UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE POLITICHE, ECONOMICHE E SOCIALI

Master degree in Data Science and Economics

Project for Algorithm of Massive Data

Title: Link Analysis on Amazon Products Using PageRank

Prepared by: Samuel Tsegai

Submitted To: Prof. Malchiodi

Academic Year: 2023/2024

Date: NOv 06, 2023

Declaration:

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.

Abstract

This report outlines the development of a PageRank algorithm for link analysis using Apache Spark. The algorithm is applied to a dataset of customer reviews and product information from an e-commerce platform (Amazon products). The goal is to calculate product rankings based on weighted links between products, facilitating personalized product recommendations.

Contents

1 Introduction

The landscape of e-commerce is dotted with countless products. However, some products naturally take precedence due to various factors, like quality, brand reputation, or sheer popularity. This project aims to extract such product significance by analyzing links formed based on user reviews.

1.1 Dataset

We selected three Amazon product review datasets such us ,Books ,Electronics and Music. These datasets contain information such as customer IDs, product IDs, ratings, and verification of the purchase. All three categories often intersect in user purchase behavior. For instance, a customer buying a book on music or electronics might also venture into purchasing relevant products in the other two categories. This interconnected nature enriches the analysis base.By leveraging multiple datasets, we can construct a more robust and comprehensive link analysis, identifying product connections across categories.

Data Organization

Datasets were loaded using PySpark, sampled to manageable sizes, and then unified into a single dataset for processing.

1.2 Setup and Data Loading

Environment Setup

The initial steps include setting up the environment by installing Spark, configuring environment variables, and initializing a SparkSession. Additionally, essential Python libraries.

Dataset Preprocessing

The dataset, sourced from Kaggle, is downloaded and extracted. Desired datasets are selected and loaded into Spark DataFrames. To optimize computation time, a 1% sample of each dataset is created and unioned into a single DataFrame. Relevant columns are selected and duplicate records are removed.

2 Exploratory data analysis (EDA)

2.1 Data Statistics

Basic statistics are computed, including the approximate number of unique products, unique customers, and the total number of reviews. Visualizations of star rating distribution, product category distribution, and the proportion of verified purchases are provided.

- **number of unique products:13879**
- **number of unique customers: 5011**
- **Total number of reviews: 15183**

2.2 Data Visualization

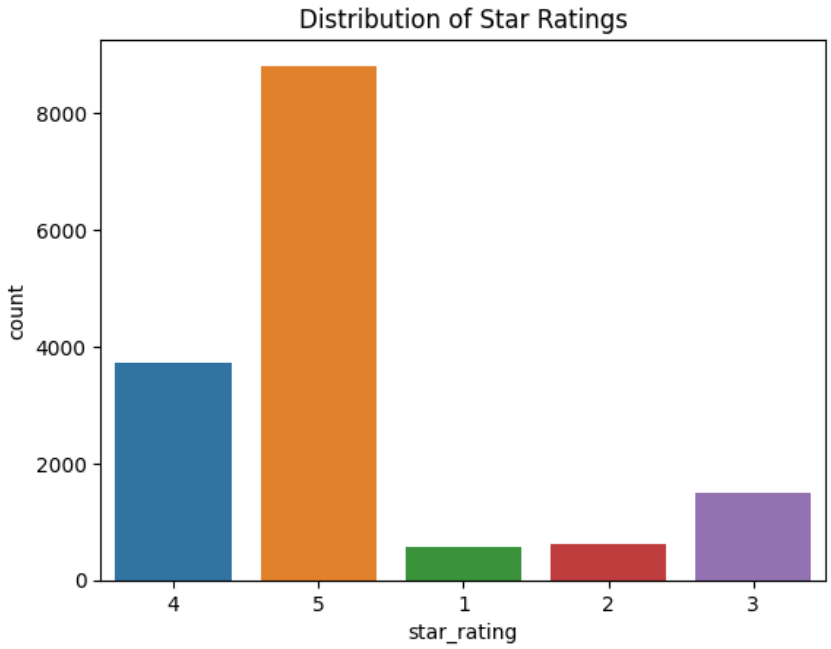

Figure 1: Distribution of Star ratings

The distribution of star ratings with a large number of 5-star ratings followed by 4, 3, 2, and 1 suggests that the majority of customers have had positive experiences with the products, but there are also some who have expressed more mixed or negative opinions.

