LINK ANALYSIS: Amazon Case

Muhammed Besim Sakaoglu 985861

Algorithms for Massive Datasets

September`23

“ I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. “

INTRODUCTION: Link Analysis and Conventional Marketing Strategies

In the realm of marketing, several interrelated notions play a vital role to shape consumers` purchasing behavior and optimizing business strategies. Tied selling, which was conventional marketing technique refers to the practice of bundling products or services together, often requiring customers to purchase one item in order to gain access to another. This strategy derived from joint demand, where the demand for one product is directly influenced by the demand for another, creating a relationship between them. When these products are complementary, they increase each other's value when used in merged, further reinforcing the concept of joint demand.[1] Thanks to tied selling strategies, companies can effectively leverage joint demand to increase sales of related products. They can also use cross price elasticity to fine-tune pricing strategies, ensuring that changes in prices have the desired impact on product demand. Additionally, offering discounts on tied-up products encourages customers to make bundled purchases, ultimately benefiting the company's revenue and boosting the customer's overall experience. Over the past decades, suppose a company sells a gaming console, along with additional controllers and a popular game as a bundle. In this scenario, the demand for the controllers and the game is directly linked to the demand for the gaming console. Additionally, they may offer a special promotion where customers receive a discount on the bundled accessories (controllers and game) when they purchase the gaming console. For instance, if a customer buys the console, they get a 30% discount on the controllers and a free game. This encourages customers to buy the bundle, as they sense a significant value in acquiring all the items together at a reduced price. To sum up, in Marketing or Economics jargon above examples discussed a lot and we have many terms to explain them with some fancy mathematical proofs. (i.e. joint demand, cross price elastic, complementary goods etc.)

In the digital age, businesses utilize the power of link analysis to uncover intricate patterns and connections among consumers and their preferences. This method involves mapping the relationships between customers, products, and interactions, providing insights into consumer behavior and market trends. Leveraging these insights, sophisticated recommendation systems emerge, which utilize algorithms to suggest products based on past behaviors and preferences. These systems increase the customer experience by leveraging the principles of joint demand and complementary products, effectively tying different offerings together for optimal results.[2]

[1] Source: Investopedia - Tied Selling Definition Source: Cleverism - Complementary Products

[2] Source: Towards Data Science - A Comprehensive Introduction to Different Types of Recommendation Systems

EDA and Dataset description

This project utilized the 'Amazon US Customer Reviews' dataset from Kaggle, obtained via the Kaggle API with username and key of mine. The dataset originally consisted of more than 30 CSV files, each representing a different product category and there were no null values. The analysis focused specifically on the "Automotive" category and most of the time I pointed out it as car products during the project implementation. However, the code can be adapted to include additional categories if needed and perfectly scalable as denoted on the task description.

Firstly, the 'marketplace' feature serves as a categorical identifier, signifying the specific marketplace within the Amazon platform. This distinction is crucial for tracking consumer activity across different sections of the digital marketplace.

The 'customer_id' feature offers a unique alphanumeric code for each customer. This identifier acts as a cornerstone for individual-level analysis, allowing for personalized insights into consumer behavior and preferences.

Complementing the 'customer_id', the 'review_id' attribute provides a distinct alphanumeric label for each review. This facilitates the precise tracking and management of individual reviews, which is particularly valuable for sentiment analysis and customer feedback assessment.

The 'product_id' feature, similar to 'customer_id', assigns a unique alphanumeric code to each product within the Amazon ecosystem. This enables a granular examination of individual products and their performance.

'Product_parent' is another key attribute, representing a distinct identifier for product families or groups. It aids in categorizing products into broader families, providing a hierarchical view of the product catalog.

The 'product_title' feature offers a clear, descriptive title for each product. This textual information is instrumental in identifying and understanding the products in question.

Categorization of products is facilitated by the 'product_category' attribute, which classifies items into specific product groups. This categorical label is essential for organizing and analyzing products within their respective categories.

Star ratings, denoted by 'star_rating', are assigned as integer values, reflecting customer satisfaction levels. This numerical feedback is a pivotal indicator of a product's performance and overall customer sentiment.

'Helpful_votes' and 'total_votes' are integral features, quantifying the number of helpful and total votes received on a review, respectively. These metrics offer insights into the perceived usefulness and engagement levels of individual reviews.

The 'vine' label, a binary classification attribute, denotes whether a review is part of the Vine program (0 for yes, 1 for no). This distinction is significant for understanding the impact of incentivized reviews on product feedback.

Similarly, the 'verified_purchase' label (0 for yes, 1 for no) indicates whether a purchase is verified. This distinction aids in differentiating between reviews from verified purchasers and those from unverified sources.

'Review_headline' and 'review_body' are textual features, presenting the headline and body of each review, respectively. These components contain the qualitative feedback and opinions shared by customers.

