Brief Description of the logic you are using to make the Top Five Recommendations. This should specifically include your logic in “Coding Step 4”, and any other updates/changes you may have made to the given logic. This should not only include what decisions you made to create the composite score, but why you made them.

**Preprocessing:**

*While pre-processing the data, I have additionally applied Robust Scaling on 3 columns of Amazon books data: Sales Rank, Total Reviews and Degree Centrality for large range and large outliers. Reason behind being: unlike Standard scaler & Min-Max scaler, the centering and scaling statistics of the Robust scaler is based on percentiles and is therefore not influenced by a few numbers of very large marginal outliers. Hence, the resulting range of the transformed feature values is larger.* ***Most*** *of the transformed values will lie in a [-2, 3] range.*

*Post pre-processing the data, we have exported the amazon-meta data into amazon-books.csv and amazon-books-copurchase.edgelist file.*

**Recommendation Model Building:**

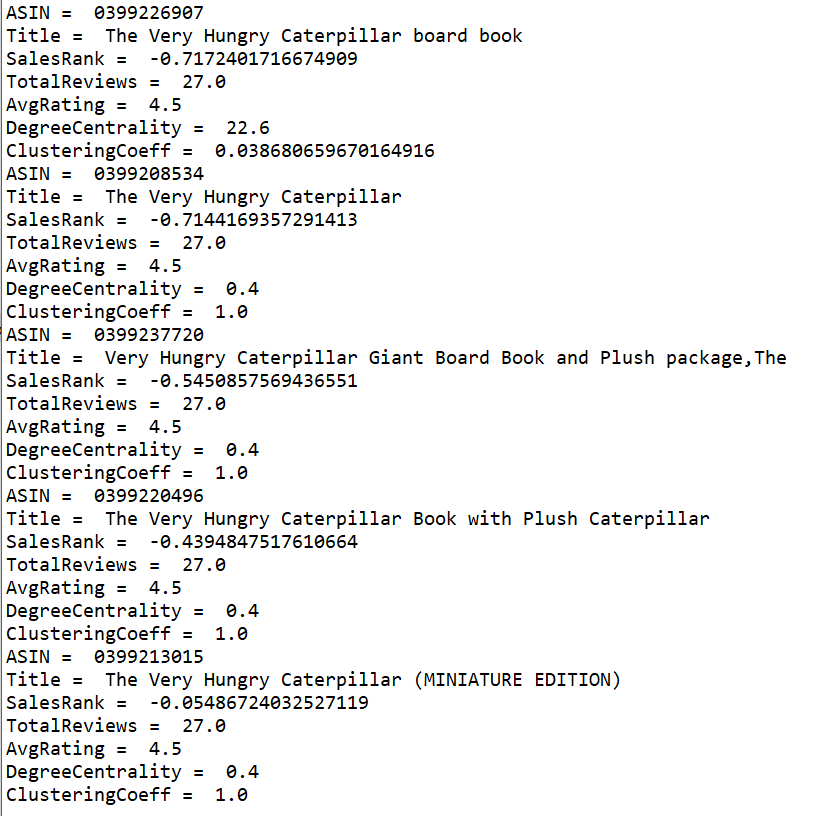
*We read the amzon-books.csv file and copurchase.edgelist file. Then, we used co-purchase.edgelist file to create a co-purchase graph structure with nodes, edges, similarity etc.*

1. *We assumed a book ASIN and then we got the list of books that had been co-purchased with our assumed book in the past. So, we get the depth 1 ego network of the purchased book from the copurchaseGraph to get the purchasedAsinEgoGraph*
2. *We now filter down to the most similar books. We use the island method on the purchasedAsinEgoGraph to only retain edges with threshold >= 0.5 and assign the resulting graph to purchasedAsinEgoTrimGraph.*
3. *We then trimmed the list of neighbors of the purchasedAsin in the purchasedAsinEgoTrimGraph and put them in purchasedAsinNeighbors.*
4. *I have chosen the following metrics: Average Rating, Total Reviews, Sales Rank, Degree Centrality and Clustering Coefficient for calculating the composite score to recommend books.*

* *To generate a common metric comprising of 5 metrics, I have made the ranges of all these metrics uniform:*
* *I have added 2.0 to Robust-scaled columns (Sales Rank, Total Reviews and Degree Centrality) to make their range positive and starting from 0*
* *I have multiplied the Clustering Coefficient by 5.0 to make its range [0,5]*
* *Average rating already has the range [0,5]*
* *For calculating the Composite score:*
* *I have added 4 metrics (Average Rating, Total Reviews, Degree Centrality and Clustering Coefficient), which tend to show better recommendations for higher values.*
* *Then subtracted the Sales Rank value from it, as it shows better recommendation for low values.*
* *The composite score with the highest value will show the most similarity as compared to the low scores.*

1. *The sort\_desc returns a tuple with ID and Score and we then make sort\_desc the key and reverse the tuple to get the maximum values for our composite scores. We then set the for range to 5, to get top 5 recommendations with respect to our assumed book and display them as below:*

**Output for the book ASIN: 0805047905, Brown Bear, Brown Bear, What Do You See?**

****