Team 14 - Strangers
Robust Movie Recommendation System

By Joong Ho Choi, Ritu Gala, Charly Jin,
Tanya Bhatnagar

Pulling data from Kafka

General:

- Learning Kafka: partitions, offsets, group id
- Tracked offset using indices and append
- Split data to separate csvs
- Dropped bad data: time, movie, user, rating

Fine tuning:

- Batch size changing - High Throughput
- Decision to change libraries?
- Large Data collection problems downstream

Data preprocessing+optimization+storage

-Formatting raw rating and mpg data into workable format

-Extraction of rating from mpg data based on thresholds derived from EDA

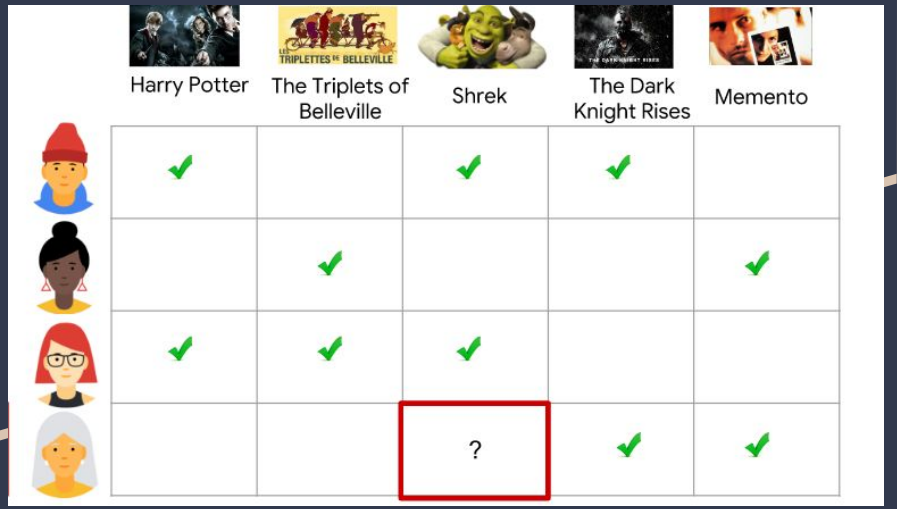
-Code speed optimization via PySpark

-Compression of csvs as zip files with appropriate naming (date,etc.)

PySpark

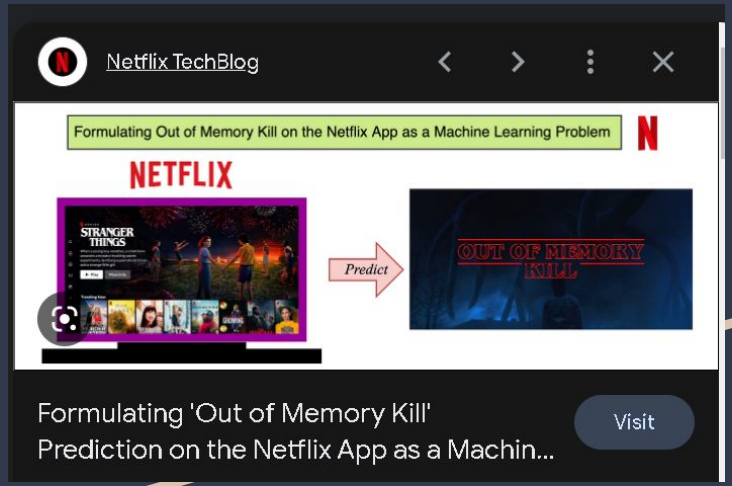
Recommendation model: Anti-TestSet

We first train an SVD algorithm on the whole dataset, and then predict all the ratings for the pairs (user, item) that are not in the training set



