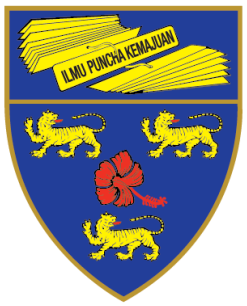


UNIVERSITI MALAYA

MASTER OF DATA SCIENCE (SEMESTER 2 – 2022/2023)

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

WQD7007 BIG DATA MANAGEMENT

GROUP ASSIGNMENT

PRODUCT RECOMMENDATION SYSTEM

GROUP MEMBERS:

STUDENT ID	NAMES
S2190151	WEE HIN SHEIK
S2196123	SAMUEL TAN JOO WOON
22051081	TAN KAI YING
22050480	LEE TSE LYN

LECTURER: DR. SHIVAKUMARA PALAIAHNAKOTE

DATE: 23 JUNE 2023

Table of Contents

1 Introduction 3

1.1 Problem Statement 3

1.2 Project Objective 4

2 Big Data Architecture & Methodology 4

2.1 Architecture of the project 4

2.2 Technology and tools used 5

2.2.1 Hadoop tools 5

2.2.2 Supporting tools for Hadoop 6

2.2.3 Infrastructure tools 7

2.2.4 Analytics and Visualization tools 8

2.3 Infrastructure of the project 9

2.4 Product Recommendation Algorithm 10

2.4.1 ChatGPT API 10

2.4.2 Fuzzy Search Algorithm 12

3 Results & Discussions 13

3.1 Datasets 13

3.2 Trending Results 14

3.3 Data Pipeline 15

3.4 Result – The data application 17

3.4.1 Streamlit 17

3.5 Comparison of MySQL and MongoDB as Data Source 18

4 Conclusions and Future Work 19

5 References 20

1 Introduction

Product recommendation systems have revolutionized the e-commerce industry, where businesses are utilizing artificial intelligence to recommend products to customers based on their browsing history and purchase patterns (Lee and Hosanagar, 2017). These systems not only enhance user experience but also have a significant impact on sales. Integrating product recommendation systems with other marketing tactics, such as email campaigns and social media ads, can further enhance their impact on sales (Lee and Hosanagar, 2017). Background information shows that with advancements in technology, companies are now able to cater to individual customer preferences by using sophisticated algorithms.

Based on the research done by Knotzer (2018), it is estimated that product recommendations account for up to 31% of total e-commerce site revenue. The rise in popularity of these recommendation systems has led many companies across all sectors to adopt them, from fashion retailers like ASOS and ZARA to online marketplaces such as Amazon (Knotzer, 2018). However innovative these systems may be, they still require human intervention and specialized knowledge for successful implementation. According to Lee (2014), "personalized product recommendations have been shown to increase customer engagement and sales conversion rates." as customers want to feel seen and understood by the brands they interact with. By providing personalized recommendations, businesses are able to demonstrate that they not only understand their customers' needs but also care about fulfilling them in a way that's tailored just for them. Furthermore, if those suggestions align with their interests or previous buying behaviors, it reinforces their trust in the brand's ability to cater to their individual needs (Lee, 2014).

1.1 Problem Statement

E-commerce companies have thousands and millions of products in their catalogues. However, not all the products are in line with the current trends. Customers would spend hours scrolling and looking for the correct products. As a result, customers would be frustrated, and customer satisfaction would be negatively impacted. Therefore, it is essential to leverage on the potential of big data to develop a product recommendation system.

1.2 Project Objective

To streamline product discovery and increase customer satisfaction, this project has 2 main objectives. They are listed as following:

1. To design and develop a product recommendation system that is based on current trends.
2. To publish the recommended products in online platform.

2 Big Data Architecture & Methodology

This chapter discusses the methodology and implementation details of the project. Section 2.1 discusses the overall architecture of the project. Furthermore, tools used in the project will be discussed in section 2.2. The system is hosted on the home server and all the tools and applications can be accessed from the public internet where the infrastructure used to power the system will be discussed further in section 2.3.