Lastly, 'review_date' records the date of each review, providing a temporal dimension to customer feedback. This temporal information is essential for trend analysis and understanding how product perceptions evolve over time.

Each dataset file contains 15 columns, To illustrate[3]

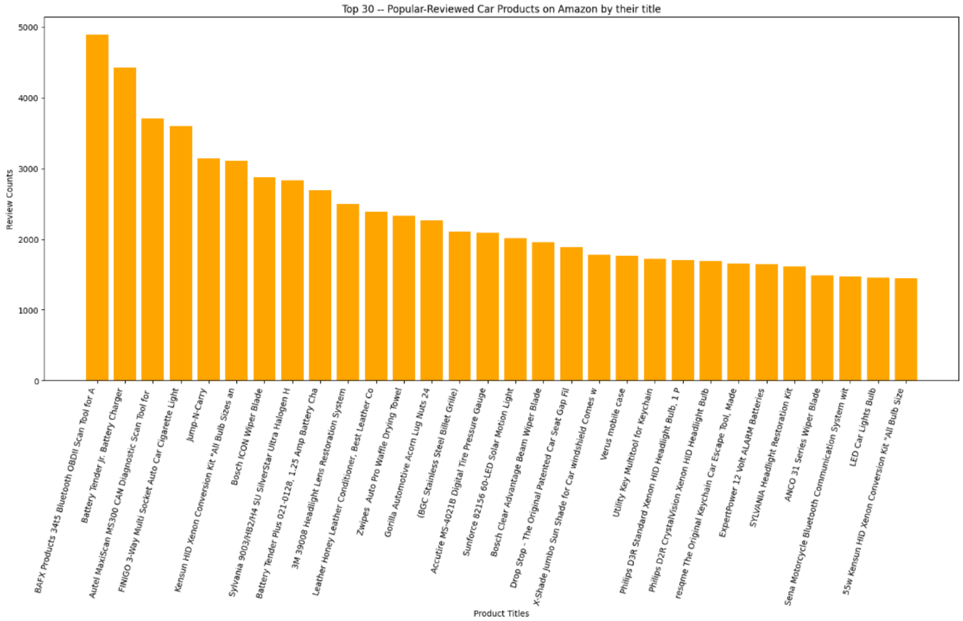
```
{
   "customer_id": "16825098",
   "helpful_votes": 0,
   "marketplace": "US",
   "product_category": "Automotive",
   "product_id": "B000E4PCGE",
   "product_parent": "694793259",
   "product_title": "00-03 NISSAN SENTRA MIRROR RH (PASSENGER SIDE), Power, Non-Heated (2000 00 2001 01 2002 02 2003 03) NS35ER 963015M000",
   "review_body": "Product was as described, new and a great look. Only bad thing is that one of the screws was stripped so I couldn't tighten all three.",
   "review_date": "2015-08-31",
   "review_headline": "new and a great look. Only bad thing is that one of ...",
   "review_id": "R2RUIDUMDKG7P",
   "star_rating": 3,
   "total_votes": 0,
   "verified_purchase": 1,
   "vine": 0
}
```

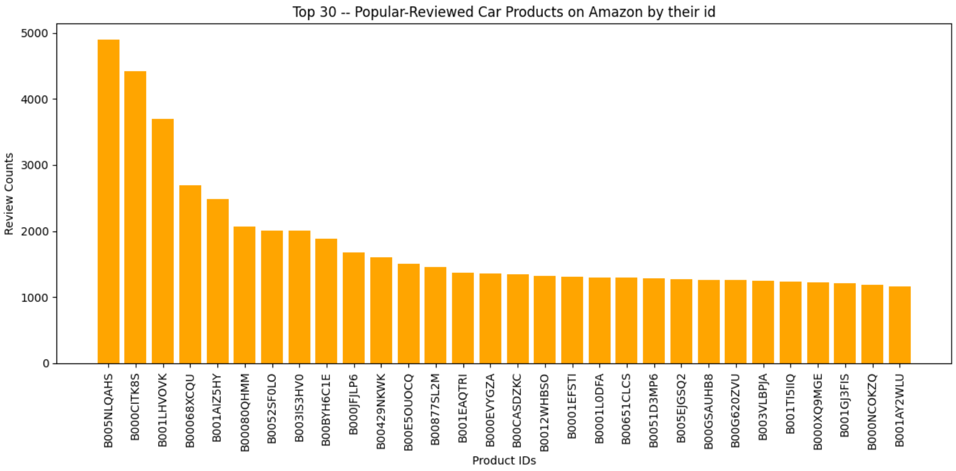
but the EDA part concentrated on specific columns including customer ID, review ID, product ID, product title, and star rating. These columns were chosen for their relevance to the analysis. Later on, Spark data frame has been converted to RDD by keeping only product_id and customer_id columns.

