Netflix Recommendation System

Team 1

Praneeth Reddy

Sonali Chaudhari

https://github.com/reddyse/Big-Data-Engineering-Using-Scala.git

Goals Of the Project

- With the help of user data and ratings, generate a relevant suggestion based on past available dataset.
- Suggesting the movie with highest predicted rating to a particular user.
- Meeting the deadline of the project.

Optional

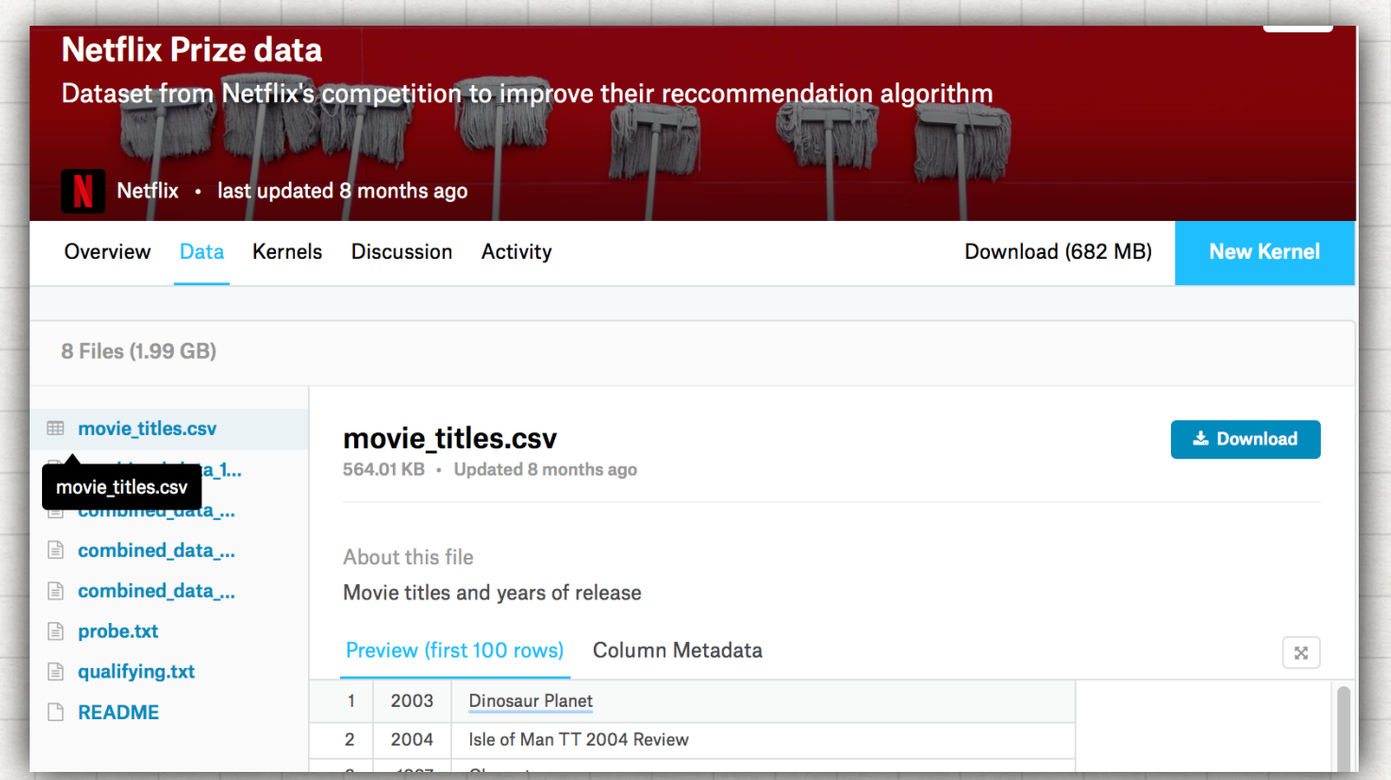
- Get the most trending items using Twitter stream.

Data Sources

- The data source we used was taken from kaggle - https://www.kaggle.com/netflix-inc/netflix-prize-data/data and has about 17770 movie records.

- Data Size - 1.99GB

- Movie dataset
- Training dataset
- Probe dataset
- Qualifying dataset



Use Case(s)

- The rating provided by the other users will be used to predict ratings for the movies in qualifying dataset.
- The user will be recommended a movie based on the rating prediction(highest rated movie).
- The user will have a profile that will contain his list of favorites and we will be able to provide movie recommendations based on artists, past history and other related information.

