# Assignment 3 (Network Analysis)

This assignment is worth 20 points. The assignment is **individual** effort.

# Problem Definition

In this Assignment, we are going to use Amazon Product Co-purchase data to make Book Recommendations using Network Analysis. This assignment has three objectives:

* Apply Python concepts to read and manipulate data and get it ready for analysis
* Apply Network Analysis concepts to Build and Analyze Graphs
* Apply concepts in Text Processing, Network Analysis and Recommendation Systems to make a product recommendation

We will be using the [Amazon Meta-Data Set](http://snap.stanford.edu/data/amazon-meta.html) maintained on the [SNAP](http://snap.stanford.edu/index.html) site. This data set is comprised of product and review metdata on 548,552 different products. The data was collected in 2006 by crawling the Amazon website. You can view the data by double-clicking on the file **amazon-meta.txt** that’s been included in **NetworkAnalysis.zip**. The following information is available for each product in this dataset:

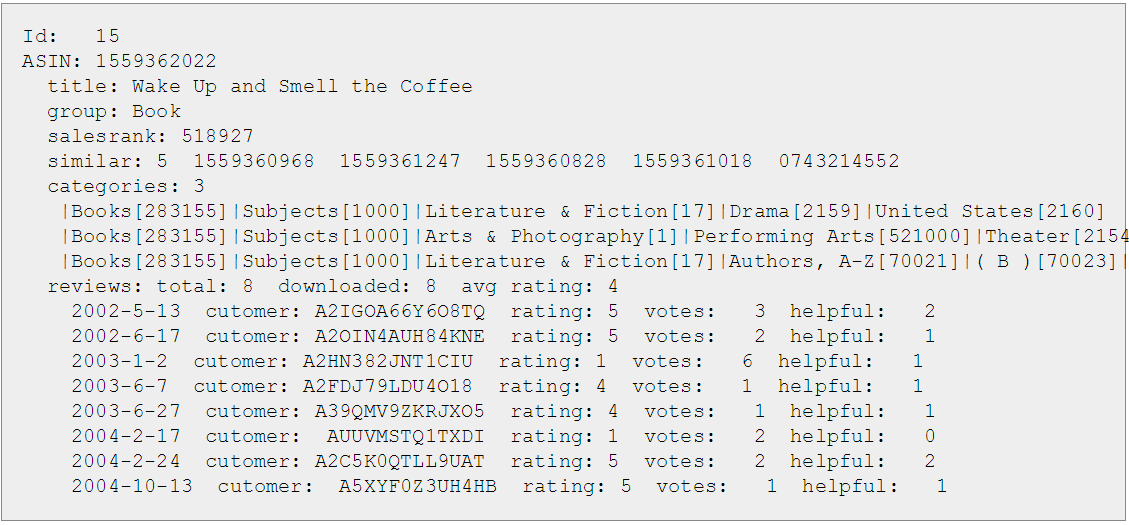
* **Id**: Product id (number 0, ..., 548551)
* **ASIN**: Amazon Standard Identification Number.

The Amazon Standard Identification Number (ASIN) is a 10-character alphanumeric unique identifier assigned by Amazon.com for product identification. You can lookup products by ASIN using following link: https://www.amazon.com/product-reviews/<ASIN>

* **title**: Name/title of the product
* **group**: Product group. Theproduct group can be Book, DVD, Video or Music.
* **salesrank**: Amazon Salesrank

The Amazon sales rank represents how a product is selling in comparison to other products in its primary category. The lower the rank, the better a product is selling.

