Distributed Product Recommendation Engine

Scalable Content-Based Recommendations Using Apache Spark, HDFS, and HBase

Contents

[**Introduction 1**](#_heading=h.gjdgxs)

[**Project Overview and Objectives 1**](#_heading=h.palrhunrame)

[**Data Processing and Cleaning Actions 2**](#_heading=h.65bdj2qufo29)

[**Big Data Technology Rationale 3**](#_heading=h.cq2a49w0igwh)

[**Dataset Analysis and Insights 3**](#_heading=h.v14vsfo2tuw3)

[**Content-Based Recommendation System 6**](#_heading=h.nqw0o5ptrgyq)

[**Conclusion 7**](#_heading=h.u3newef11zd8)

[**Screenshots 7**](#_heading=h.i52lwce4q61d)

[Hadoop Distributed File System Commands 7](#_heading=h.hmnp8ov07lva)

[Apache Spark Commands 8](#_heading=h.rlro2emcthuj)

[Apache HBase Commands 8](#_heading=h.rnhj33pwwfl1)

[**References 10**](#_heading=h.n0256wii3knl)

[**Bibliography 10**](#_heading=h.wlm8ot9vpa1k)

# Introduction

There has been rapid growth in e-commerce, leading to the generation of massive amounts of data on a daily basis. Big data technologies can be leveraged to handle high-volume, high-variety, and high-velocity data for analysis and to support core system functions. My motivation for this project is to apply data technologies to analyze products and develop a content-based recommendation system. For this project, I will be utilizing the Amazon Product Sales Dataset 2023 from Kaggle (Parab, 2021), which covers product data from 11/03/2023 to 18/03/2023. The dataset contains detailed information about product listings on Amazon, including the name, main category, sub-category, image, link, rating, number of ratings, discount price, and actual price.

My primary objective is to process, analyze, and build a content-based recommendation system using this dataset by storing files in the Hadoop Distributed File System, managing data in Apache HBase, and applying Apache Spark for analysis and real-time processing.

# Project Overview and Objectives

As part of this project, I will leverage Big Data technologies and frameworks to effectively collect, store, process, and analyze the dataset. The workflow will include the following steps:

1. **Data Storage**: Load and store the dataset in the Hadoop Distributed File System (HDFS).
2. **Data Processing**: Perform data cleaning and transformation using Apache Spark.
3. **Data Persistence**: Save the processed and cleaned data into Apache HBase for efficient retrieval and management.
4. **Data Analysis & Visualization**: Retrieve data from Apache HBase, perform analytical operations with Apache Spark, and generate insightful visualizations.
5. **Recommendation Functionality**: Develop a content-based recommendation system using Apache Spark.
6. **Recommendation Storage**: Store the generated product recommendations in Apache HBase based on its product ID for easy access.

The workflow above will demonstrate Big Data technologies by utilizing Hadoop Distributed File System (HDFS) to store data, Apache Spark for high-performance processing, and Apache HBase for fast read/write access to data.

# 

# 

# 

# 

# 

# Data Processing and Cleaning Actions

Data processing and cleaning are critical steps for effective analysis and for building a content-based recommendation system. The Amazon Product Sales Dataset 2023 contained inconsistencies, missing values, and duplications that could interfere with both analysis and the recommendation functionality. I utilized Apache Spark to load the dataset stored in the Hadoop Distributed File System (HDFS) and performed the following data processing and cleaning tasks:

|  |  |
| --- | --- |
| **Initial Data Cleaning** | The initial data cleaning step I took involved dropping the column labeled “c0,” which contained duplicate IDs for the products in the dataframe. To ensure consistency, I created a new “product\_id” column and generated unique IDs for each product. |
| **Handling Missing Values** | I used Apache Spark to check for null values across all columns in the DataFrame. I found that the “rating”, “no\_of\_ratings”, “discount\_price”, and “actual\_price” columns contained null values.  Rows that were missing both “discount\_price” and “actual\_price” were removed as they didn’t provide essential pricing information for the product. For the “discount\_price” column, missing values were filled by matching its value with the corresponding “actual\_price”column as there was no discount value for the product. For the other missing values such as the “rating”, and “no\_of\_ratings”, they were filled with the value of zero to maintain data consistency. |
| **Data Validation** | For the data integrity of the dataframe, I have removed duplicate entries based on the product name to avoid bias for the recommendation functionality.  I have also conducted validations for the “rating,” and “no\_of\_ratings” columns. I ensured that the “rating” column was numeric and fell between the expected range of 0 to 5.0. For the “no\_of\_ratings” column, I have removed the commas and filtered out non-numeric values to maintain a consistent integer format. |
| **Currency Conversion** | I have converted the “discount\_price” and “actual\_price” columns from Indian Rupees to Euros for readability based on a European context for the analysis for this project. |

After cleaning and transforming the DataFrame, I exported it as a CSV file and saved it back to the Hadoop Distributed File System (HDFS). Additionally, I saved the data to Apache HBase for efficient data management and retrieval.