2.1 Architecture of the project

This section discusses the overall architecture and methodology of the product recommendation system as shown in Figure 2.1. The project simulates the data pipeline flow in the real-life situations by segregating the data pipeline into five parts, which are data source, data integration, data warehouse, analytics environment, and visualization & deployment. The data source consists of MySQL and MongoDB where MySQL is used to store the product data, which is used to mimic the product database in the ecommerce website. On the other hand, MongoDB is used to store trending items retrieved from Google Trends where it is scheduled to run daily. To integrate the data from multiple sources into data warehouse, Apache Nifi comes into place combining the data from MySQL and MongoDB in Hive, the data warehouse in this project. After the data pipeline is done, the data in data warehouse is then used to determine the product to recommend according to the current trend. To identify the product to recommend, ChatGPT is come into place to convert Google Trend item into related products. The related product keywords are then used to find similar products in the product table using fuzzy search algorithm (Paiva, 2013). After a list of similar products is generated, the list is saved into MongoDB for visualization and deployment

purposes. The recommended products are then displayed with streamlit applications. More details of the result will be discussed in Chapter 3.

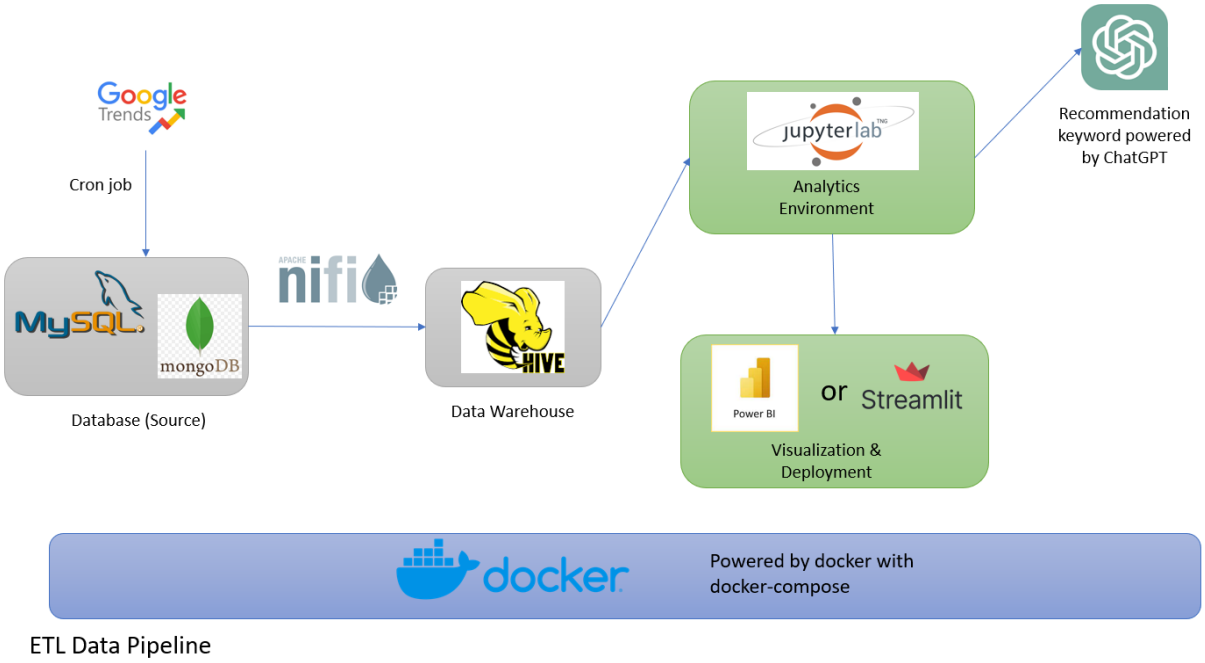


Figure 2.1: Architecture of the project

2.2 Technology and tools used

The technology and tools used in the project can be separated into four categories, which are tools in Hadoop environments, supporting tools that are used to ease the uses of Hadoop tools such as UI to access the server, infrastructure tools that used to power the infrastructure for the project and analytics and visualization tools that used to turn it into a data application.

2.2.1 Hadoop tools

2.2.1.1 MySQL

MySQL is an open-source relational database management system used to store, manage, and retrieve structured data efficiently. In this project, it is used as one of the data sources that stores product data to simulate operational database in real life and compare with MongoDB.

2.2.1.2 MongoDB

MongoDB is a NoSQL document-oriented database system designed for flexible and scalable data storage and retrieval. In this project, it is used as one of the data sources that store trend data to simulate operation database in real life and compare with MySQL.

2.2.1.3 Apache Nifi

Apache NiFi is an open-source data integration platform that facilitates the seamless flow, transformation, and routing of data between various systems. In this project, Nifi is used to integrate data from MySQL and MongoDB to Hive. This is to simulate the data pipeline from transactional database to data warehouse.