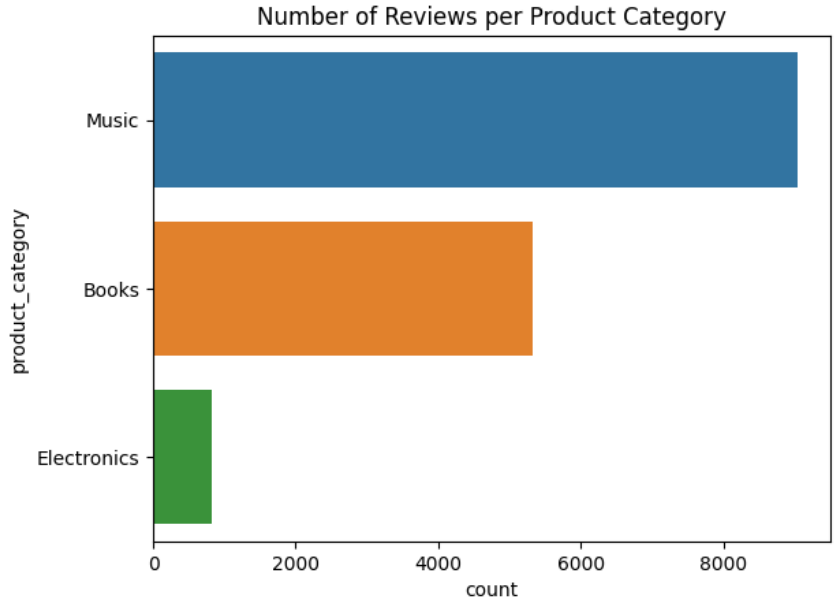

Figure 2: Number of Reviews per Product Categories

The distribution of Number of Reviews per Product Categories with a large number of reviews on Music followed by Books and Electronics.

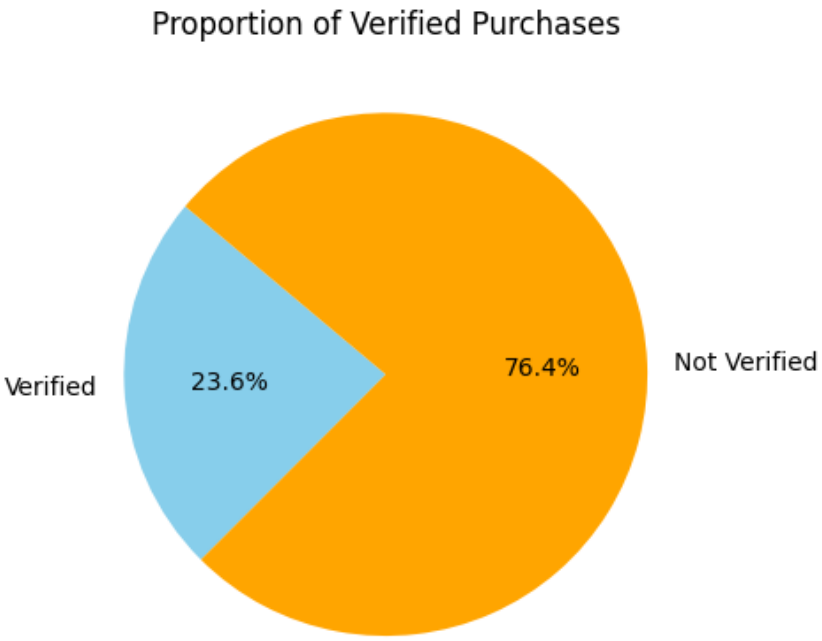

Figure 3: Proportion of Verified Purchases

Analyzing product reviews and ratings, knowing the proportion of verified and not verified reviews is essential. It helps in understanding the credibility of the feedback provided by customers. Verified reviews are often given more weight in assessments of product quality and customer satisfaction because they are tied to confirmed purchases, which implies a direct experience with the product the distribution indicates that a significant portion of the reviews (76.3%) are not associated with verified purchases, while a smaller portion (23.7%) are from customers who have verified their product purchases.