# Big Data Technology Rationale

In this project, I have leveraged a combination of Hadoop Distributed File System (HDFS), Apache Spark and Apache HBase each serving a specific role while working with the dataset.

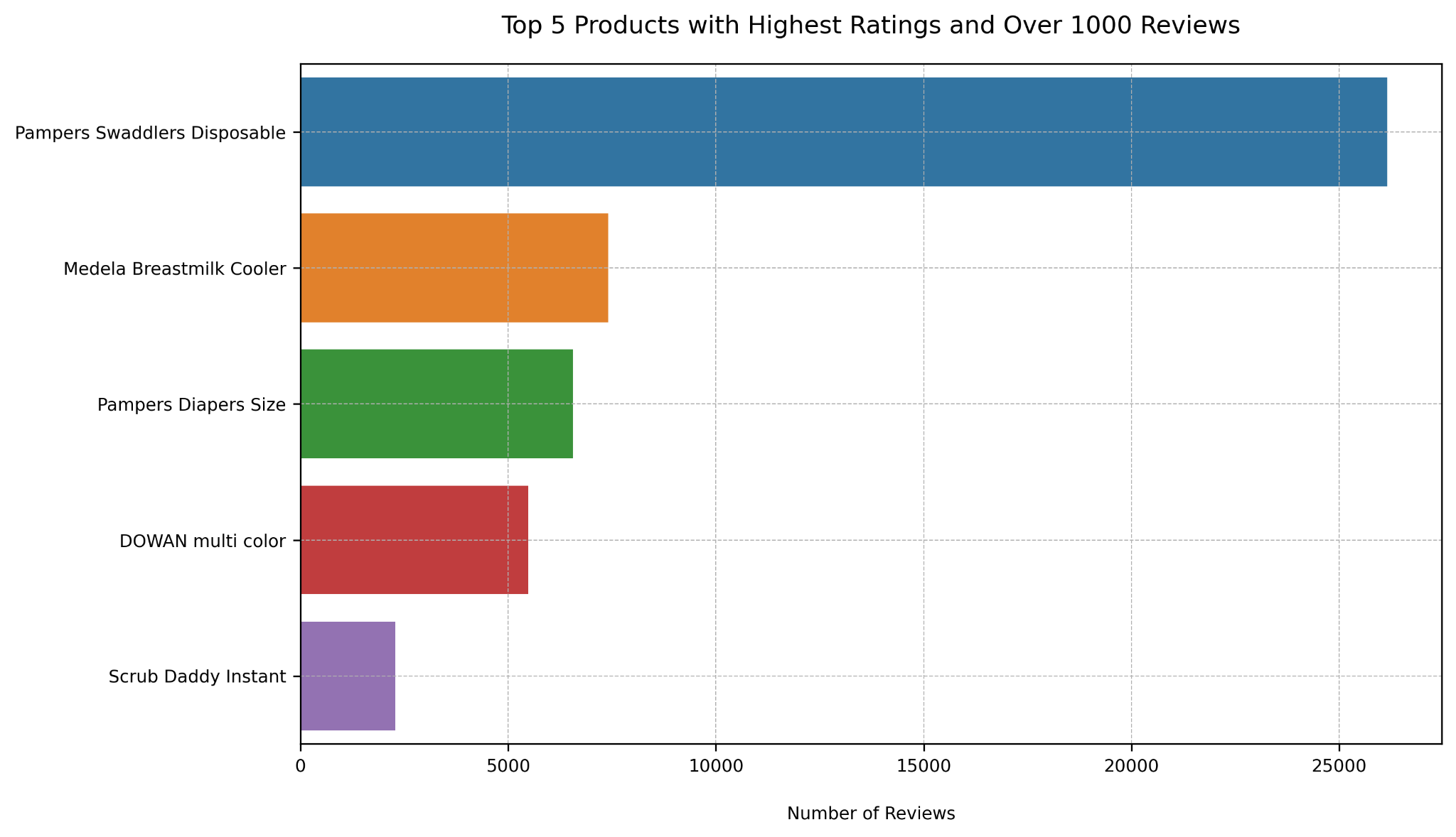
**Hadoop Distributed File System** was chosen for its scalability, fault tolerance and integration with big data tools such as Apache Spark and HBase. I have used the Hadoop Distributed File System as a base layer for storing data and to allow Apache Spark to access and process the data. For this project, I primarily used it to store the main dataset, and upload the visualizations that I have created.