2.2.1.4 Hive (Data Warehouse)

Hive is a data warehouse infrastructure built on top of Hadoop that provides a SQL-like query language for analyzing and processing large datasets. In this project, Hive is used as the data warehouse for analytics purposes.

2.2.1.5 Hadoop

Hadoop is a distributed computing framework that allows for the storage and processing of large-scale data across clusters of commodity hardware. In this project, HDFS is used for storage and MapReduce is used to power Hive as the processing engine.

2.2.2 Supporting tools for Hadoop

2.2.2.1 PhpMyAdmin

PhpMyAdmin is a web-based graphical interface that allows users to easily manage and interact with MySQL through a browser.

2.2.2.2 MySQL Workbench

MySQL Workbench is a comprehensive visual tool that enables developers and database administrators to design, model, and manage MySQL databases in a user-friendly interface. It allows access to MySQL database from the local computer.

2.2.2.3 Hue

Hue is a web-based user interface that provides a comprehensive platform for interacting with Apache Hadoop and its ecosystem, facilitating easy data querying, analysis, and job scheduling. In this project, Hue is used to provide access to Hive data warehouses through a browser.

2.2.2.4 DbVisualizer

DbVisualizer is an SQL client and database management software. The software comes with a commercial and free version. In this project, the free version of the software is used to access the Hive data warehouse from the local computer.

2.2.3 Infrastructure tools

2.2.3.1 Home Server

The home server is running with proxmox, an open-source, Type 1 hypervisor where the project is running in Ubuntu Server 22.04 LTS virtual machine. Figure 2.2 shows the screenshot of the proxmox that used to run the virtual machine.

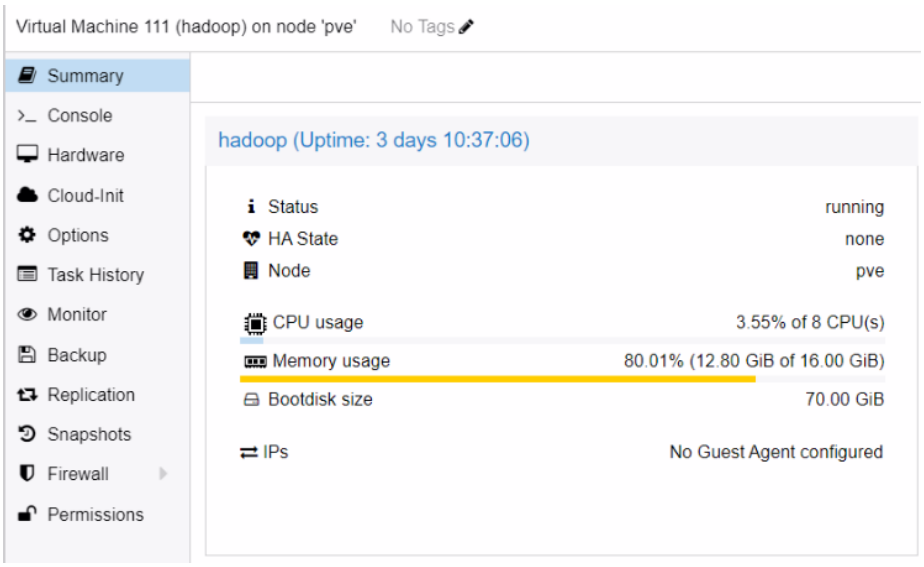


Figure 2.2: Screenshot of Proxmox and virtual machines

2.2.3.2 Docker and docker-compose

Docker is an open-source platform that enables developers to automate the deployment and management of applications within lightweight, isolated containers. Docker Compose is a tool that simplifies the orchestration of multiple Docker containers, allowing for the definition and management of multi-container applications. In this project, docker compose is used to run the Hadoop and development application in the server.

2.2.4 Analytics and Visualization tools

2.2.4.1 JupyterLab

JupyterLab is an interactive, web-based development environment that allows users to create and share documents containing live code, visualizations, and narrative text. In this project, JupyterLab is used as the coding platform where all the team members can write the code in the online platform to avoid environment conflicts in local system of the team members.

2.2.4.2 ChatGPT API

ChatGPT is a powerful language model developed by OpenAI that utilizes deep learning techniques to generate human-like responses and engage in conversational interactions with users

(OpenAI, 2023). In this project, ChatGPT is used to get recommendations for products to buy based on current trending items.

2.2.4.3 Streamlit

Streamlit is an open-source software framework mainly utilized for data science and machine learning (Sarangpure et al., 2023). Streamlit uses a few lines of code to develop the applications, then the code is immediately updated in the current kernel when one types in the source file when the specified source file is saved. After then, the output will appear on the screen.