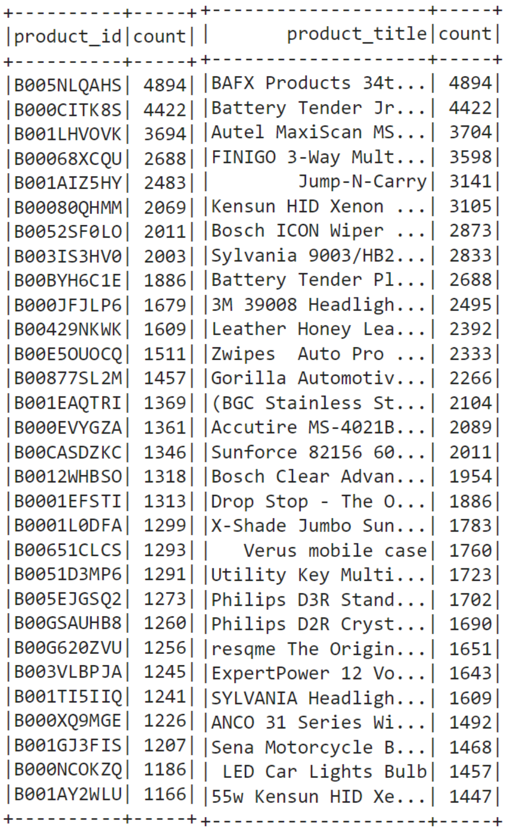
[3] https://huggingface.co/datasets/amazon_us_reviews#automotive_v1_00

Two products will be linked if they have been reviewed at least by the same customer and products considered as nodes and edge between them is a shared customer. To ensure that, customers who did not review more than one product are eliminated.

I checked mostly reviewed top 30 products by their title and their IDs :



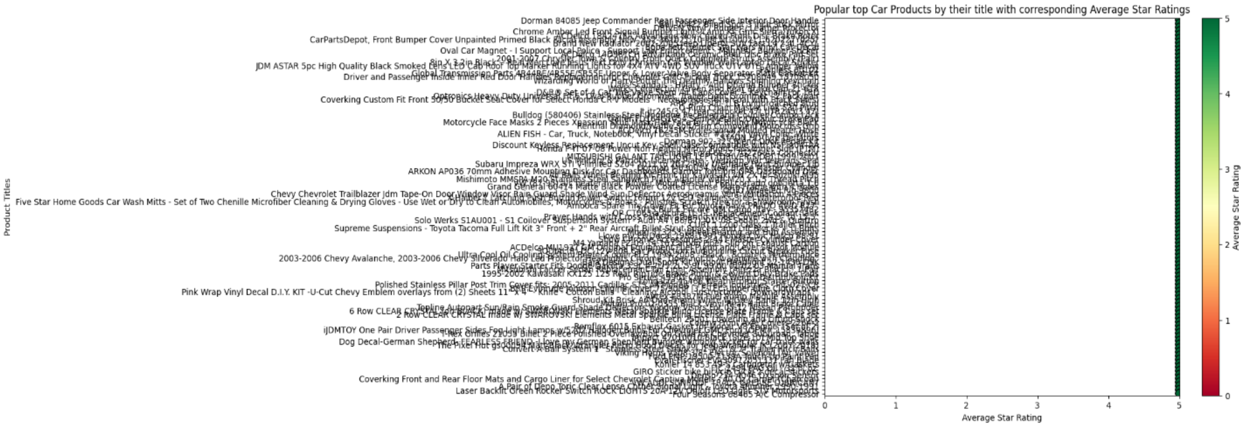




```
+----------+-----+ +--------------------+-----+
|product_id|count| |       product_title|count|
+----------+-----+ +--------------------+-----+
|B005NLQAHS| 4894| |BAFX Products 34t...| 4894|
|B000CITK8S| 4422| |Battery Tender Jr...| 4422|
|B001LHVOVK| 3694| |Autel MaxiScan MS...| 3704|
|B00068XCQU| 2688| |FINIGO 3-Way Mult...| 3598|
|B001AIZ5HY| 2483| |         Jump-N-Carry| 3141|
|B00080QHMM| 2069| |Kensun HID Xenon ...| 3105|
|B0052SF0LO| 2011| |Bosch ICON Wiper ...| 2873|
|B003IS3HV0| 2003| |Sylvania 9003/HB2...| 2833|
|B00BYH6C1E| 1886| |Battery Tender Pl...| 2688|
|B000JFJLP6| 1679| |3M 39008 Headligh...| 2495|
|B00429NKWK| 1609| |Leather Honey Lea...| 2392|
|B00E5OUOCQ| 1511| |Zwipes  Auto Pro ...| 2333|
|B00877SL2M| 1457| |Gorilla Automotiv...| 2266|
|B001EAQTRI| 1369| |(BGC Stainless St...| 2104|
|B000EVYGZA| 1361| |Accutire MS-4021B...| 2089|
|B00CASDZKC| 1346| |Sunforce 82156 60...| 2011|
|B0012WHBSO| 1318| |Bosch Clear Advan...| 1954|
|B0001EFSTI| 1313| |Drop Stop - The O...| 1886|
|B0001L0DFA| 1299| |X-Shade Jumbo Sun...| 1783|
|B00651CLCS| 1293| |    Verus mobile case| 1760|
|B0051D3MP6| 1291| |Utility Key Multi...| 1723|
|B005EJGSQ2| 1273| |Philips D3R Stand...| 1702|
|B00GSAUHB8| 1260| |Philips D2R Cryst...| 1690|
|B00G620ZVU| 1256| |resqme The Origin...| 1651|
|B003VLBPJA| 1245| |ExpertPower 12 Vo...| 1643|
|B001TI5IIQ| 1241| |SYLVANIA Headligh...| 1609|
|B000XQ9MGE| 1226| |ANCO 31 Series Wi...| 1492|
|B001GJ3FIS| 1207| |Sena Motorcycle B...| 1468|
|B000NCOKZQ| 1186| | LED Car Lights Bulb| 1457|
|B001AY2WLU| 1166| |55w Kensun HID Xe...| 1447|
+----------+-----+ +--------------------+-----+
```