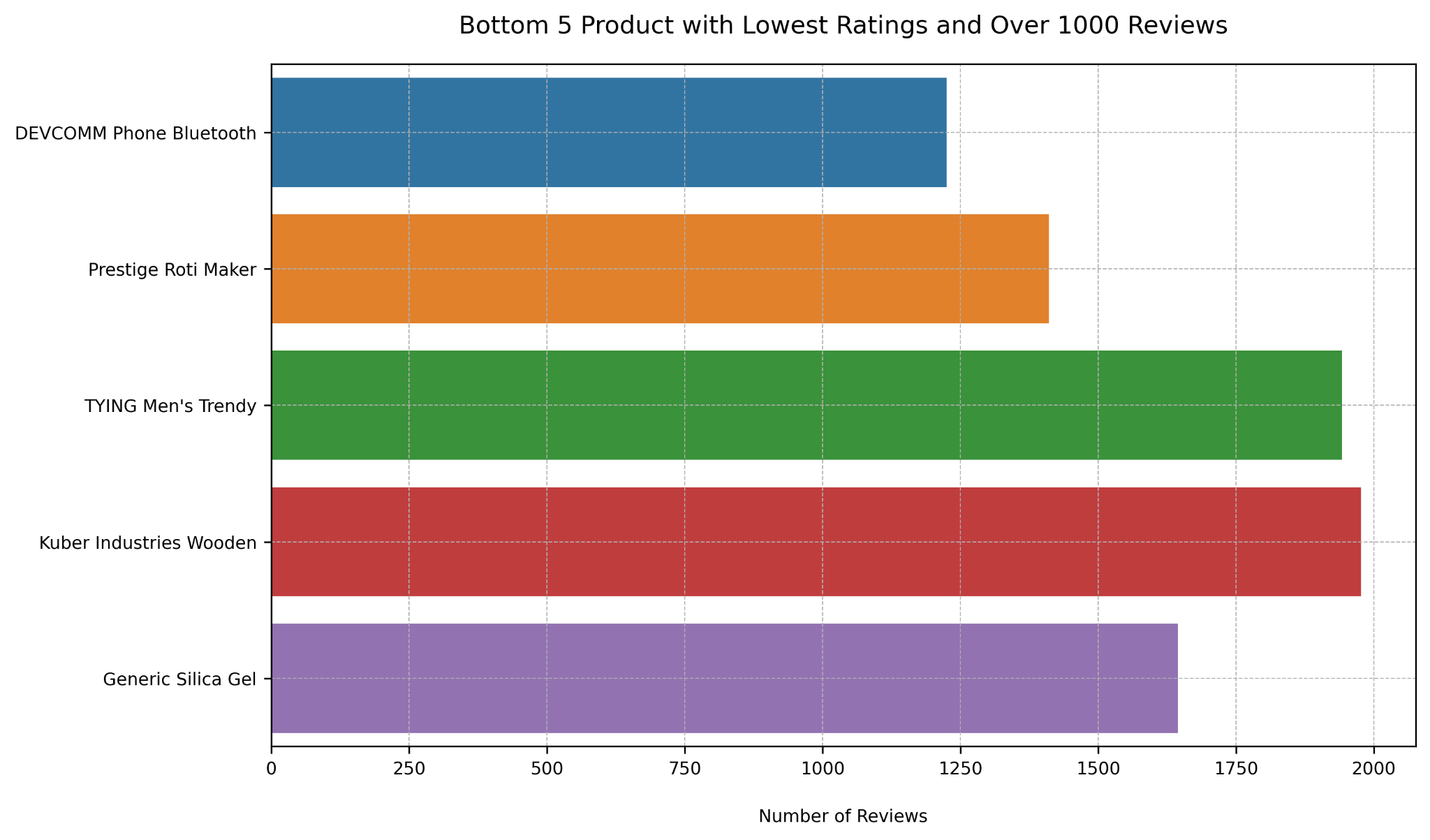
**Apache Spark** was chosen for its fast performance and scalability, making it ideal for processing a large volume of data efficiently. Its in-memory computing capability speeds up its data processing and tasks. I have also chosen Apache Spark due to its ease of use compared to implementing a Hadoop MapReduce approach. In this project I was able to conduct data cleaning, transformation, analysis and pipelines to the dataset. Additionally, Apache Spark integrates smoothly with the Hadoop Distributed File System to retrieve and store data.