2.3 Infrastructure of the project

This section discusses the infrastructure used in the project. The project is hosted on the home server and tools can be accessed from the internet with the URL provided in table 2.1. The application and the tools are powered by docker running in container with docker-compose.

Tools used	**URL to access the tools**
MySQL	tersakiti.top:3306
MongoDB	tersakiti.top:27017
Apache Nifi	http://nifi.tersakiti.top:8080/nifi
JupyterLab	http://jupyterlab.tersakiti.top:8080
Mongo Express	http://mexpress.tersakiti.top:8080
Hue	http://hue.tersakiti.top:8080
PhpMyAdmin	http://phpmyadmin.tersakiti.top:8080
HDFS Namenode	http://namenode.tersakiti.top:8080

Table 2.1: URL to access the tools used in the project.

2.4 Product Recommendation Algorithm

2.4.1 ChatGPT API

In this project, ChatGPT (OpenAI, 2023) is used to turn the trending item obtained from Google Trend into products to buy. This gives an idea of which product to buy based on the current trend. The example is shown in Figure 2.3. However, ChatGPT is not accessed through Graphical User Interface (GUI), but Application Programming Interface (API) as shown in Figure 2.4. To call ChatGPT API, an OpenAI account is needed. The account can be created by registering at OpenAI website. Until the time of writing, the usage of ChatGPT API is costing 0.04 USD as shown in Figure 2.5. The model used in this project is *gpt-3.5-turbo* which is the default model for ChatGPT.

Figure 2.3: The use of ChatGPT to recommend products to buy based on current trend.

```
message = "Can you recommend product to buy based on the given word, be short and concise, using ':' as delimiter for each item during answer: "
message += keywords
# print(message)

gpt_response = call_chat_gpt(message, openai_key)
print(gpt_response)
```

```
Malta vs England: England
Poland vs Germany: Germany
Tesla: Model S
F1: Mercedes
KidZania: Roleplay
Kick: Soccer
Fall: Autumn
Azizulhasni Awang: Cycling
Nicolas Jackson: Actor
Boris Johnson: Prime Minister
Black Clover: Sword of the Wizard King: Anime
Ross Butler: Actor
Ted Lasso: TV show
Kourtney Kardashian: Reality star
France: Paris
Extraction 2: Action movie
Happy Father's Day: Gifts
Adipurush film: Bollywood
England vs Australia: Cricket
Spirit: Horse
```

