Rithvik Aleshetty

RUSpark Report and Findings

- **RedditPhotoImpact**
  - For this program, first off, I created a schema based on the csv file with the columns: image_id, unixtime, title, subreddit, upvotes, downvotes, comments
  - Then connected the schema to the csv
  - After this, I created a dataframe using the groupby transformation. I grouped the values by the image_id and calculated the sum of all the upvotes, downvotes and comments using the. sum function
  - Next, I created a user defined function which adds up 3 parameters
  - Then I created a new dataframe with the result of applying this user defined function, and this gave the sum(impact score) for each image_id
  - **Based on the result from RedditPhotoImpact, the most impactful photo for the whole data set had an id of 1437 with an impact score of 192,896**
- **RedditHourImpact**
  - The implementation for this program was similar to the photoimpact but I had to convert to the hour offset from the unixtime format which was given as default
  - After creating that UDF and applying it to the dataframe, I was able to use the groupby transformation and follow the same steps to calculate the impact score as the previous program
  - **Based on the result from RedditHourImpact, the most impact hour was hour 20 or 8 PM with an impact score of 15057971**
- **NetflixMovieAverage**
  - This program started off just like the previous ones. I created a scheme based on the columns of the csv, the columns were movie_id, customer_id, rating, and date
  - I connected this to the csv dataframe
  - Then, a groupby transformation was done on movie_id as well as a avg calculation and a sort by movie_id
  - That line took care of the calculations and then I just printed the results
  - **Based on the result from NetflixMovieAverage, 4.72 was the highest average rating and multiple movies had this: the ids being 7230, 1496. The next highest had an average rating of 4.7 and the movie ID 7057 had this.**
- **NetflixGraphGenerate**
  - Once again, here I created the schema and attached it to the dataframe from the csv
  - Then, I used spark sql to write a query for the table. The query included selecting the movie_id and customer IDs where the movie_id was the same, rating was the same but customer_id were not the same (the first id had to be less than the second)
    - It was an implicit join with itself
  - This gave the customer_ids which I could then count to see how many pairs there were
  - I did this counting using a hashmap and then printed the final results
- **Conclusion: Overall thoughts and Reflection**
  - Overall, the most challenging part of the project was how to use spark and the transformations that come with it.

Rithvik Aleshetty

- Once I was able to get the first program, the rest excluding the graph one was simple and straightforward
- The graph part of the project was definitely the most challenging. I used a SQL query for this since using the spark transformation functions was difficult to figure out for a more complex query as this one