Movie Recommendation Service

Team Jurassic Park

Cora, Harsh, Rohan & Kartik

Outline

- Introduction
- Approach
- Challenges
- Scope and Future work
- Teamwork

Introduction

- Our project involved building a production grade recommendation service to handle a reasonably large number of users.
- Once the service was able to service over 4.7 Million recommendations this past month and haven’t experienced any significant downtime.
- We recommend 20 unique movies to each user, these recommendations are based primarily on ratings event data provided to us.
- We see that our time to service a query is roughly around 400ms.

Approach

- Data collection: We collected and processed data from the ratings event stream and user data
- Model used: Neural Network
- Inference: Models outputs saved and subsequently queried to reduce latency.
- Infrastructure and Deployment: Kubernetes to deploy our model, our CI pipeline enabled loading appropriate files for modelling and the publishing an artifact to DockerHub
- Monitoring: We used Grafana to monitor the health of our service

Challenges

Modelling & Inference

- Package (Surprise) vs Implementation from scratch
- Different modelling techniques -
  - SVD vs Neural Networks
- Techniques to improve efficiency -
  - Precomputing recommendations

Data and Provenance:

- Data cleaning and formatting for ML tasks.
- Versioning stream data.

Challenges

Infrastructure:

- Dockerizing the inference server
- Running MicroK8S on AWS
- Load balancing

CI pipeline:

- End-to-end setup from commit to container generation
- Versioning containers based on model to be deployed

Future Scope

- Infrastructure autoscaling
- Automating continuous deployment
- Building a versioning tool for streaming data.
- Parallelizing precomputation of recommendations
- Visualization for AB test validation

Teamwork

Clear work division and frequent team meeting

- No overlap of work assignment (Trello is a great tool)
- Meet everyday when working on the project

Clear Code Versioning

- Strict code review / pull request / CI

Poor time management especially for Milestone 3

- Not good at estimating time when we don’t know the expected outcome of our experimentation

Future collaboration

- Check expected time with TA or professors before starting the task

Q&A