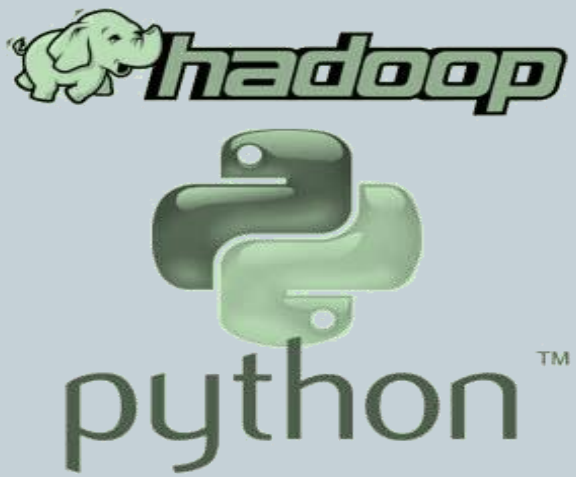
CS 579
ONLINE SOCIAL NETWORK ANALYSIS
FINAL PROJECT – FALL 2015

0

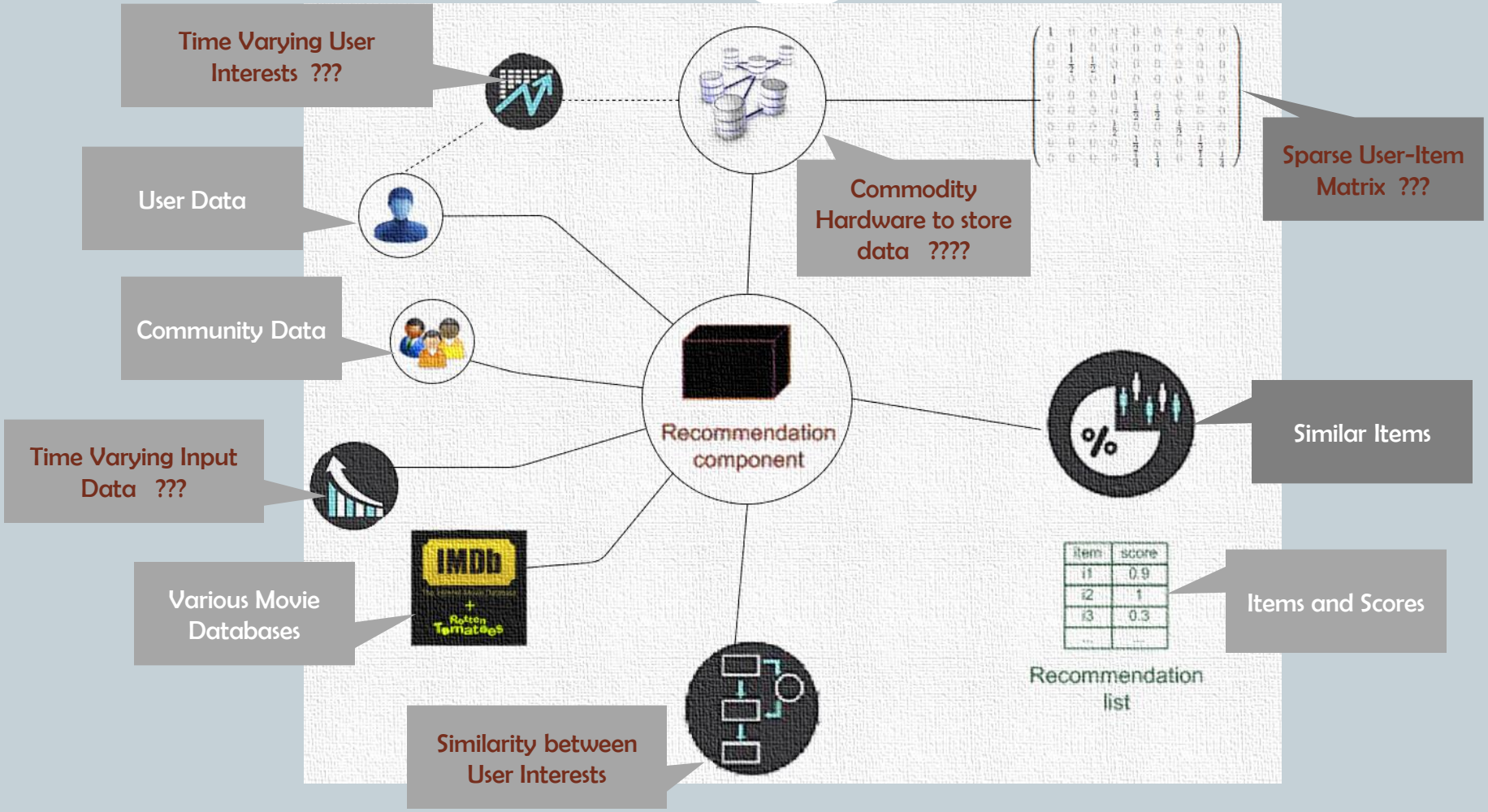
MOVIE RECOMMENDATIONS USING MAP REDUCE



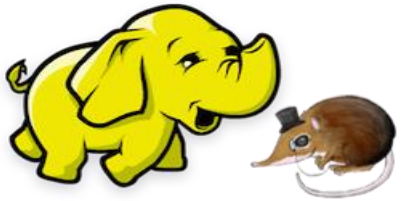
PRESENTED BY
J V P S AVINASH
RAKSHITH MUNIRAJU

THE PROBLEM

1



OUR APPROACH



01 DATA COLLECTION

We collected movies data from Rotten Tomatoes. We collected the data for last two years.

02 DATA STORAGE

We moved the data from local file system to Hadoop Distributed File System.

03 MAP REDUCE WORK FLOW

For the data stored in HDFS, we designed a Mapper, Combiner and Reducer to implement the MR Job

04 FIND MOST SIMILAR ITEMS

We applied various similarity measures to find the items which are more similar.

05 REPORTING RESULTS

We used the similarity measures to find the similar items and generated results based on them and compared by running them in local.

What data records do we collect ?

Movies/ Ratings/User Data

- ✓ **User Name**
- ✓ **Movie Name**
- ✓ **Rating given by user**

Rotten Tomatoes API

How much data?

- ***Top 100 Best Movies*** *rated in years 2014 and 2015.*
- ***Page Limit of 5*** *per each movie*
- *An approximate of 100 users per each movie are recorded.*

20,000 ratings corresponding to all the movies are extracted from API.

How did we collect?

- ***Rotten Tomatoes Movie Names and URLs*** *was pulled out of API. All the movie ratings with user details are not returned. So, we used a web crawler to extract the rating information.*

Where did we store?

- *All the data are initially extracted to the local file system and then moved into Hadoop Distributed File System*

RESULTS

4

1
Data Storage