* **similar**: ASINs of co-purchased products (people who buy X also buy Y)
* **categories**: Location in product category hierarchy to which the product belongs (separated by |, category id in [])
* **reviews**: Product review information: total number of reviews, average rating, as well as individual customer review information including time, user id, rating, total number of votes on the review, total number of helpfulness votes (how many people found the review to be helpful)



**The first step** we have to perform is read, preprocess, and format this data for further analysis. You have been provided with a Python script called **PreprocessAmazonBooks.py** that’s been included in **NetworkAnalysis.zip**. This script takes the “amazon-meta.txt” file as input, and performs the following steps:

* Parses the **amazon-meta.txt** file
* Preprocesses the metadata for all ASINs, and writes out the following fields into the **amazonProducts** Nested Dictionary (key = ASIN and value = MetaData Dictionary associated with ASIN):
  + **Id**: same as “Id” in amazon-meta.txt
  + **ASIN**: same as “ASIN” in amazon -meta.txt
  + **Title**: same as “title” in amazon-meta.txt
  + **Group**: product group that the product belongs to
  + **Categories**: a transformed version of “categories” in amazon-meta.txt. Essentially, all categories associated with the ASIN are concatenated, and are then subject to the following **Text Preprocessing** steps: lowercase, remove digit/punctuation, remove stop words, retain only unique words. The resulting list of words is then placed into “Categories”.
  + **Copurchased**: a transformed version of “similar” in amazon-meta.txt. Essentially, the copurchased ASINs in the “similar” field are filtered down to only those ASINs that have metadata associated with it. The resulting list of ASINs is then placed into “Copurchased”.
  + **SalesRank:** same as “salesrank” in amazon-meta.txt
  + **TotalReviews:** same as total number of reviews under “reviews” in amazon-meta.txt
  + **AvgRating:** same as average rating under “reviews” in amazon-meta.txt
* Filters amazonProducts Dictionary down to only Group=Book, and write filtered data to **amazonBooksND** Dictionary
* Uses the co-purchase data in amazonBooksND Dictionary to create the **copurchaseGraph** structure as follows:
  + **Nodes**: the ASINs are Nodes in the Graph
  + **Edges**: an Edge exists between two Nodes (ASINs) if the two ASINs were co-purchased
  + **Edge Weight (based on Category Similarity)**: since we are attempting to make book recommendations based on co-purchase information, it would be nice to have some measure of Similarity for each ASIN (Node) pair that was co-purchased (existence of Edge between the Nodes). We can then use the Similarity measure as the Edge Weight between the Node pair that was co-purchased. We can potentially create such a Similarity measure by using the “Categories” data, where the Similarity measure between any two ASINs that were co-purchased is calculated as follows:

**Similarity** = (Number of words that are common between Categories of connected Nodes)/

(Total Number of words in both Categories of connected Nodes)

The Similarity ranges from 0 (most dissimilar) to 1 (most similar).

* Adds the following graph-related measures for each ASIN to the **amazonBooksND** Dictionary:
  + **DegreeCentrality:** associated with each Node (ASIN)
  + **ClusteringCoeff:** associated with each Node (ASIN)
* Converts the amazonBooksND dictionary into the amazonBooks pandas dataframe, drops the “Copurchased” and “Group” columns, and then writes out the amazonBooks dataframe to the **amazon-books.csv** file
* Writes out the copurchaseGraph graph to the **amazon-books-copurchase.edgelist** file

Please read the **PreprocessAmazonBooks.py** script to ensure you are able to relate the code back to the preprocessing steps described above. Then, execute the script. It could take ~15 minutes to run. NOTE: while you are not required to, you are welcome to update/change this file as you see fit to complete this assignment.

**The next step** is to use this transformed data to make Book Recommendations. You have been provided with a Python script called **“Assignment3 - Framework.py”** that’s been included in **NetworkAnalysis.zip**. This script takes the “amazon-books.csv” and “amazon-books-copurchase.adjlist” files as input, and performs the following steps to get you started. This is the script you will need to update to complete Assignment 3.

