MOVIE RECOMMENDATION SERVICE (PUMKINMETER).

COLLABORATIVE FILTERING IN PYSPARK

BUAN 5315: BIG DATA ANALYTICS
PASCAL NTAGANDA

SOURCE:
HTTPS://WWW.CODEMENTOR.IO/@JADIANES/BUILDING-A-RECOMMENDER-WITH-APACHE-SPARK-PYTHON-EXAMPLE-APP-PART1-DU1083QBW

INTRODUCTION

- Movie recommender uses the concept of COLLABORATION FILTERING.

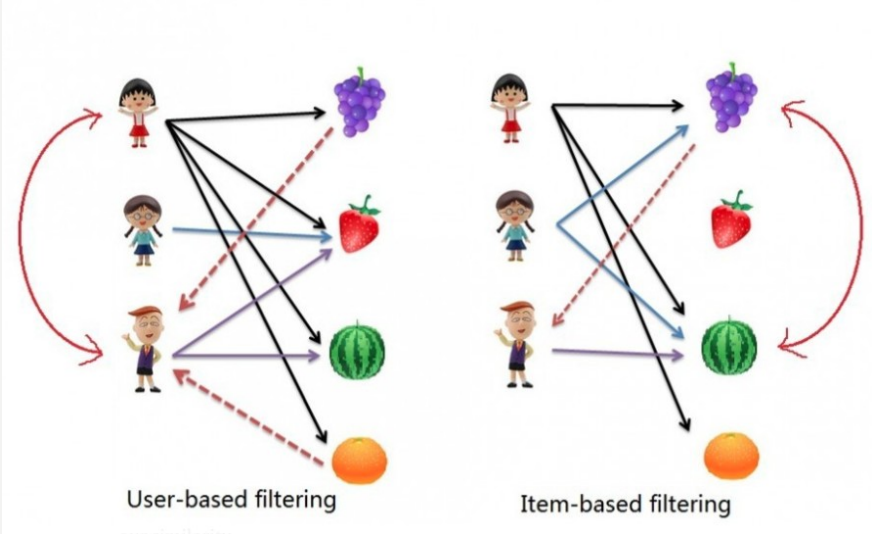
Collaboration filtering: recommends movies and shows based on customer interests, in relation to the interests and preferences of other member users to the network .

- Tool used: Pyspark with Machine Learning library (MlLib), enable real-time and large-scale data processing.
- Done in web based interactive computing platform → Jupyter notebooks.

DATASET (Grouplens Research website)

- Small dataset: 100,000 ratings, 3600 tag application on 9,000 movies by 600 users.
- Large dataset: 27,000,000 ratings,1,100,000 tag applications on 58000 movies by 280,000 users.

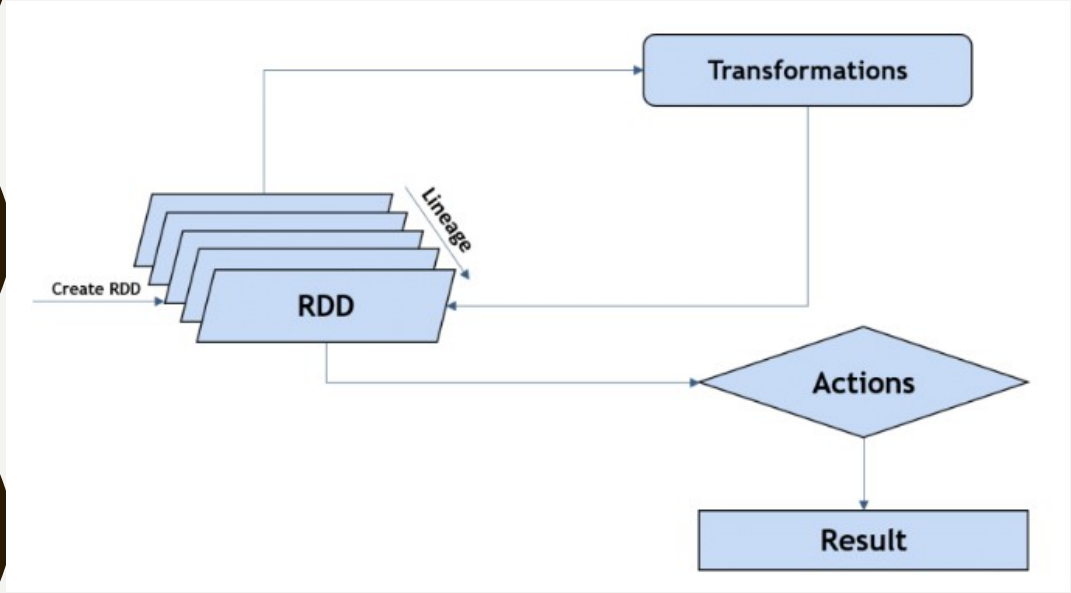
LI



https://link.springer.com/article/10.1007/s11227-020-03266-2

DATA PROCESSING.