Moved the data from local environment to HDFS. Thus the complexity in storing huge data is no more.

3
Map Reduce
By using Map Reduce as the framework, we cut down time taken to run to less than 2 minutes.

2
Similarity Measures
Applied various similarity measures on the item-item ratings to find the similar items to each other.

4
Efficiency
The algorithm reduces the complexity of storing the data and performing analysis on it. Efficiency got increased.

CONCLUSION

Lessons

- Learned how to extract data from Rotten Tomatoes API.
- As it does not return the expected information like user details and the rating given by them, we wrote a web crawler which extracts the data from the given page URL.
- Also, the API failed to provide the details of all the movies in a particular year. So, we got restricted to only top 100 movies.
- Also, we observed that in all the top 100 movies, there are unequal distribution of user ratings.
- So, from our data we observed that there is equal distribution of ratings even though of top rated movies are taken into account.
- Integrating **Python with Hadoop** which is used for Big Data Analytics
- Using various features of **mrjob** package in Python.

Insights

- When finding the similarity between items, we considered various similarity measures like Correlation, Cosine, Jaccard and their generalizations.
- For a movie having a genre of “Thriller”, we are able to recommend movies which are of same genre.
- Hadoop provides a job environment which contains the list of all executed jobs. It contains all the resource management information, scheduling information etc.
- The details of that information are given in report.