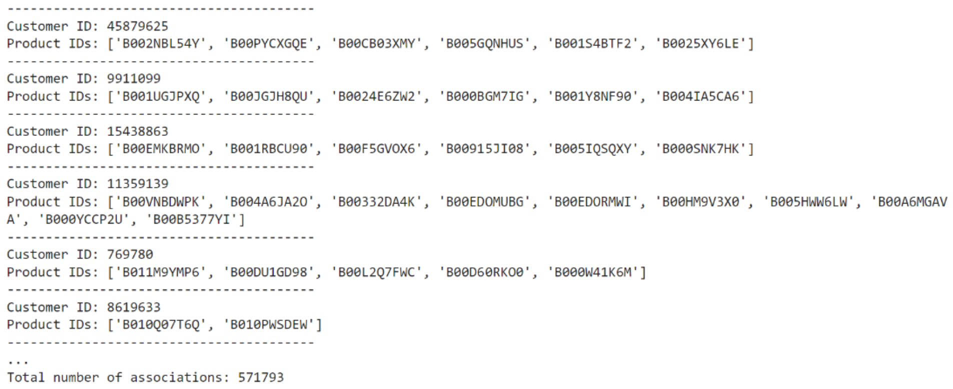
We can spot the difference here if we focus on product title `Bosh ICON Wiper Blade`, it ranked on 7th by title. However, if we had keep our analyses on by product_category let`s say Wiper Blade category for car products expectedly would get higher reviews count. Also, by product_id on the 7th place we observe only 2011 reviews count and we can think that 762 reviews difference comes from other Wiper Blade model from Bosh with different product_id.

Also, I thought that checking average stars for popular car products would be insightful for Explanatory Data Analysis (EDA) part. For further inspection, I may expand it by performing additional analysis by visualizing the distribution of star ratings for these products that have highest PageRank scores or identifying patterns between high-rated products and certain review attributes.