Data processing refers to the manipulation of data in terms of raw data conversion, flow of data in processing unit and memory, and out put (and transformation output).



https://blog.knoldus.com/things-to-know-about-spark-rdd/

Resilient Distributed Datasets (RDDs) used due to the need for:

- Low level data transformation.
- Type safety and fast
- In memory computation.
- Need for parallelization especially since a virtual machine is used hence could reduce processing time by creating different nodes or data points.
- Most importantly, RDDs support data transformations like filtering and union; actions like fetching elements and counting elements.

MACHINE LEARNING USING PYSPARK-MLLIB

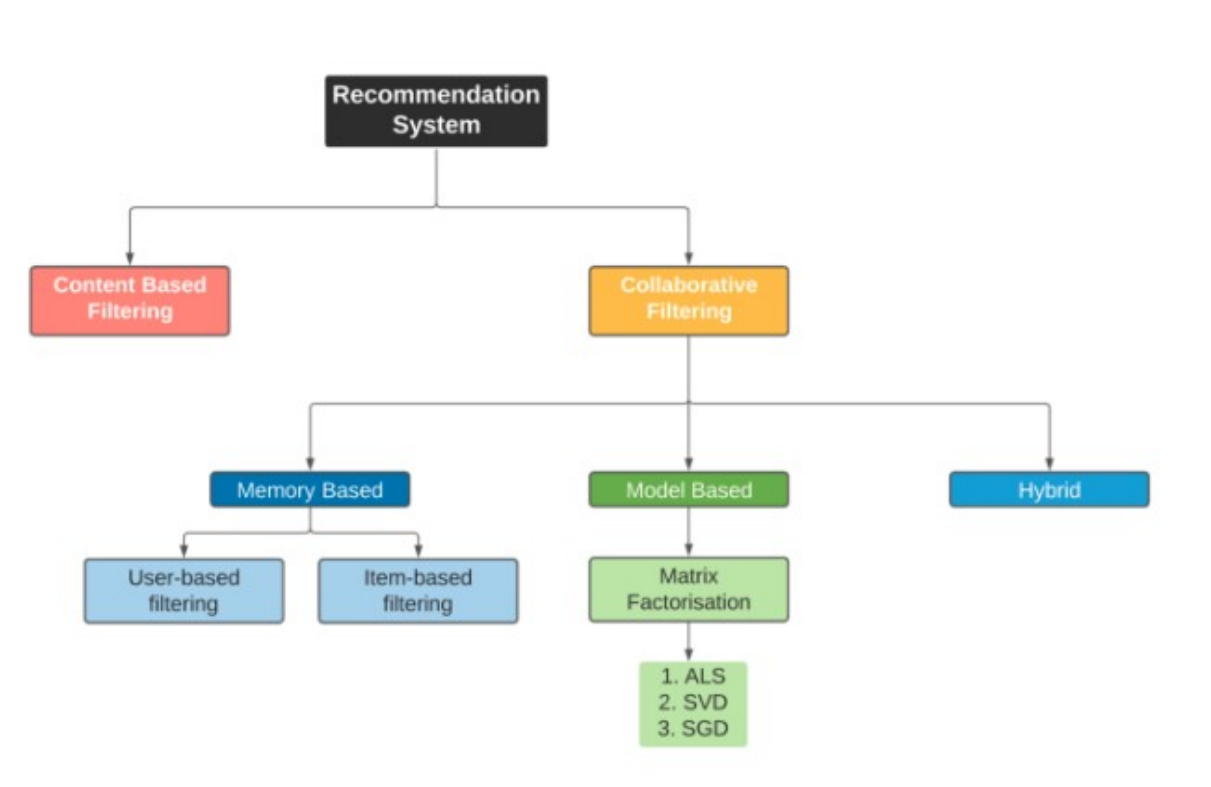
ALTERNATE LEAST SQUARE MODEL:

This is an algorithm used to find the similarities between the fitted data.

It creates lower dimensionality matrices with a product equal to the original one, hence expressing similar features.

- The machine learning library in Pyspark is used to train the ALS model which works by factoring the user to item matrix, into user to feature matrix.

Approaches to recommendation including ALS described as matrix factorization.



https://datasciencemadesimpler.wordpress.com/tag/alternating-least-squares/

https://towardsdatascience.com/build-recommendation-system-with-pyspark-using-alternating-least-squares-als-matrix-factorisation-ebe1ad2

RESULTS

→ After training, evaluating and validating the ALS model from the smaller dataset, it is also trained with the large dataset.