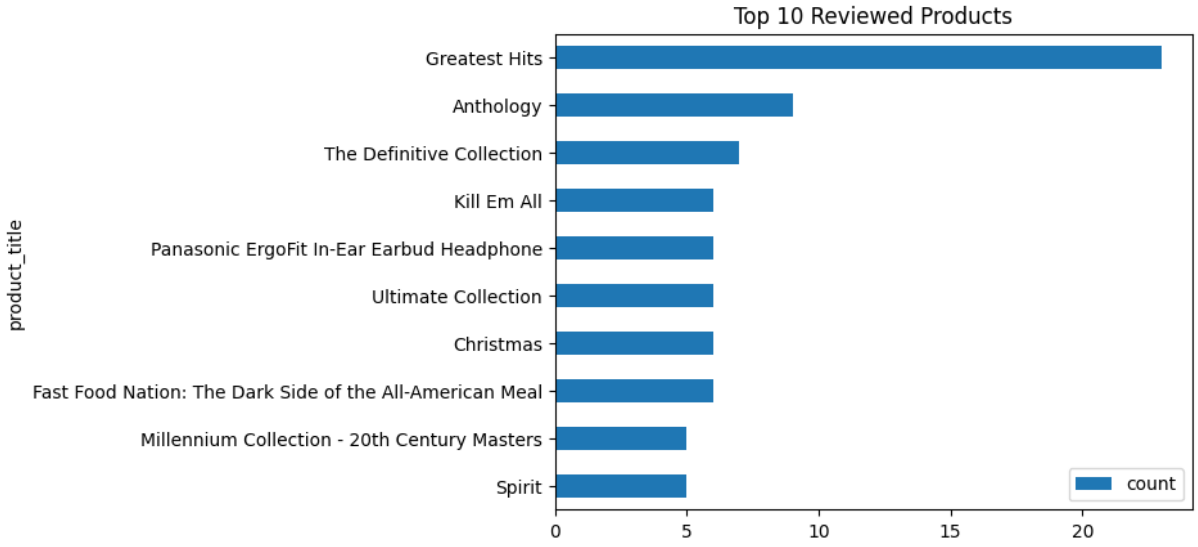

Figure 4: Top 10 products

3 Page Rank Algorithm Implementation

3.1 Link Analysis and Weight Calculation

The key component of the PageRank algorithm is the creation of weighted links between products. Links are generated by combining product pairs, incorporating various factors such as customer ratings, verification status, and product popularity. These factors are used to calculate the weight of each link.

- **How the weight is Calculate** To determine the weight of the connections between products, we consider several factors:
- **Ratings** We calculate the average rating given by customers to both products in a pair.
- **Verification** If reviews for both products are verified, we boost the connection's weight by a small value (0.5), enhancing the credibility of the connection.
- **Verified Purchase Boost** If both products were purchased with a 'verified purchase,' the weight is further increased (boosted) by 0.5.

$$weight = weight + 0.5 \quad (Only_i f_b oth products'_p urchases_a re_v erified)$$

For example, if product_A has a rating of 4, product_B has a rating of 5, and both products have been verified upon purchase, the initial weight (from average rating) is calculated as:

$$Initial_w eight(from_a verage_r ating) = \frac{4+5}{2} = 4.5$$

Since both are verified purchases, we add a boost:

$$Weight_a fter_b oost = 4.5 + 0.5 = 5$$

The final weight of the link between these two products will be 5. This approach ensures that product pairs with high ratings and verification receive a higher weight, indicating a stronger link.

- **Popularity** We also consider the average popularity of both products to adjust the connection weight.
- **Building the Connections** Every customer who reviewed more than one product helps us draw these connections. For example, if a customer reviewed products A, B, and C, then we draw connections (or links) between A-B, B-C, and A-C. We then calculate weights for each of these links.

4 Implementation

In this section, we describe the key steps of the implementation.