Milestone

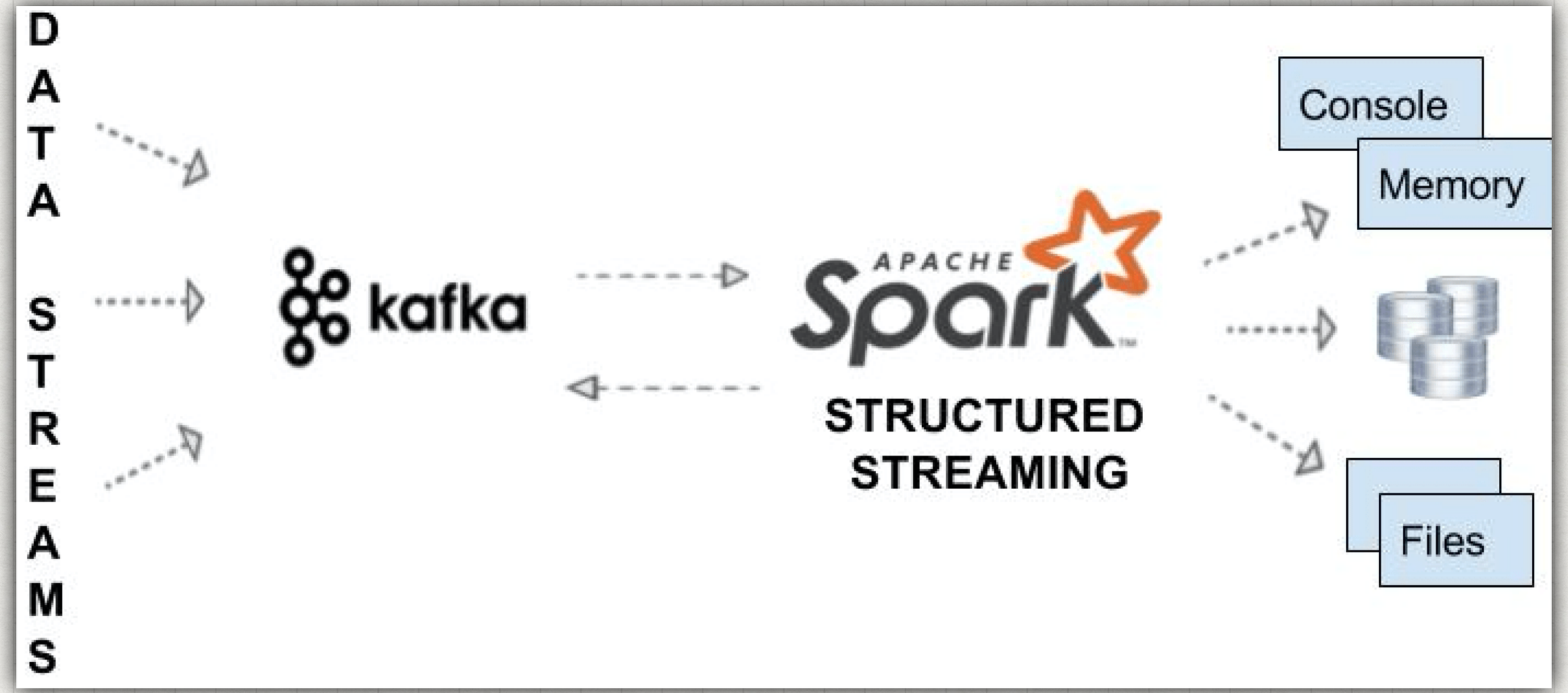
- Key dates - Using Agile Software Development

Explore, Write and Test Apache Kafka producers and consumers.	3/29
Exploring and implementing Play, Actor model and spark and unit tests	4/15
Integrating Spark and Play	4/18
Functional and load testing /self acceptance	4/23

Clean and ready the data and Explore Apache Spark.	3/29
Explore various algorithms that can be applied for predictive analytics.	4/15
Integrating Spark and Play	4/18
Functional and load testing /self acceptance	4/23

Methodology

- We will be implementing content based recommendation system.
- Use Avro for standardization of data.
- Build an reactive application using Play framework and Actor Model.
- Use of Spark and Spark Mlib to come up with the recommendations for the user.


DATA STREAMS
kafka
APACHE Spark
STRUCTURED STREAMING
Console
Memory
Files

Scala Programming

- Kafka Producers and Consumers
- Play
- Spark
- Spark MLib

Acceptance Criteria

- Application should be able to handle at least 2500 requests simultaneously and the model should be scalable to add new data sources as and when required.

- Achieve >90% accuracy using probe dataset.

Thank you!