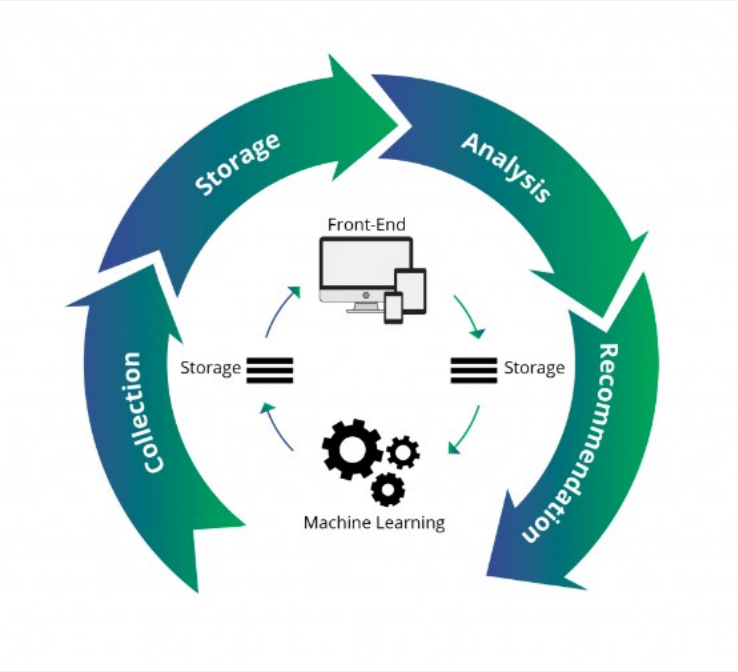
The Recommender algorithm is then run on the new user data ratings added to the original data at different scenarios created:

- FULL dataset, filtering out movies with less than 25 ratings (meaning 25 or more ratings)
- FULL dataset, filtering out movies with less than 100 ratings (meaning 100 or more ratings)

The recommender algorithm provides efficient results with recommendations similar to the user ranking data added to the algorithm.

Sample results

- TOP 15 recommended movies (with more than 25 reviews): ('Death on the Staircase (Soupçons) (2004)', 3.04336670799772, 130) ('"Godfather', 3.0033750435078446, 60904) ('The Godfather Trilogy: 1972-1990 (1992)', 2.9843748479297076, 421) ('The Adventures of Sherlock Holmes and Dr. Watson: Bloody Signature (1979)', 2.980737231675196, 141) ('"Civil War', 2.955527279394424, 431) ('Planet Earth (2006)', 2.9483240239893327, 1384) ('"Silence of the Lambs', 2.94218766564331, 87899) ('"Godfather: Part II', 2.9337260156261262, 38875) ('George Carlin: You Are All Diseased (1999)', 2.9294454315304694, 163) ('"Shawshank Redemption', 2.926813364476317, 97999) ('Harakiri (Seppuku) (1962)', 2.920121358352784, 679) ("Schindler's List (1993)", 2.917782358496197, 71516) ('Rear Window (1954)', 2.916815819306585, 22264) ('Wallace & Gromit: The Best of Aardman Animation (1996)', 2.9096256056171192, 9674) ('Bill Hicks: Revelations (1993)', 2.906212543056636, 158)

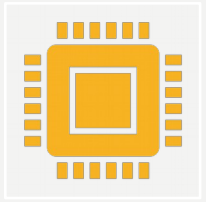


https://www.smarthint.co/en/ai-product-recommendation-engine/

FINAL REMARKS

Understanding the procedure is a little different from obtaining results.

During the process, it shows that even though the use of Resilient Distributed Datasets is effective to reduce processing time and work on large datasets, there is still need of some form of high computing power and storage.



This is due to the large dataset used in the final model.

Support in terms of computing power and storage is still needed but still outweighs the costs of not using Pyspark

References

[1]Dianes, A Jose, *Building a Movie Recommendation Service with Apache Spark & Flask - Part 1, 2015*.
https://www.codementor.io/@jadianes/building-a-recommender-with-apache-spark-python-example-app-part1-du1083qbw
[2]Ramsingh, J. (2022). PySpark toward Data Analytics. In *Big Data Applications in Industry 4.0* (pp. 297-330). CRC Press.
Singh, P. (2022). Recommender Systems. In *Machine Learning with PySpark* (pp. 157-187). Apress, Berkeley, CA.
[3]Ajitsaria, Abhinav. Build a recommendation engine with collaborative filtering, 2022.
https://realpython.com/build-recommendation-engine-collaborative-filtering/#:~:text=Collaborative%20filtering%20is%20a%20technique,similar%20to%20a%20particular%20user