- **Link Creation and Weight Calculation**
  - **Link Creation:** Products reviewed by the same customer were paired, forming the network's links. If a customer reviews products A and B, it means there's some correlation, possibly indicating that they are complementary or from the same category.
  - **Weighting Strategy:** The weight between two products considers the average of their ratings and incorporates a boost if both reviews are verified. Additionally, a product's popularity (number of reviews) contributes to the weight, emphasizing more popular products.
- **Transition Matrix Formation**

  - The transition matrix is essential for PageRank computation. In this step, we construct a matrix where each entry (i,j) represents the probability of transitioning from product i to product j. This matrix is computed using the normalized weights from the link creation step, ensuring that the sum of transition probabilities from any product equals 1.

- **Iterative PageRank Computation**
  - Our PageRank computation starts with equal ranks for all products and iteratively updates these ranks based on the transition matrix and a damping factor.

4.1 Normalization

The weights of the links are normalized using Min-Max Scaling, ensuring that the values fall within a consistent range

Table 1: Product Links with Weights

product_A	product_B	weight	normalized_weight
0805073396	0375752285	5.5	0.467
0375752285	0805073396	5.5	0.467
B000FVHKFW	B000005H1Q	6.0	0.533
B000005H1Q	B000FVHKFW	6.0	0.533
B000FVHKFW	B000002OBK	5.5	0.467
B000002OBK	B000FVHKFW	5.5	0.467
B000FVHKFW	B0000023TW	5.5	0.467
B0000023TW	B000FVHKFW	5.5	0.467
B000FVHKFW	B000M02D26	6.0	0.533
B000M02D26	B000FVHKFW	6.0	0.533

4.2 Transition Matrix

Transition matrix" that captures the probability of moving from one product to another based on their connections and popularity. We first determine the total popularity score for each product by aggregating the weights of all its links. We then compute the probability of transitioning to a specific linked product by dividing the popularity score of that specific link by the product's total popularity score. This matrix offers a structured way to understand and quantify how likely one product is to be associated with or lead to another in the dataset.

Table 2: Transition Matrix

product_A	product_B	weight	normalized_weight	total_weight	probability
0060279257	B000BICOO6	4.5	0.333	0.667	0.5
0060279257	0782124313	4.5	0.333	0.667	0.5
0151011168	0972429514	4.0	0.267	1.4	0.190
0151011168	0151008124	4.5	0.333	1.4	0.238
0151011168	0375409440	5.0	0.4	1.4	0.286
0151011168	B005FDXZJU	5.0	0.4	1.4	0.286
0306812959	0912500972	6.0	0.533	131.4	0.004
0306812959	0741409399	6.0	0.533	131.4	0.004
0306812959	063121447X	6.0	0.533	131.4	0.004
0306812959	0736625968	6.0	0.533	131.4	0.004

4.3 Initialization PageRank

Initialization: All products are initially assigned equal ranks. This ensures a fair starting point.

4.4 Iterative PageRank Calculation

It begins by setting up important parameters, such as a damping factor to prevent infinite loops and the number of iterations for the PageRank algorithm. The iterative loop where it calculates the significance of each product. In each iteration, it combines a transition matrix with current product ranks, calculates contributions from one product to another, and updates the ranks based on these contributions and the damping factor. This iterative process refines product ranks over multiple cycles. Finally, the it generates a bar chart to visualize the top 10 products with the highest PageRank scores, showcasing the most influential products in the simulated user interactions. In essence, it simulates user behavior and determines the importance of products based on their interactions, offering a visual representation of the most influential products in the dataset.