**Apache HBase** was used to store the cleaned data and to update the rows with product recommendations for this assessment. I have chosen Apache HBase as its a NoSQL database built on HDFS that allows fast storing and querying individual product entries. Its schema-less structure provides flexibility in managing diverse data formats for the product information. I also used the HappyBase library for easy access to store, retrieve, and update data in Apache HBase while working with Python.

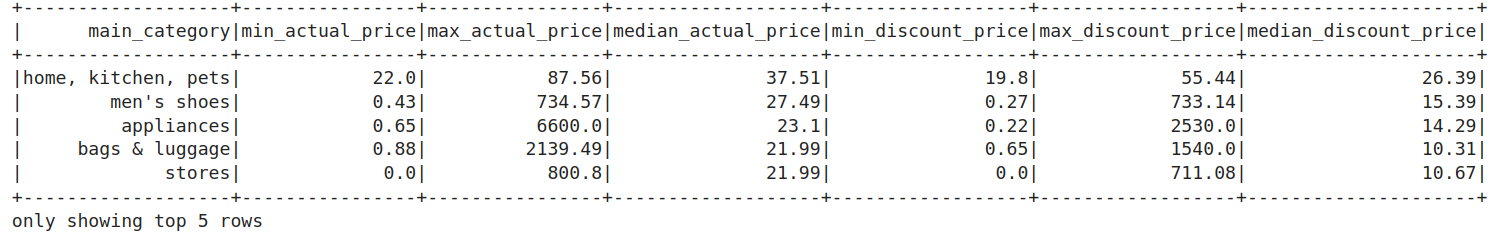
# Dataset Analysis and Insights

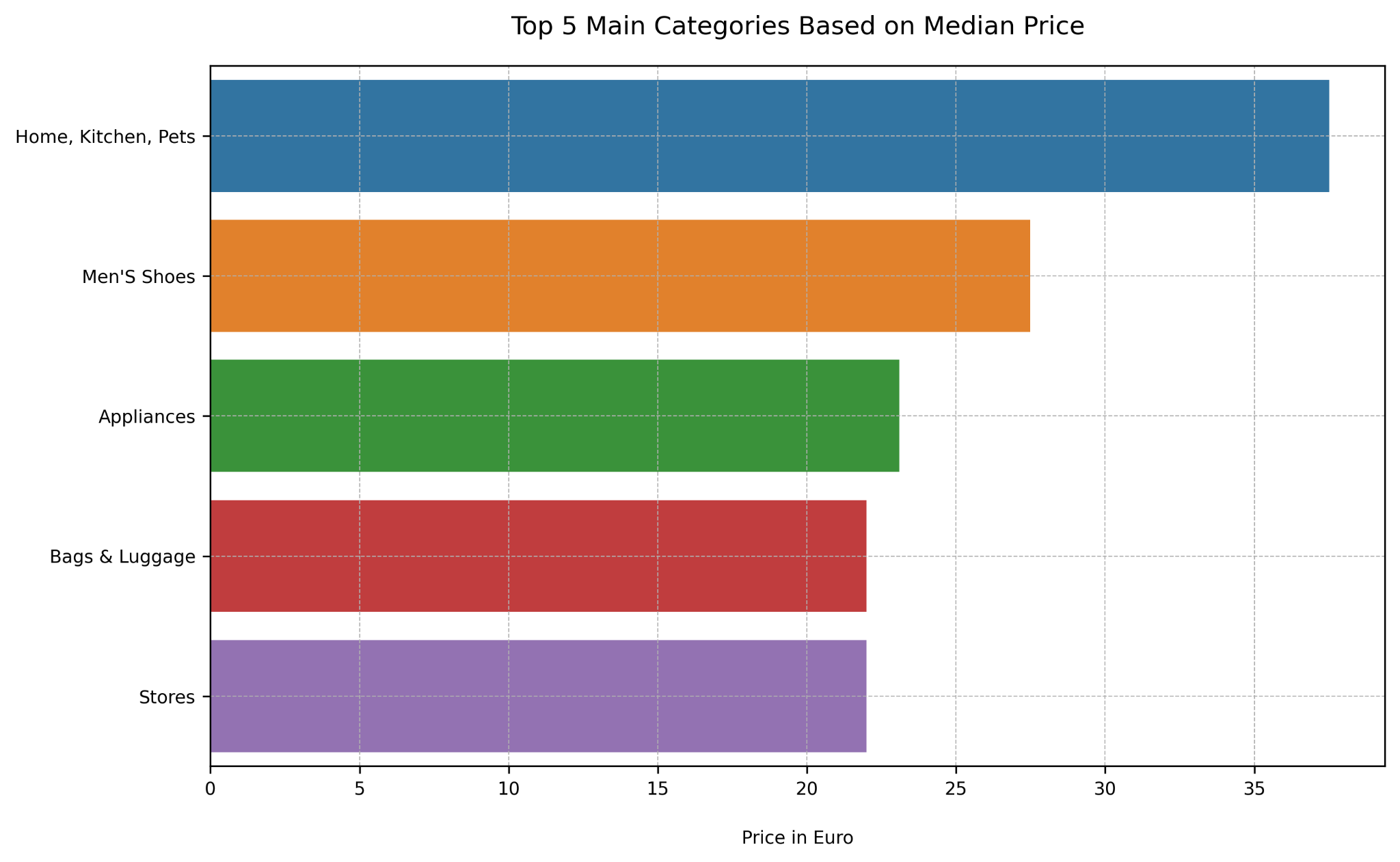
For the dataset analysis and insights, I utilized Apache Spark along with Python libraries such as Matplotlib, Seaborn, and Pandas to conduct the following analyses:

**Top 5 Products:** Identified the highest-rated products (ratings ≥ 4.5) with over 1,000 customer reviews, highlighting strong performing products in the dataset.

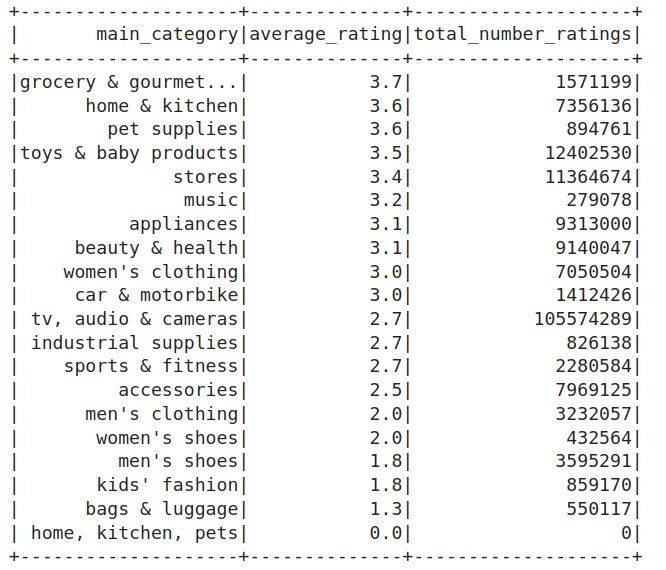
**Bottom 5 Products:** Identified the lowest-rated products (ratings >= 1) with over 1,000 customer reviews, highlighting underperforming products in the dataset.

**Top 5 Main Categories Based on Median Price:** Calculated key pricing statistics including minimum, maximum, and median values for each main category to analyze price distributions. A visualization was then created to highlight the top five main categories with the highest median prices.