Building blocks of link analysis

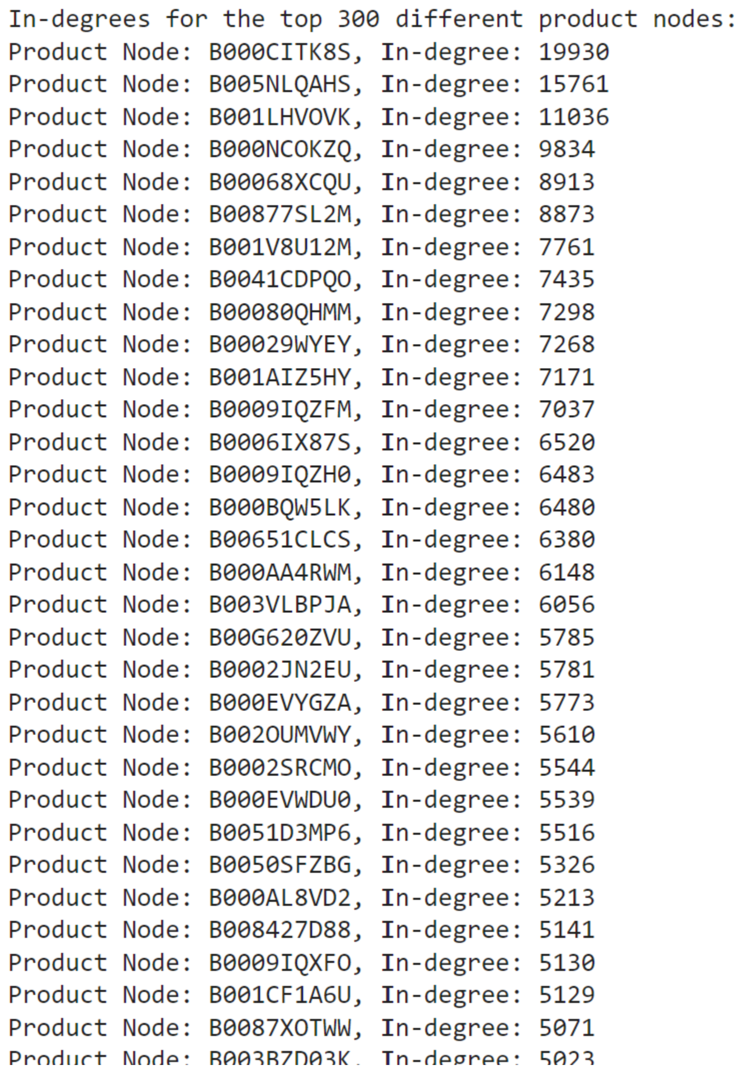
I constructed RDD that contains elements where each customer ID is associated with a list of product IDs they have reviewed. This RDD is the basis for building customer connections based on their shared product reviews, which is a key part of link analysis. Meaningly, key-value pairs are created, where the customer ID served as the key and the values were lists of products reviewed by each customer. Moreover, filtered out customers who have reviewed fewer than 2 products as I mentioned before. To illustrate, I performed a sanity check on 50 customer and output of last 6 customers was like this:



```
----------------------------------------
Customer ID: 45879625
Product IDs: ['B002NBL54Y', 'B00PYCXGQE', 'B00CB03XMY', 'B005GQNHUS', 'B001S4BTF2', 'B0025XY6LE']
----------------------------------------
Customer ID: 9911099
Product IDs: ['B001UGJPXQ', 'B00JGJH8QU', 'B0024E6ZW2', 'B000BGM7IG', 'B001Y8NF90', 'B004IA5CA6']
----------------------------------------
Customer ID: 15438863
Product IDs: ['B00EMKBRMO', 'B001RBCU90', 'B00F5GVOX6', 'B00915JI08', 'B005IQSQXY', 'B000SNK7HK']
----------------------------------------
Customer ID: 11359139
Product IDs: ['B00VNBDWPK', 'B004A6JA2O', 'B00332DA4K', 'B00EDOMUBG', 'B00EDORMWI', 'B00HM9V3X0', 'B005HWW6LW', 'B00A6MGAV
A', 'B000YCCP2U', 'B00B5377YI']
----------------------------------------
Customer ID: 769780
Product IDs: ['B011M9YMP6', 'B00DU1GD98', 'B00L2Q7FWC', 'B00D60RKO0', 'B000W41K6M']
----------------------------------------
Customer ID: 8619633
Product IDs: ['B010Q07T6Q', 'B010PWSDEW']
----------------------------------------
...
Total number of associations: 571793
```

Figure: customer ID is associated with a list of product IDs they have reviewed.

Then I also checked incoming-links for each product_id and first few output is:



```
In-degrees for the top 300 different product nodes:
Product Node: B000CITK8S, In-degree: 19930
Product Node: B005NLQAHS, In-degree: 15761
Product Node: B001LHVOVK, In-degree: 11036
Product Node: B000NCOKZQ, In-degree: 9834
Product Node: B00068XCQU, In-degree: 8913
Product Node: B00877SL2M, In-degree: 8873
Product Node: B001V8U12M, In-degree: 7761
Product Node: B0041CDPQO, In-degree: 7435
Product Node: B00080QHMM, In-degree: 7298
Product Node: B00029WYEY, In-degree: 7268
Product Node: B001AIZ5HY, In-degree: 7171
Product Node: B0009IQZFM, In-degree: 7037
Product Node: B0006IX87S, In-degree: 6520
Product Node: B0009IQZH0, In-degree: 6483
Product Node: B000BQW5LK, In-degree: 6480
Product Node: B00651CLCS, In-degree: 6380
Product Node: B000AA4RWM, In-degree: 6148
Product Node: B003VLBPJA, In-degree: 6056
Product Node: B00G620ZVU, In-degree: 5785
Product Node: B0002JN2EU, In-degree: 5781
Product Node: B000EVYGZA, In-degree: 5773
Product Node: B002OUMVWY, In-degree: 5610
Product Node: B0002SRCMO, In-degree: 5544
Product Node: B000EVWDU0, In-degree: 5539
Product Node: B0051D3MP6, In-degree: 5516
Product Node: B0050SFZBG, In-degree: 5326
Product Node: B000AL8VD2, In-degree: 5213
Product Node: B008427D88, In-degree: 5141
Product Node: B0009IQXFO, In-degree: 5130
Product Node: B001CF1A6U, In-degree: 5129
Product Node: B0087XOTWW, In-degree: 5071
Product Node: B003BZD03K, In-degree: 5023
```

I noticed that Product Node: B000NCOKZQ, In-degree: 9834 has now on top 4 in terms of in-link but if we check product reviews counts by product_id solely it was barely ranked 29th. if a product has a low review count but a high incoming link count, it may mean that the product is popular among other websites but not among customers. The PageRank score of a product page may depend on both factors, as well as the PageRank scores of the pages linking to it. Therefore, it is possible that a product with a lower review count may have a higher PageRank score than a product with a higher review count, depending on the link structure of the web. I think that it was a good starting point before diving into PageRank and Directed Graph Analysis.