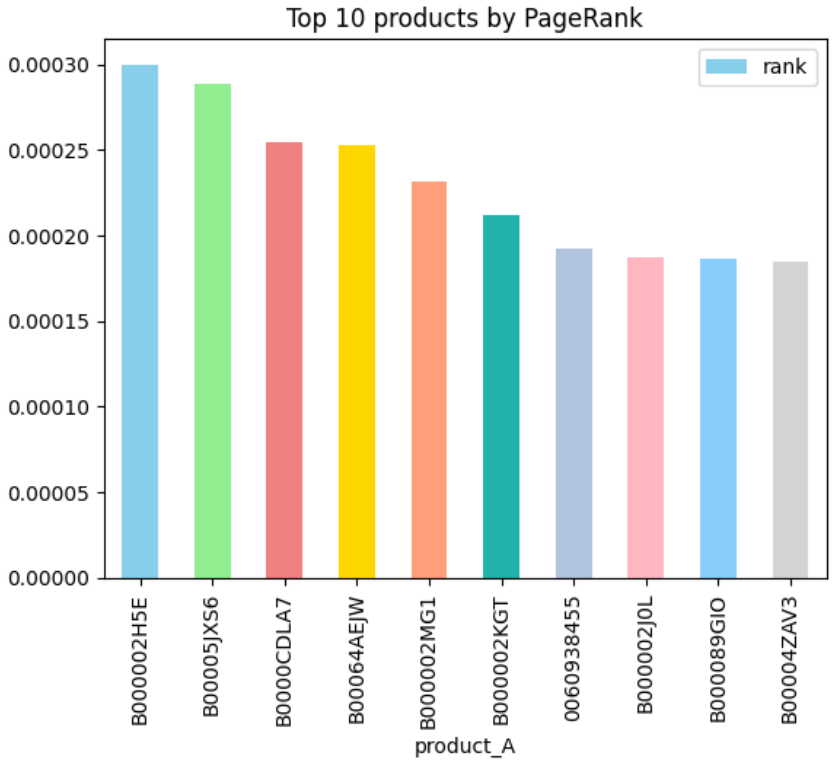

Figure 5: Top 10 products

5 Results and Discussion

The experiment highlighted several products as central nodes. These products, even if not the highest sellers, influence customer shopping behavior significantly. Such insights can aid in recommendation systems, targeted marketing, and inventory management.

6 Personal Recommendation

recommendation system that leverages customer interactions, transition data between products, and PageRank scores to suggest products that are likely to be of interest to a specific customer.in this case, custome id'31898244' used.

Table 3: Product Rankings

product_B	rank
B000003N7C	7.45×10^{-5}
1569247757	6.93×10^{-5}
0822209713	6.42×10^{-5}

6.1 Scalability

Given the implementation in PySpark, the solution scales horizontally by adding more nodes to the Spark cluster. The choice of PySpark ensures distributed processing, making it possible to handle even the full Amazon reviews dataset, beyond the sampled 1% .

6.2 Experiments

With the dataset loaded and pre-processed, the primary experiment was to compute PageRank values for each product.

Regarding the Necessity of a Damping Factor

The PageRank values are iteratively updated using the transition matrix and a damping factor (typically 0.85). This factor ensures the algorithm converges by considering the possibility of a user randomly jumping to any product.

1. Dead Ends:
    - Dead ends are nodes with no outgoing links. Given the way the dataset is structured and processed:
    - Importantly, filter out customers who have given only one review.
2. Given this filtering, there's less chance for a product to be a dead end. Why? Because for a product to appear, it has been reviewed by a customer. And because considering only customers with more than one review, it implies that every product appearing in this processed dataset is reviewed by customers who have also reviewed other products. Thus, chances of dead ends are significantly reduced, but not completely eliminated (since there could still be products reviewed by multiple customers where none of those customers reviewed any other products).
3. Spider Traps:

- Given the nature of the dataset, spider traps are less likely. A spider trap in this context would mean a subset of products that are exclusively reviewed by a subset of users, and these users haven't reviewed any other products. Given that, filtering out customers with only one review, and drawing from a diverse set of Amazon products, it's unlikely (though not impossible) that a spider trap will form.

7 conclusion

In conclusion, this report has detailed the development of a PageRank algorithm for link analysis using Apache Spark. The algorithm was applied to a dataset of customer reviews and product information to calculate product rankings. The report covered environment setup, data preprocessing, link analysis, PageRank implementation, and result visualization. Additionally, a basic recommendation system was integrated into the process. The developed algorithm holds promise for improving product recommendations in e-commerce platforms