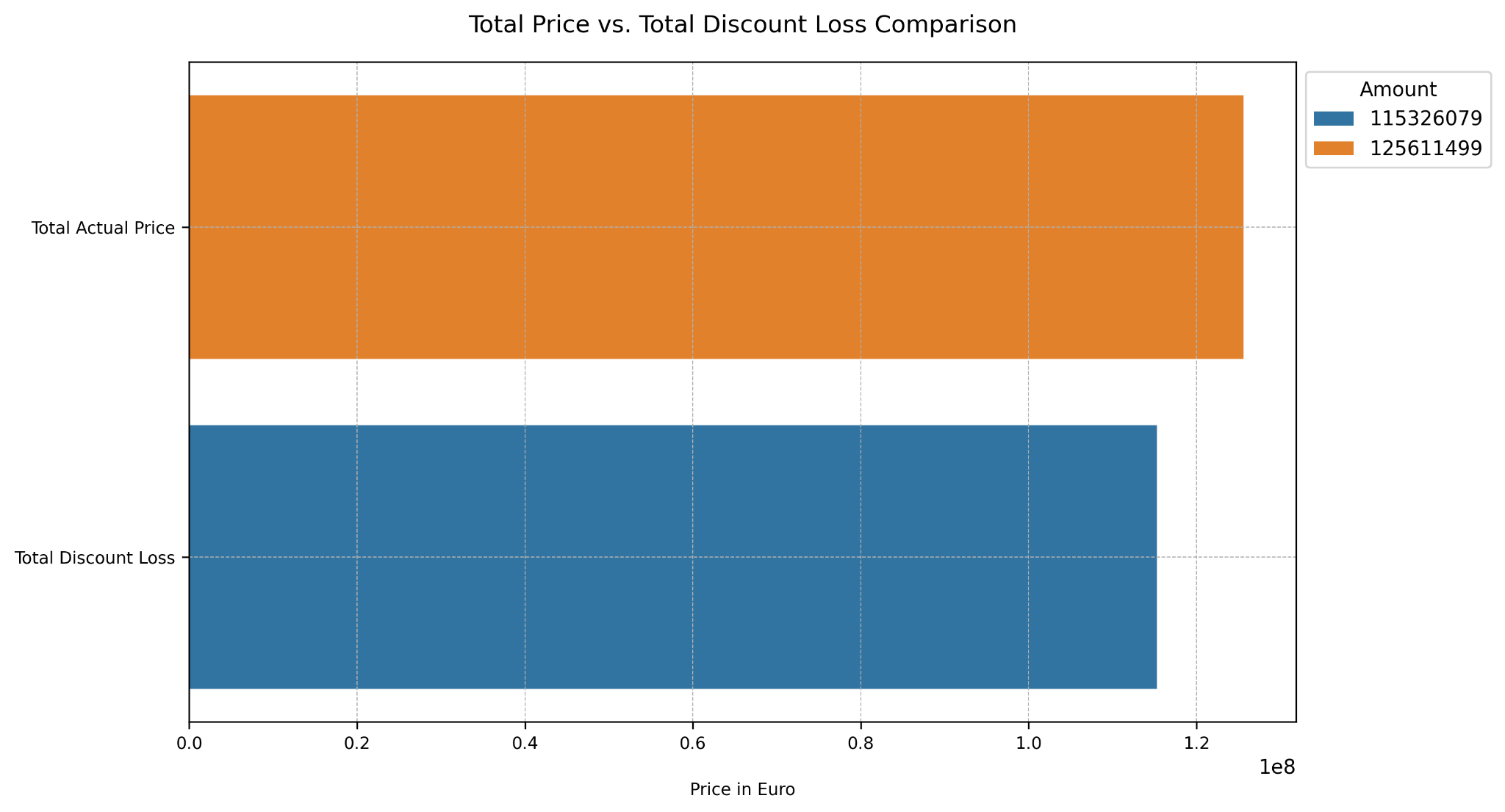




**Top 5 Main Categories Based on Average Rating:** Calculated the average rating and total number of ratings for each main category to assess overall customer satisfaction. A visualization was created to showcase the top five main categories with the highest average ratings.



**Discount Loss:** Calculated the difference between original and discounted prices across products to quantify the total potential revenue loss due to discounts.



# Content-Based Recommendation System

I have developed a content-based recommendation system using Apache Spark. This system identifies and suggests similar products based solely on product features, without relying on user interactions. The goal is to enhance product discoverability when a user is viewing a specific product. To achieve this, I implemented a cosine similarity approach to measure the similarity between products.

The product features for the comparison included main\_category, sub\_category, ratings, no\_of\_ratings, discount\_price, and actual\_price. Since main\_category and sub\_category are categorical, I first transformed them into numerical form using StringIndexer and OneHotEncoder. After encoding, I combined all the features into a single vector using VectorAssembler.