The PageRank Algorithm

PageRank, a groundbreaking algorithm developed by Google's founders, and its name comes from one of the founders who is Larry Page, revolutionized web search by providing a dynamic measure of web page importance. Back in the days, before Google implemented the PageRank algorithm it was easy to rank on top of the web browsers. There were many search engines before Google. Largely, they worked by crawling the Web and listing the terms (words or other strings of characters other than white space) found in each page. Web-spammers saw the opportunity to fool search engines into leading people to their page. PageRank was used to simulate where Web surfers, starting at a random page, would tend to congregate if they followed randomly chosen outgoing links from the page at which they were currently located, and this process was allowed to iterate many times. Pages that would have many surfers were considered more "important" than pages that would rarely be visited.[4]Google prefers important pages to unimportant pages when deciding which pages to show first in response to a search query. The content of a page was judged not only by the terms appearing on that page, but by the terms used in or near the links to that page. Note that it was easy for spammers to add false terms to a page they control like keywords (term spam) from the top pages, they cannot as easily get false terms added to the pages that link to their own page, if they do not control those pages. Moreover, its applications extend to various network structures, including social networks and, notably, our Amazon customer reviews dataset.

I used Teleportation algorithm which is initialized by considering an imaginary surfer who is randomly clicking on links will eventually stop clicking. taking into account β instead using our previous update line $\underline{\mathrm{V}}_{t+1}=\mathbf{M}.\underline{\mathrm{V}}_t$

Since I wanted to avoid `Spider Trap` and `Deadends`. However, it will not completely eliminate the problem but it will somehow relieve the problem. Below formula and its code implications used for the PageRank calculations, for this particular project.

$$\underline{\mathbf{v}}_{t+1} = \beta\, \mathbf{M}\, \underline{\mathbf{v}}_{t} + \frac{(1-\beta)}{n}\,(\underline{1})$$

The matrix **M** is known as the transition matrix, which encapsulates the likelihoods of moving from one node to another using the available outgoing links. where β ∈ (0,1) is a chosen constant, usually in the range 0.8 to 0.9 and Google uses 0.85, `$\underline{1}$` is a vector of all 1's with the appropriate number of components, and `n` is the number of nodes in the Web graph. The triggering vector which is this part on the equation $\frac{1}{n}\,.\,(\underline{1})$ it represents random surfers' probability distribution of being on the any

[4] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Mining of Massive Datasets. Cambridge University Press.

webpages(n) as during the class discussion it was denoted by the $\underline{\mathbf{v}}_{\mathbf{0}}$. The term β **M** $\underline{\mathbf{v}}_t$ represents the case where, with probability β, the random surfer decides to follow an outgoing link from their present page. The term $\frac{(1-\beta)}{n}$ $(\underline{1})$ is a vector each of whose components has value $\frac{(1-\beta)}{n}$ and represents the introduction, with probability 1 − β, of a new random surfer at a random page.

Why do we use Taxation name while we called Teleportation?

Another possible interpretation would be each time I have to pay a tax which is some portion of households who are sitting in a node who do not follow the random surfing process, but they are automatically reassigned democratically. In other saying, taxes would be distributed uniformly over all the population. It ensures that even pages with no outgoing links or pages in isolated parts of the web have a chance to receive some "tax revenue."

APPLICATION OF ABOVE FORMULATIONS AND NOTATIONS

I tried to mimic what we discussed during the lectures, and I could manage to implement as described below:

1. **Generate Combinations of Product Pairs :**
    - The **my_combination** function takes a row from the data, which contains a customer and the list of products they reviewed. I choosed that name so that it does not collide with itertools.combinations() function from the itertools module.
    - It generates combinations of product pairs for each customer. These combinations represent edges in the graph. For example, if a customer reviewed products x, y, and z, the function generates pairs (x, y), (x, z), and (y, z).
    - These edges indicate that customers who reviewed the same products are connected in the graph.
2. **Create the Graph (Edges):**
    - The code creates a directed graph where nodes represent products, and edges represent shared customer reviews. If two products have been reviewed by the same customer, there is a directed edge between them.
3. **Calculate Out-Degrees:**
    - The out-degree of a node (product) is the number of outgoing edges (the number of other products it has been reviewed with by the same customer).
    - The code calculates the out-degree for each product by counting the number of outgoing edges.
4. **Initialize PageRank Parameters:**
    - PageRank is an iterative algorithm, and it requires several parameters:

- **alpha**: A damping factor (usually set to 0.85) that represents the probability a user will continue clicking on links.
- **iterations**: The maximum number of iterations for the PageRank algorithm.
- **threshold**: A convergence threshold that determines when the algorithm has converged (when the change in PageRank scores falls below this threshold).