Figure 2.4: Call to ChatGPT API and sample response

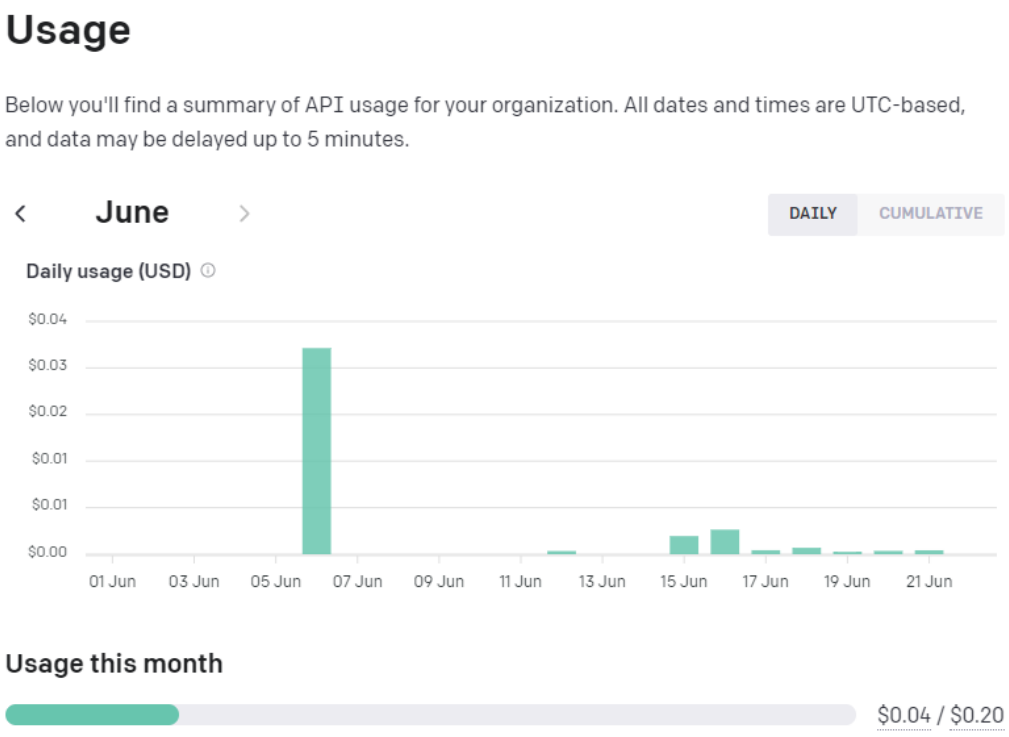


Figure 2.5: Usage of ChatGPT

2.4.2 Fuzzy Search Algorithm

Fuzzy search (Paiva, 2013) is an algorithm that is used to find patterns in strings (approximate string matching). The algorithm works by calculating the minimum cost associated with creating a string equal to a pattern. Fuzzy search algorithms can find relevant strings even if the original string contains typo errors and misspellings.

The fundamental principle behind fuzzy search algorithms is the use of a string metric, a mathematical measure of difference and similarity between two strings. One of the most used metrics is the Levenshtein distance (V. I. Levenshtein, 1965). This measures the minimum number of single-character edits, including insertions, deletions, or substitutions, needed to change one word into the other. In simple terms, the fewer the changes required, the closer the match.

A fuzzy search algorithm works by calculating the string metric between the search term and each piece of data in the set. Matches are then scored based on this calculation, with lower scores indicating a closer match. A threshold is typically set to determine which scores are good enough to be returned as matches. In this project, the python package, fuzzy wuzzy is used to find the similar product in the product database according to the product recommend by ChatGPT based on the trending item. The threshold for the Levenshtein distance is set at 60. Figure 2.6 shows the python function code that finds a similar product in the product table.

```python
def find_similar_product(product_keyword, products_df):
    # max_ratio = 0
    # max_ratio_product = None
    recommended_product_df = pd.DataFrame(columns=["ratio_score", "product_id"])
    products_dict = products_df.to_dict('records')

    for row in products_dict[:]:
        ratio = fuzz.token_sort_ratio(row['product_name'], product_keyword)

        if ratio > 60:
            recommended_product_df = pd.concat([recommended_product_df, pd.DataFrame({"ratio_score": [ratio], "product_id": [row['product_id']]})]

    return recommended_product_df.sort_values(by=['ratio_score'], ascending=False)
```

Figure 2.6: Python function that find similar product with Fuzzy Search Algorithm

3 Results & Discussions

This chapter presents the result and discussion for the project. Section 3.1 discusses the dataset used in the projects. Next, the result of trending items from Google Trend and data pipeline are discussed in section 3.2 and 3.3 respectively. Furthermore, section 3.4 discusses the final product of data application developed with streamlit. Lastly, section 3.5 compares using MySQL and MongoDB as data sources.

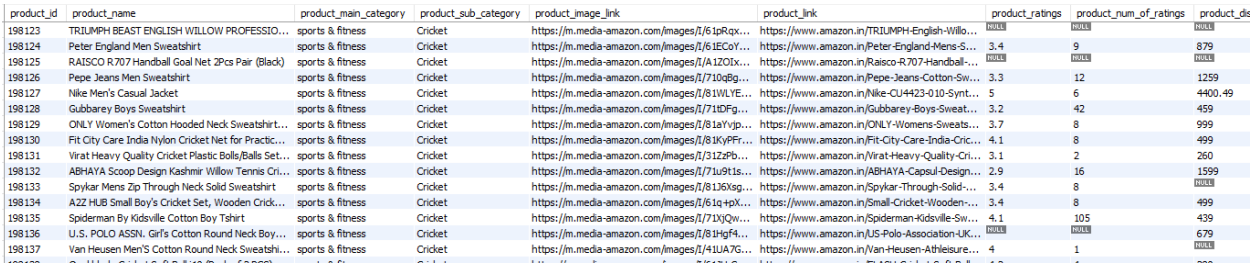
3.1 Datasets

The dataset used in this project is taken from Kaggle of Amazon product sales dataset 2023 (Parab, 2023). The dataset contains 142 categories of Amazon products with more than 300,000 products. The dataset contains 11 attributes which are listed in table 3.1. *product_id* attribute is added after combining all the csv file from the Kaggle into one big dataset. The dataset is then split into two datasets by splitting from the middle. This is to split the dataset into two different inputs for two groups. For our group, we are using the second part of the dataset as the input, the example of the dataset is shown as Figure 3.1. As can see from Figure 3.1, the product id starts from 198,123 instead of 1, this is to indicate that the input dataset is different with another group. As the input dataset is different, hence the recommended product at the final product of data application is also different.