I created a Spark Pipeline to streamline the process of applying the indexing, encoding, and assembling steps to each row of the DataFrame. This allowed each product to be represented as a feature vector, which was then used to calculate the cosine similarity scores against other products. The function I created filtered out the reference product and ranked all other products based on their cosine similarity score, returning the top-N most similar products.

Once I retrieved the product recommendation data that included the product ID and cosine similarity score, it was stored in Apache HBase related to the reference product. I stored it in Apache HBase to enable fast retrieval of recommendations for future use, such as in front-end applications.

# 

# 

# Conclusion

From this project, I implemented Big Data technologies to store, process, and analyze the Amazon Product Sales Dataset 2023 by integrating Hadoop Distributed File System (HDFS), Apache Spark, and Apache HBase. I developed a data pipeline that encompassed data storage, cleaning, analysis, and product recommendation generation.

A key component of this project was the development of a content-based recommendation system using Apache Spark’s MLlib. It leveraged product attributes to suggest similar items, enabling personalized recommendations at scale. For future work, I aim to enhance the recommendation system by incorporating user interaction data, allowing for more dynamic and personalized suggestions.

Additionally, I plan to explore the broader capabilities of Apache Spark MLlib to uncover deeper insights from the dataset beyond product recommendations. This includes experimenting with clustering, classification, and regression techniques. By leveraging a wider range of features, I hope to implement additional machine learning tasks that can solve business challenges in e-commerce and other domains.

# Screenshots

## Hadoop Distributed File System Commands

1. Starting Hadoop

****

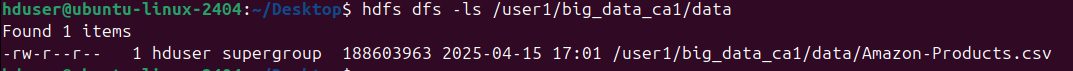
2. Creating the directory /user1/big\_data\_ca1/data

## 

3. Uploading the dataset to the folder path /user1/big\_data\_ca1/data

## 

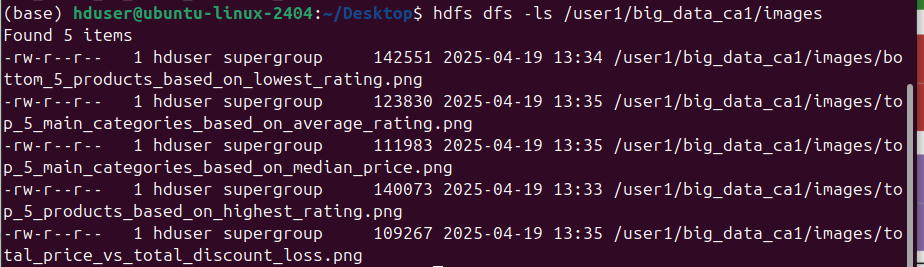
4. Confirming that the dataset has been uploaded to the directory

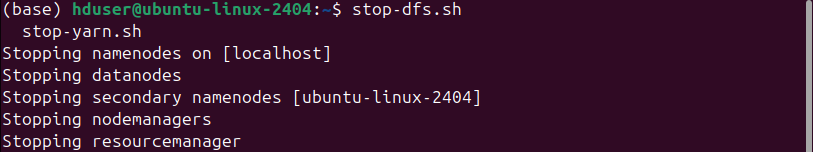


5. Creating the directory /user1/big\_data\_ca1/images



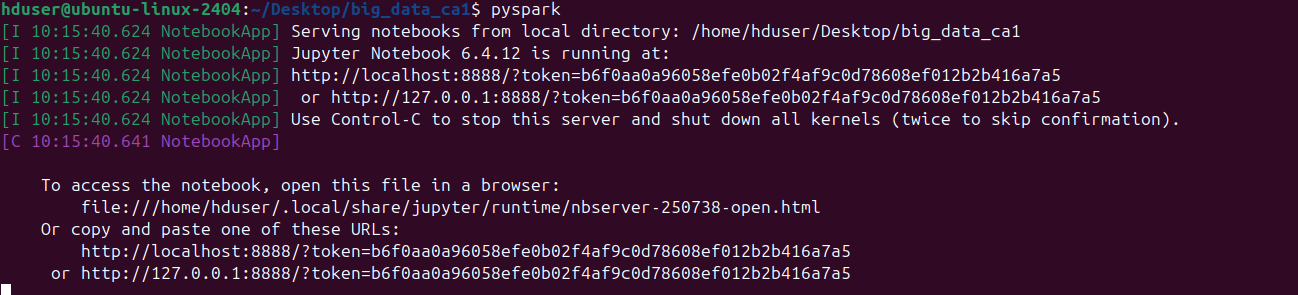
6. Listing the directory /user1/big\_data\_ca1/images