5. **Create Transition Matrix (Probability Matrix):**
    - PageRank uses a transition matrix that defines the probabilities of transitioning from one product (node) to another.
    - The code creates this matrix based on the edges and out-degrees of the products.
6. **Calculate the Google Matrix (GM):**
    - The Google Matrix is a modified version of the transition matrix that incorporates the damping factor (β) and accounts for the probability of teleporting to any page.
    - The code computes the Google Matrix.
7. **Initialize PageRank Vector:**
    - The PageRank vector represents the importance scores of each product (node). Initially, all products are assumed to have equal importance, so the vector is initialized with uniform values.
8. **Perform PageRank Iterations:**
    - The PageRank algorithm iteratively updates the PR scores based on the graph structure.
    - In each iteration, it calculates a new PageRank vector by applying the Google Matrix to the previous PageRank vector.
    - The code performs a specified number of iterations or until convergence (when the change in PageRank scores is below the threshold, detailed explanation on code line #comments).
9. **Check for Convergence:**
    - After each iteration, the code checks if the PageRank scores have converged (the change is below the threshold). If convergence is reached, the algorithm stops early.
10. **Display PageRank Scores:**
    - Finally, the code sorts the products based on their PageRank scores in descending order and displays the top 25 products.
    - These products are considered the most influential or important in the network of customer reviews.

Visualizations of PR and Graph results:

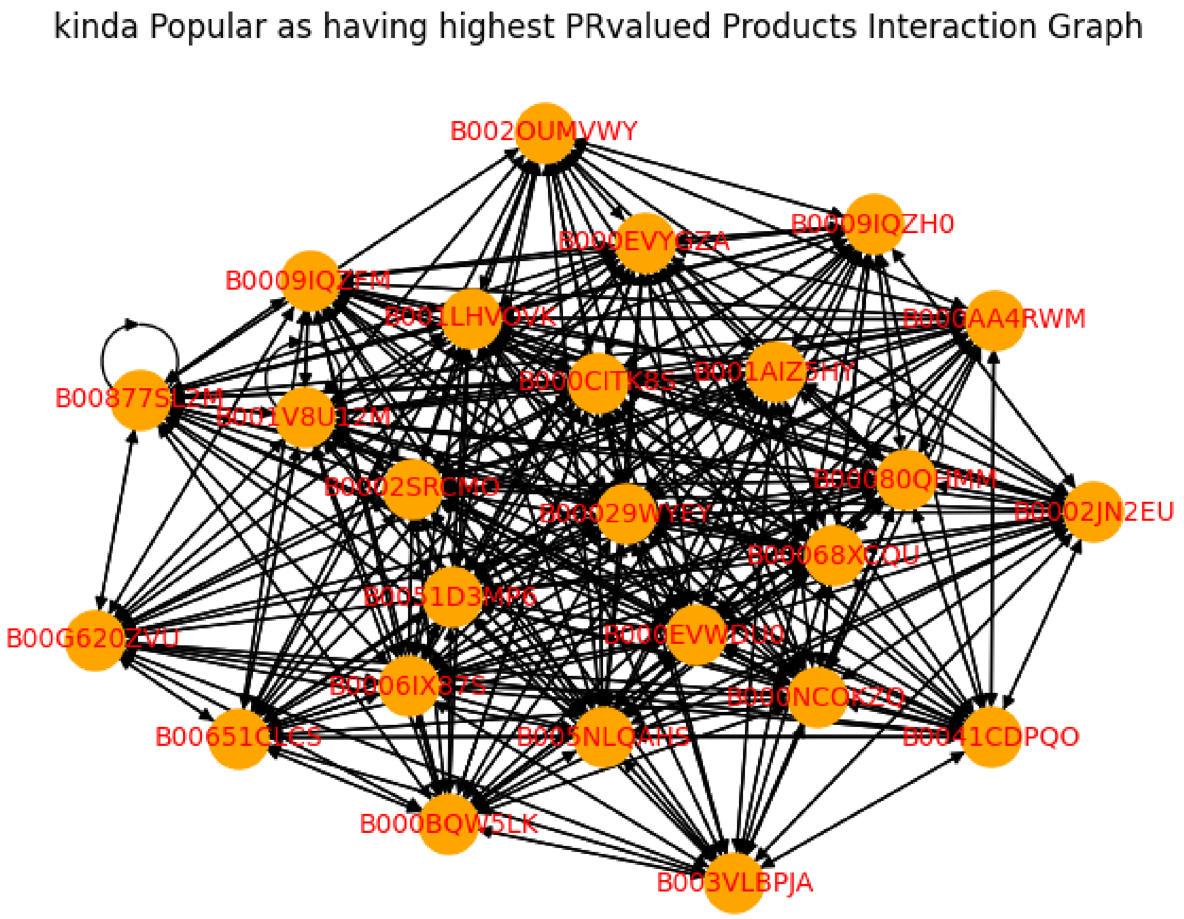


Figure: Having Highest PageRank Score of 25 Car Products represented in Directed Graph

No Deadends and Spider Trap have been found, thanks to Teleportation\Taxation algorithm.

