Big Data Analytics

Dr Sirintra Vaiwsri | Email: sirintra.v@itm.kmutnb.ac.th

Recommendation System

Recommendation System

(Chambers and Zaharia, 2018; Guller, 2015; Geeksforgeeks, 2023)

- A recommendation system is used to recommend a product or item to the user.
- The recommendation system learns from user's behaviours and preferences in the past to recommend a product or item.

Recommendation System

(Chambers and Zaharia, 2018; Guller, 2015; Geeksforgeeks, 2023)

- Explicit preference - express preferences through ratings.
- Implicit preference - through observation such as the number of clicks, number of likes, number of loves.
- Recommendation forms:
  - Content based
  - Collaborative filtering
  - Hybrid (combination of content based and collaborative filtering)

Content based (Chambers and Zaharia, 2018; Guller, 2015)

- It recommends a product based on its characteristics that match the previous product which the user is interested in.
- It uses explicit preference to determine product similarity for making recommendations.
- For example, Netflix recommends movies based on the genre that a user often watches, such as an action, a romantic, or a comedy movie.

Collaborative Filtering

(Chambers and Zaharia, 2018; Guller, 2015)

- It recommends a product to a user based on the preferences of users who have similar interests.
- It learns users with similar preferences and similar properties of products using data from rows in a tabular input dataset, where each row contains a user ID, product ID, and rating.
- The products that will be recommended to a user are the products with high rates from other users who have similar preferences.

Collaborative Filtering with Alternating Least Squares (Chambers and Zaharia, 2018)

- Alternating Least Squared (ALS) is a popular collaborative filtering recommendation system.
- ALS finds the k-dimensional feature vector for each user and product.
- ALS conducts a dot product of each user's feature vector with each item's feature vector, thus, it can approximate the user's rating for that product.

Collaborative Filtering with Alternating Least Squares (Chambers and Zaharia, 2018)

- It requires a tubular input dataset where each row contains a user ID, product ID, and rating.
- Each rating can be an explicit (numerical rating) or an implicit (such as the number of visits to a particular page).
- It uses input Dataframe to predict user's ratings for products which have not yet been rated.

Cold Start Problem (Chambers and Zaharia, 2018)

- It arises when new users or products have no rating history.
- It also occurs when using a random split because users or products in the testing set are not in the training set.
- Spark will assign NaN prediction.
- This can ruin the ability of your model evaluation.

Cold Start Problem (Chambers and Zaharia, 2018)

- Assigning NaN can be useful as you can design an overall system to fall back on default recommendations when a new user or new product is added to the system.
- Spark *coldStartStrategy* parameter is allowed to be used to drop any rows in the DataFrame of predictions that contain NaN values.
- Therefore, the evaluation can be conducted over non-NaN data in the Dataframe.

Root Mean Square Error (RMSE) (Sciencedirect, 2024)

- Root Mean Square Error (RMSE) is the measure of the differences between predicted and actual values.
- The smaller the RMSE is the better of predictions from the model.

Recommendation System Implementation

- Use the book_ratings.csv file as a dataset
- Assume we want to predict user's ratings for books.
- We evaluate the model using RMSE
- We show the Book ID, User ID, Rating, and Prediction for a User ID = 53
- We also show 5 recommended books for all users and 5 recommended users for all books.

Recommendation System Implementation

Import Libraries:

- SparkSession
- RegressionEvaluator
- ALS (from pyspark.ml.recommendation)

Recommendation System Implementation

To use ALS:

- Define maxIter by the number of maximum iteration
- Define userCol by assigning the column to be used for users
- Define itemCol by assigning the column to be used for items
- Define ratingCol by assigning the column to be used for ratings
- coldStartStrategy = “drop” can be used for dropping NaN values

Recommendation System Implementation

To evaluate:

- Define metricName as rmse
- Define labelCol by assigning the column to be used for the label
- Define predictionCol by assigning the column to be shown as a prediction
- Evaluate using the function evaluate()

Recommendation System Implementation

To show a user with a specific ID:

- The function filter() can be used.
  - For example, <your DF>.filter(<your DF[‘<column>’] == <value>)
- Show the result to check if the filter result is correct.

Recommendation System Implementation

To show the prediction of the user:

- Transform the model by using the filtered result as an input
- Show the result
  - The orderBy() function can be used for sorting the result based on the defined column

Recommendation System Implementation

To show the recommendation for all users:

- Use recommendForAllUsers(<number of recommendations>).show()
- To show full output, use truncate = False

Recommendation System Implementation

To show the recommendation for all items:

- Use recommendForAllItems(<number of recommendations>).show()
- To show full output, use truncate = False

Assignment (1 point)

- Please implement the recommendation system and show the results to get 1 point.
- The results include: RMSE and 3 dataframes

References

- Chambers, B., & Zaharia, M. (2018). *Spark: The definitive guide: Big data processing made simple*. " O'Reilly Media, Inc.".
- Guller, M. (2015). Big data analytics with spark.
- Geeksforgeeks. https://www.geeksforgeeks.org. Accessed: 2023-09-14.
- Sciencedirect. https://www.sciencedirect.com. Accessed: 2024-09-17.