7. Stopping Hadoop

## Apache Spark Commands

1. Turning on Apache Spark and opening Jupyter notebook



## Apache HBase Commands

1. Starting HBase



2. Starting HBase Thrift Server

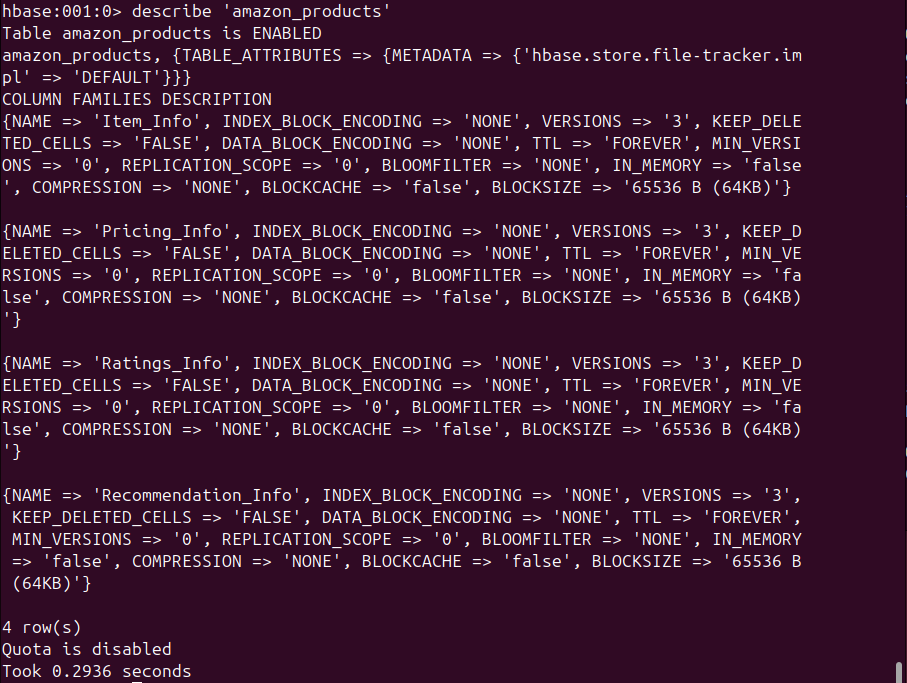


3. Starting HBase Shell

4. Listing HBase Table



5. Describe table “amazon\_products”



# 

# 

# References

* *Parab, L., 2021. Amazon Products Dataset. [online] Kaggle. Available at:* [*https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset?select=Amazon-Products.csv*](https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset?select=Amazon-Products.csv) *[Accessed 21 Apr. 2025].*

# Bibliography

* *Hien Luu (2021) Beginning Apache Spark 3 : With DataFrame, Spark SQL, Structured Streaming, and Spark Machine Learning Library. New York: Apress.*
* *Parab, L., 2021. Amazon Products Dataset. [online] Kaggle. Available at:* [*https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset?select=Amazon-Products.csv*](https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset?select=Amazon-Products.csv) *[Accessed 21 Apr. 2025]*
* *Taunk, A.K., Parmar, A.K. and Srivastav, R. (2013) ‘The Hadoop Distributed File System’, International Journal of Computer (IJC); Vol. 8 No. 1 (2013); 8-152307-45232307-4523 [Preprint]. Available at:* [*https://research.ebsco.com/linkprocessor/plink?id=e0eea4ec-02aa-3870-9674-dcd91a2362c2*](https://research.ebsco.com/linkprocessor/plink?id=e0eea4ec-02aa-3870-9674-dcd91a2362c2) *(Accessed: 20 April 2025).*
* *Vohra, D. (2016) Apache HBase primer. Berkeley, CA: Apress.*
* *Yang, Y., Jo, J. and Lim, H. (2022) ‘Unifying user preference and item knowledge-based similarity models for top-N recommendation’, Personal & Ubiquitous Computing, 26(2), pp. 407–416. doi:10.1007/s00779-019-01252-x.*