Advantage: Time for predicting Recommendations for a user is **O(1)**

Out of Memory Error

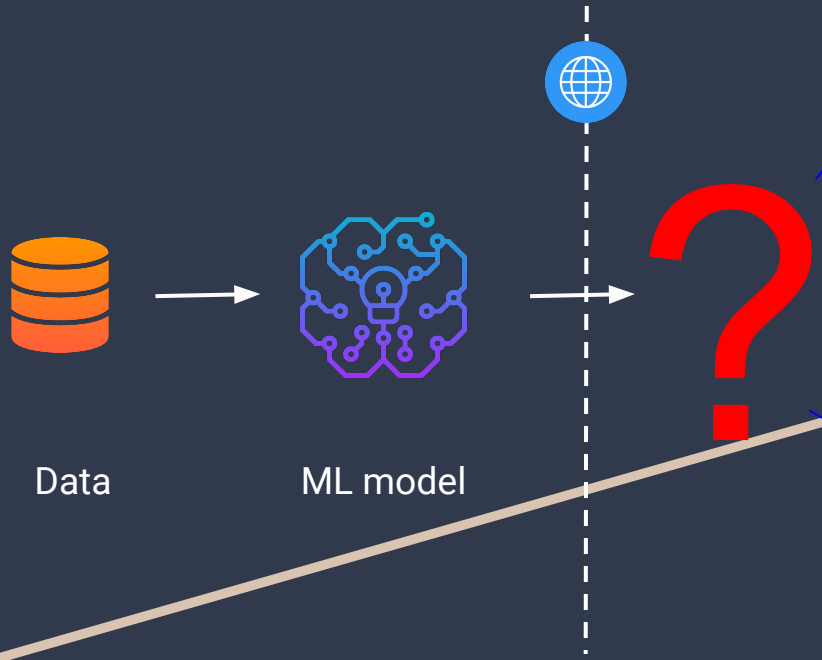


Reason: Most users obviously haven't rated most movies.
=> Anti test is much bigger
=> Building the antitest set kills the entire process

Solution: Create Predictions on the go

Tradeoff: increased time required per response (O(n) per user)

Deployment on Flask



Serve Application

Communicate between client and server

Issue: **Response timeout, handle bad requests**
Solutions: Dropped sparse rows
Exception handling

Load balancer + Experimentation tracking

-4 servers

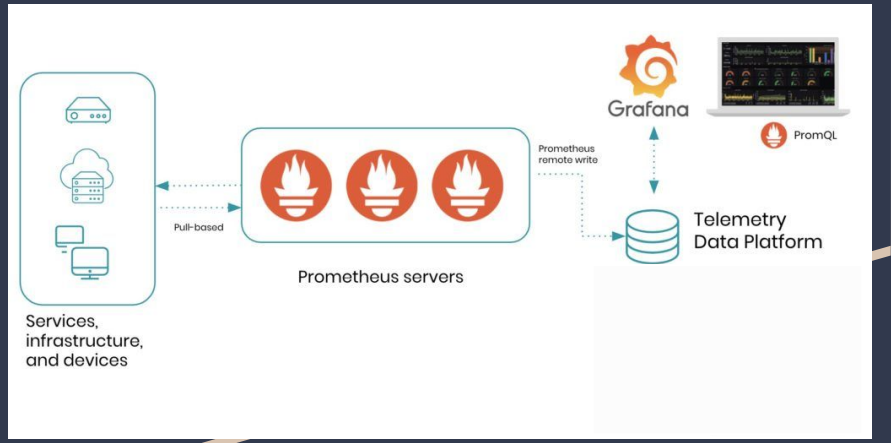
*hosting both NMF and SVD

*random traffic split based on uniform distribution

-A/B testing through paired t-test

-MLFlow to track the models and view exactly what model and dataset have been used to create a given prediction.

Monitoring: Prometheus & Grafana



Prometheus: Collect metrics

Grafana: Transform metrics to visualizations

Pros:

Easy Integration

Powerful visualization

Cons:

Limitations on dashboard organization and design

Reflection

Good

- Consistent weekly meetings
- Work allocation by specializations and learning

Bad

- Legacy code and design constraints

Team

Found great friends!