Product_id `B00877SL2M` can be interpreted as self-loop, but it is not a Spider Trap.

For instance, if a customer purchased a car product (e.g., B00877SL2M) and then later bought it again and reviewed it, you would have a self-loop in the graph representing this data. This is relatively rare in practice because customers typically review a product only once, but it can occur for various reasons such as updating a previous review, purchasing a replacement for the same product, or mistakenly reviewing a product multiple time.

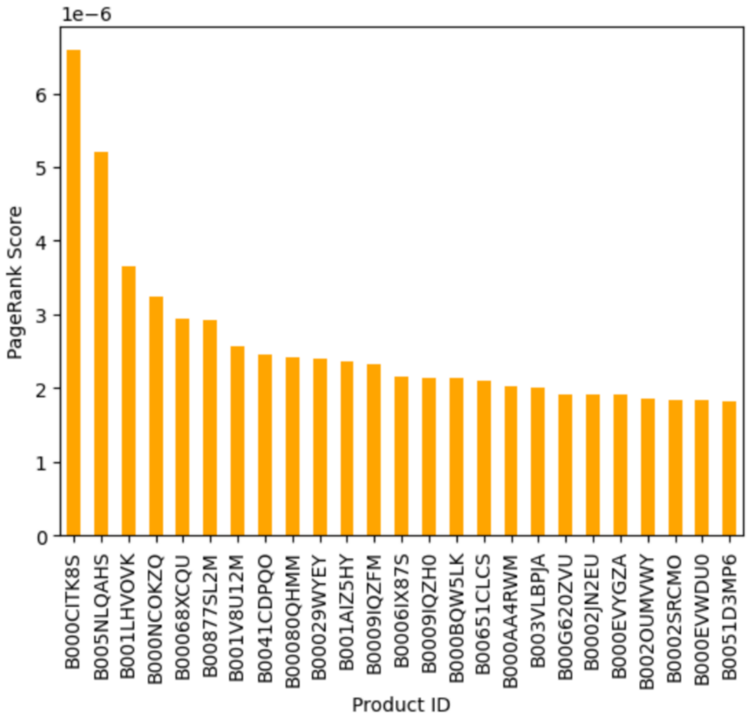
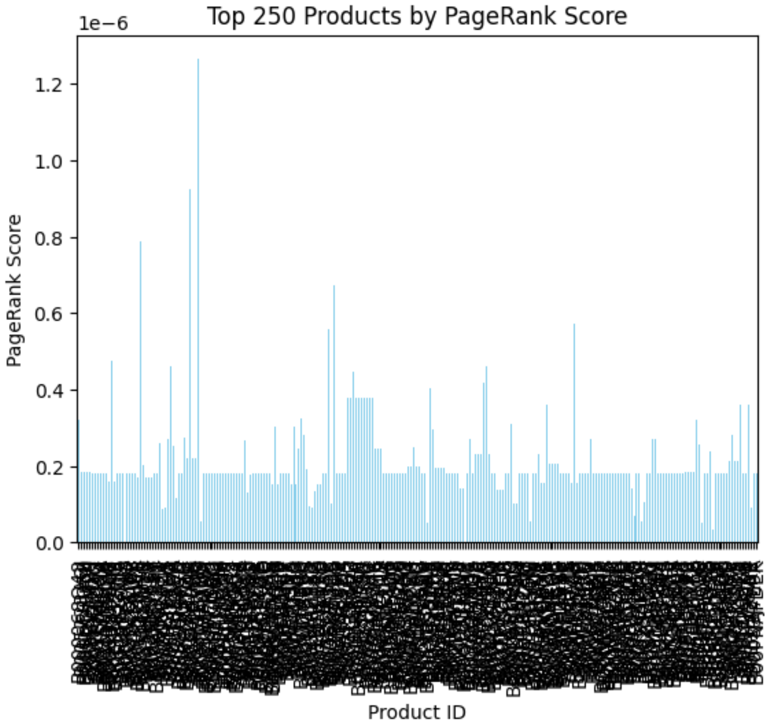


Figure: Having Highest PageRank Score of 25 Car Products represented in bar plot

Figure: Experimental part with different hyperparameters on Digital_Ebook_Purchase_v1_00


Top 250 Products by PageRank Score
1e−6
PageRank Score
Product ID