

A top-down photograph of a red bowl filled with white popcorn, sitting on a wooden surface. Several popcorn kernels are scattered around the bowl. A black remote control is positioned to the right of the bowl, also on the wooden surface. The background is a light-colored wooden surface with a visible grain.

# MOVIE RECOMMENDER SYSTEM

---

OPTIMIZERS:

Charitha Basani

Deepthi Moti

Jaya Kumar

# PROCESS OVERVIEW



Data Source



ETL/ELT Process



Architecture



Recommendation Engines



Challenges



References

# **PROBLEM STATEMEN T**

---

- In this Internet based world, with enormous collection of movies available, people have a hard time selecting the movies they actually want to see





Users Be Like

**This is exactly what I wanted**

## **SOLUTION**

- In order to ease the process of selecting a movie to watch from the large number of available movies, we suggest a Hybrid recommender system based on several features that would predict the movies a user would want to watch



# DATA SOURCE



## Movielens Dataset

• 26,000,000 ratings, 750,000 tags, 45,000 movies & 270,000 users

• Source – TMDb API

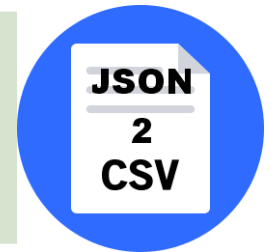
• Features – vote\_count, vote\_average, cast, crew, genre, title etc.

700 users

# ETL/ELT PROCESS



## Data Wrangling



## Exploratory Data Analysis

Most Popular, voted and critically acclaimed movies

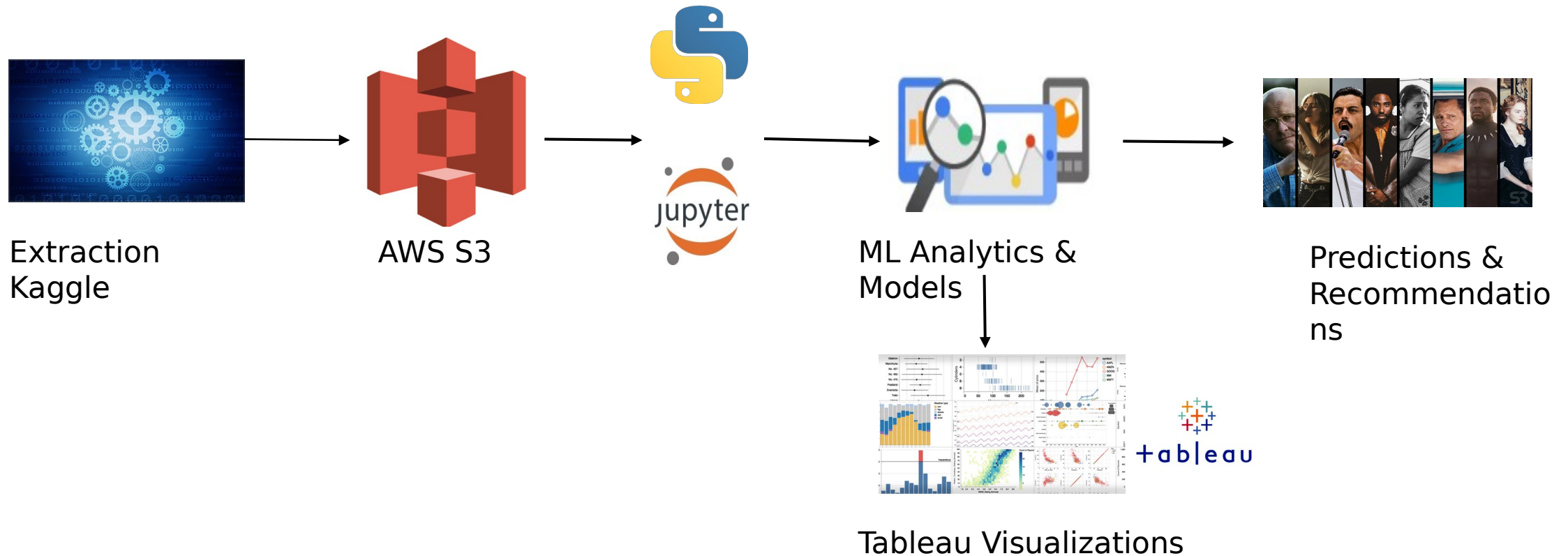
Movie Release Dates

Does the number of spoken languages influence the success of the movie?

Does the returns earned depend on the duration of a movie?

Summary statistics on Budget feature

# ARCHITECTURE





# RECOMMENDATION ENGINES

- Simple Recommender
- Content Based Recommender
- Collaborative Filtering
- Hybrid Engine



