguracije,
3. zagon eksperimenta,
4. pregled datoteke `predictions.json`,
5. izvedba analize rezultatov.

4 Priprava vhodnih podatkov

Vhodne datoteke za eksperiment so v mapi:

```
llm-trader/input_data/
```

Uporabnik naj preveri, da so prisotne naslednje datoteke:

- `markets.json`,
- `daily_data.json`,
- `news.json`,
- `instructions.json`,
- `configurations.json`.

Če je treba podatke ponovno zbrati, se uporabijo skripte v mapi:

`data-collection/data_collection_runners/`

Končni izvoz podatkovnega dela se nahaja v:

`data-collection/collected_data/6_llm_trader_export/`

Te datoteke se nato uporabijo kot vhod za `llm-trader`.

5 Izbira trga

Vsak trg ima svoj `market_id`. Primeri trgov v trenutnem naboru so predsedniška inauguracija, volitve za župana New Yorka ter odločitve ameriške centralne banke o obrestnih merah. Vsak od teh trgov ima svoj identifikator v vhodni datoteki.

Celoten seznam je v datoteki `llm-trader/input_data/markets.json`.

6 Izbira konfiguracije

Konfiguracija določa, katere informacije model vidi v pozivu. Sistem uporablja tri faktorje:

- prikaz tržnih verjetnosti: da ali ne,
- prikaz novic: novice trenutnega dne ali brez novic,
- opis drugih udeležencev: LLM agenti, ljudje ali brez opisa.

V trenutnem naboru je pripravljenih 12 konfiguracij. Celoten seznam je v:

`llm-trader/input_data/configurations.json`

7 Zagon testnega eksperimenta

Najprej se premaknemo v mapo `llm-trader`:

```
cd llm-trader
```

Za preverjanje, da cevovod deluje, uporabimo `mock` ponudnika:

```
python3 run_experiment.py \  
  --market-id who_will_be_inaugurated_as_president \  
  --configuration-id 1 \  
  --provider mock
```

Ta zagon ne kliče zunanjega modela. Za vsak dan vrne deterministično vrednost `0.5`, zato je primeren za preverjanje podatkov, argumentov in zapisovanja rezultatov.

8 Zagon z LLM modelom

Za dejanski LLM zagon se uporabi ponudnik `ollama`:

```
python3 run_experiment.py \  
  --market-id who_will_be_inaugurated_as_president \  
  --configuration-id 1 \  
  --provider ollama \  
  --model-name qwen3.5:4b
```

Privzeti naslov strežnika je:

```
http://hivecore.famnit.upr.si:6666
```

Če je treba uporabiti drug strežnik, dodamo argument:

```
--ollama-base-url http://naslov-streznika:vrata
```

9 Pregled rezultatov

Po končanem zagonu sistem ustvari novo mapo v:

```
llm-trader/results/
```

Ime mape ima obliko:

```
<market_id>_<configuration_id>_<YYYYMMDD_HHMMSS>
```

Glavna datoteka je:

```
predictions.json
```

V njej je za vsak datum shranjeno:

- vprašani trg in uporabljena konfiguracija,
- datum napovedi,
- utemeljitev modela,
- napovedana verjetnost `p_yes`,
- komplementarna verjetnost `p_no`,
- celoten poziv,
- celoten pogovor z modelom.

10 Analiza rezultatov

Analiza se nahaja v mapi `analysis/`. Priporočeni postopek je:

```
cd analysis
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install -r requirements.txt
jupyter lab
```

Nato se odpre zvezek:

```
notebooks/preliminary_statistical_analysis.ipynb
```

Analiza pričakuje rezultate v:

```
../llm-trader/results/*/predictions.json
```

Izhodi se shranijo v:

- `analysis/outputs/figures/` za slike,
- `analysis/outputs/tables/` za CSV tabele,
- `analysis/report/` za LaTeX poročilo in tabele.

11 Pogoste napake

Neznan market_id

Če sistem javi, da trg ne obstaja, je treba preveriti, ali je uporabljeni market_id zapisan enako kot v markets.json.

Neznan configuration_id

Če konfiguracija ne obstaja, je treba preveriti datoteko configurations.json. Identifikatorji so zapisani kot nizi, na primer "1".

Napaka v vhodnih podatkih

Sistem pred začetkom preveri manjkajoča polja, datume, reference in verjetnosti. Če se pojavi validacijska napaka, je običajno treba popraviti eno od datotek v input_data/.

LLM odgovor ni veljaven JSON

Model mora vrniti samo JSON objekt z utemeljitvijo in vrednostjo probability_yes. Če odgovor ni pravilen, sistem zapiše diagnostično datoteko v mapo trenutnega zagona.

Strežnik ni dosegljiv

Pri ponudniku ollama je treba preveriti naslov strežnika, omrežno povezavo, API ključ in argument -timeout-seconds.

12 Priporočila za uporabo

Pred daljšimi zagoni je smiselno vedno izvesti kratek mock zagon. Pri dejanskih LLM zagonih je priporočljivo izvajati en trg in eno konfiguracijo naenkrat, ker se rezultati shranjujejo v ločene mape in so tako preglednejši. Po vsakem večjem sklopu zagonov je priporočljivo odpreti predictions.json in preveriti, ali so pozivi, datumi in verjetnosti zapisani pričakovano.