* Read **amazon-books.csv** data into the **amazonBooks** pandas dataframe
* Read **amazon-books-copurchase.edgelist** into the **copurchaseGraph** structure
* We then assume a User has purchased a Book with ASIN=0805047905. The question then is, how do we make other Book Recommendations to this User, based on the Book copurchase data that we have? We could potentially take ALL books that were ever copurchased with this book, and recommend all of them. However, the Degree Centrality of Nodes in a Product Co-Purchase Network can typically be quite large. We should therefore come up with a better strategy.
* We examine the metadata associated with the Book that the User is looking to purchase (purchasedAsin =0805047905), including Title, SalesRank, TotalReviews, AvgRating, DegreeCentrality, and ClusteringCoefficient. We notice that this Book has a DegreeCentrality of 216 – which means 216 other Books were copurchased with this Book by other Customers. So yes, it would indeed make sense to come up with a better strategy of recommending copruchased Books. **This is the point where you need to start coding…**
* [**Coding Step 1**] Get the books that have been co-purchased with the purchasedAsin in the past. That is, get the depth-1 ego network of purchasedAsin from copurchaseGraph, and assign the resulting graph to purchasedAsinEgoGraph.
* [**Coding Step 2**] Filter down to the most similar books. That is, use the island method on purchasedAsinEgoGraph to only retain edges with threshold >= 0.5, and assign resulting graph to purchasedAsinEgoTrimGraph
* [**Coding Step 3**] Get the books that are still connected to the purchasedAsin by one hop (called the neighbors of the purchasedAsin) after the above clean-up, and assign the resulting list to purchasedAsinNeighbors.
* [**Coding Step 4**] Come up with a **composite measure** to make the Top Five book recommendations based on one or more of the following metrics associated with neighbors in purchasedAsinNeighbors: SalesRank, AvgRating, TotalReviews, DegreeCentrality, and ClusteringCoeff. Feel free to compute and include other metrics here. Think through the composite measure carefully. Your composite measure must be mathematically and logically sound. Think of how you might need to transform your features, how these might influence one another, and therefore how you might want to combine them to create your composite score. You’ll also want to look at the recommendations you are ending up with, and see if you can improve upon them with pre or post processing. **DO NOT code step 4 as a series of sorts! You must come up with a composite measure. Points will be deducted if you come up with a trivial method that hasn’t been thought through, and that doesn’t have a mathematical and/or theoretical basis to it.**
* [**Coding Step 5**] Print Top 5 recommendations (ASIN, and associated Title, Sales Rank, TotalReviews, AvgRating, DegreeCentrality, ClusteringCoeff) based on your composite measure.

Please read the **“Assignment3 - Framework.py”** script to ensure you are able to relate the code and comments back to the processing steps described above, as well as the coding requirements that you need to complete.

# Requirement for this Assignment

Here are the Requirements for this Assignment:

1. Complete the steps highlighted above:
   1. Download and unzip the NetworkAnalysis.zip file from BB
   2. Read, understand, and execute the PreprocessAmazonBooks.py script and ensure the “amazon-books.csv” and “amazon-books-copurchase.adjlist” files have been generated
   3. Read and understand “Assignment3 - Framework.py” script and ensure you are able to understand what five steps you need to code
2. Submit the files called out below.

# Submission for this Assignment

Submit the following for this Assignment:

1. **Brief Description** of the logic you are using to make the Top Five Recommendations. This should specifically include your logic in “Coding Step 4” above, 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. Submit the logic in a word document.

<Cohort X><First Name><Last Name>Assignment3.docx

*[Example: CohortBHinaAroraAssignment3.docx]*

1. **Updated script** that implements the five required coding steps called out in “Assignment3 - Framework.py”.

Once you have written up the script, save it as follows. Submit the script by uploading your python script. Note: upload the actual script – DO NOT attach a screenshot of the script!

<Cohort X><First Name><Last Name>Assignment3.py

*[Example: CohortBHinaAroraAssignment3.py]*

1. **If (and only if)** you updated the PreprocessAmazonBooks.py script, upload that as well. If you did not update this file in anyway, please **do not** upload it.

<Cohort X><First Name><Last Name>PreprocessAmazonBooks.py

*[Example: CohortBHinaArora*PreprocessAmazonBooks*.py]*

The submitted scripts will be run **as-is** for grading. **I will be plugging in different asins for purchasedAsin to see if your code is giving me Top Five recommendations for different asins.**

Points will be deducted if your composite score is too trivial, not well thought-out, not mathematically sound, or not logically sound.

Points will be also be deducted for scripts that:

* are difficult to read/follow
* don’t compile/run
* don’t have all the various pieces of code required
* have hard-code values instead of using variables
* have logical errors
* don’t result in the expected output