# SIMPLE RECOMMENDER

---

Offers  
generalized  
recommendations  
to every user

Based on movie  
popularity and  
genre

No personalized  
recommendations

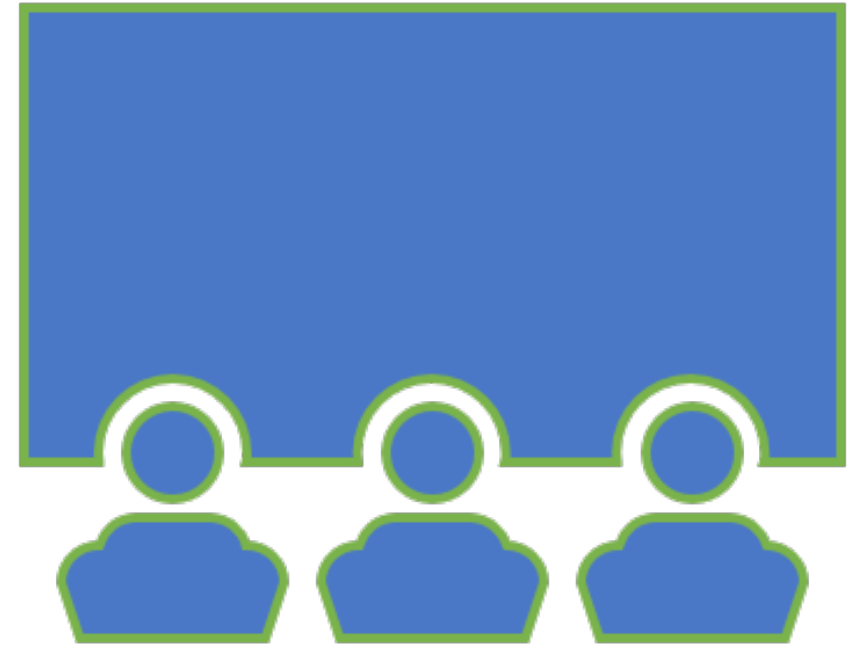
# CONTENT BASED RECOMMENDER

- Gives personalized recommendations
- Use of TF-IDF Vectorizer
- Can be built on multiple content-based recommendations
- Predicts movies only close to certain movies

# COLLABORATIVE FILTERING

---

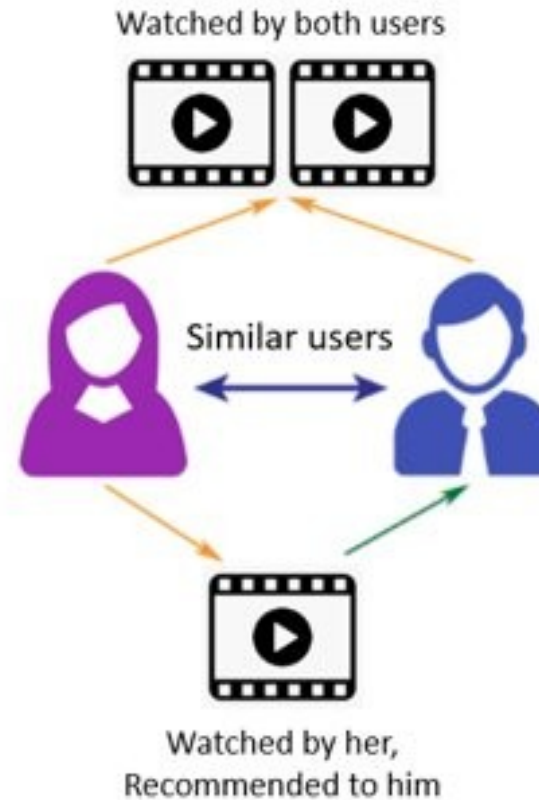
- Limitations from content-based recommendations
- Recommend movies watched by similar users
- Use of Singular Value Decomposition (SVD) algorithm



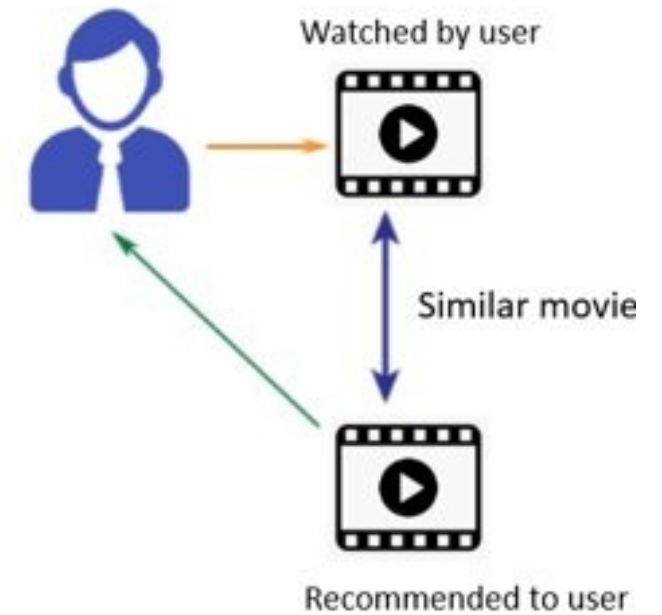
# HYBRID ENGINE

- Combination of both content based and collaborative filtering

## Collaborative Filtering



## Content-Based Filtering





# CHALLENGES

---



Very large  
dataset



Cold start



Latency

# REFERENCE S

- <https://www.kaggle.com/rounakbanik/movie-recommender-systems>
- <https://www.geeksforgeeks.org/python-implementation-of-movie-recommender-system/>
- [https://medium.com/@springboard\\_ind/how-netflixs-recommendation-engine-works-bd1ee381bf81](https://medium.com/@springboard_ind/how-netflixs-recommendation-engine-works-bd1ee381bf81)



# QUESTIONS ?

