Datapao Take-home Test

The Big IMDB quest

András Novoszáth

2024

Topics

1. Quick recap
2. Approach & trade-offs
3. Environment & code structuring
4. Data collection & ranking calculations
5. Data storage & pipeline

Quick recap

Goal

Generate list of top 20 movies by adjusted ranking

Deliverables

- Functions + tests
  - Scraper (title, rating, # ratings, # Oscars)
  - Oscar calculator: rating reward based on # of Oscars
  - Review penalizer: penalize rating with few votes
- Dataset
- Instructions

Approach & Trade-offs

- Python 3.10, BeautifulSoup, pandas, pandera, pytest
- Environment & dependency management: Poetry
- Configurations: `.env`
- Single-script module
- Testing: pytest
- Data: JSON

Environment & Code Structuring

- Dependency management
  - `venv` & `pip`: robust
  - `uv`: speed, dependency resolution
  - Makefile
- Code structure: breaking up the script into modules

```
src
    main.py
    data_collector.py
    ranking_calculators.py
test
    ...
```

- Package & CLI application
- Configuration: TOML or YAML

Data Collection & Ranking Calculations

- Scraping
  - Use the IMDB API
  - Scrapy or other scraping library
  - Validation: fallback mechanism
- Penalization & Oscar calculator
  - Dynamic mapping and scoring sourced from config
  - Relative scoring
  - Domain/project-specific scoring
- Test
  - Breaking up functions for more granular testing
  - Integration test
  - Document test functions

Data storage & Pipeline

- Data storage
  - Dynamic file path + naming construction
  - Using `Pathlib` (vs `os.path`)
  - Record-based JSON structure
  - Two separate data for the different orderings
- Data pipeline
  - Schedule regular data updates
  - Ingest data into a database
  - Logging & Monitoring

Thank You

linkedin.com/in/andrasnovoszath