Attribute name	**Description**
product_id	The unique identifier of the product
product_name	The name of the product
product_main_category	The main category of the product belongs
product_sub_category	The subcategory of the product belongs
product_image_link	The image of the product look like
product_link	The amazon website reference link of the product
product_ratings	The ratings given by amazon customers of the product

product_num_of_ratings	The number of ratings given to this product in amazon shopping
product_discount_price	The discount prices of the product
product_actual_price	The actual MRP of the product

Table 3.1: Attributes in the dataset



product_id	product_name	product_main_category	product_sub_category	product_image_link	product_link	product_ratings	product_num_of_ratings	product_dis
198123	TRIUMPH BEAST ENGLISH WILLOW PROFESSIO...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/61pRqx...	https://www.amazon.in/TRIUMPH-English-Willo...	NULL	NULL	NULL
198124	Peter England Men Sweatshirt	sports & fitness	Cricket	https://m.media-amazon.com/images/I/61ECoY...	https://www.amazon.in/Peter-England-Mens-S...	3.4	9	879
198125	RAISCO R.707 Handball Goal Net 2Pcs Pair (Black)	sports & fitness	Cricket	https://m.media-amazon.com/images/I/A1ZOIx...	https://www.amazon.in/Raisco-R.707-Handball-...	NULL	NULL	NULL
198126	Pepe Jeans Men Sweatshirt	sports & fitness	Cricket	https://m.media-amazon.com/images/I/710qBg...	https://www.amazon.in/Pepe-Jeans-Cotton-Sw...	3.3	12	1259
198127	Nike Men's Casual Jacket	sports & fitness	Cricket	https://m.media-amazon.com/images/I/81WLYE...	https://www.amazon.in/Nike-CU4423-010-Synt...	5	6	4400.49
198128	Gubbarey Boys Sweatshirt	sports & fitness	Cricket	https://m.media-amazon.com/images/I/71tDFg...	https://www.amazon.in/Gubbarey-Boys-Sweat...	3.2	42	459
198129	ONLY Women's Cotton Hooded Neck Sweatshirt...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/81aYvjp...	https://www.amazon.in/ONLY-Womens-Sweats...	3.7	8	999
198130	Fit City Care India Nylon Cricket Net for Practic...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/81KyPFr...	https://www.amazon.in/Fit-City-Care-India-Cric...	4.1	8	499
198131	Virat Heavy Quality Cricket Plastic Bolls/Balls Set...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/31ZzPb...	https://www.amazon.in/Virat-Heavy-Quality-Cri...	3.1	2	260
198132	ABHAYA Scoop Design Kashmir Willow Tennis Cri...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/71u9t1s...	https://www.amazon.in/ABHAYA-Capsul-Design...	2.9	16	1599
198133	Spykar Mens Zip Through Neck Solid Sweatshirt	sports & fitness	Cricket	https://m.media-amazon.com/images/I/81J6Xsg...	https://www.amazon.in/Spykar-Through-Solid-...	3.4	8	NULL
198134	A2Z HUB Small Boy's Cricket Set, Wooden Crick...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/61q+pX...	https://www.amazon.in/Small-Cricket-Wooden-...	3.4	8	499
198135	Spiderman By Kidsville Cotton Boy Tshirt	sports & fitness	Cricket	https://m.media-amazon.com/images/I/71XjQw...	https://www.amazon.in/Spiderman-Kidsville-Sw...	4.1	105	439
198136	U.S. POLO ASSN. Girl's Cotton Round Neck Boy...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/81Hgf4...	https://www.amazon.in/US-Polo-Association-UK...	NULL	NULL	679
198137	Van Heusen Men'S Cotton Round Neck Sweatshi...	sports & fitness	Cricket	https://m.media-amazon.com/images/I/41UA7G...	https://www.amazon.in/Van-Heusen-Athleisure...	4	1	NULL

Figure 3.1: Second part of Amazon product dataset

3.2 Trending Results

The search trends are retrieved from Google Trends using pytrends library. The trending keywords are scraped from Google Trends daily. The date is included to differentiate the data from one another. The results are kept in MongoDB under the “GoogleTrend” database and “Trending” collection. A sample of the keywords data stored in MongoDB is shown in Figure 3.2.

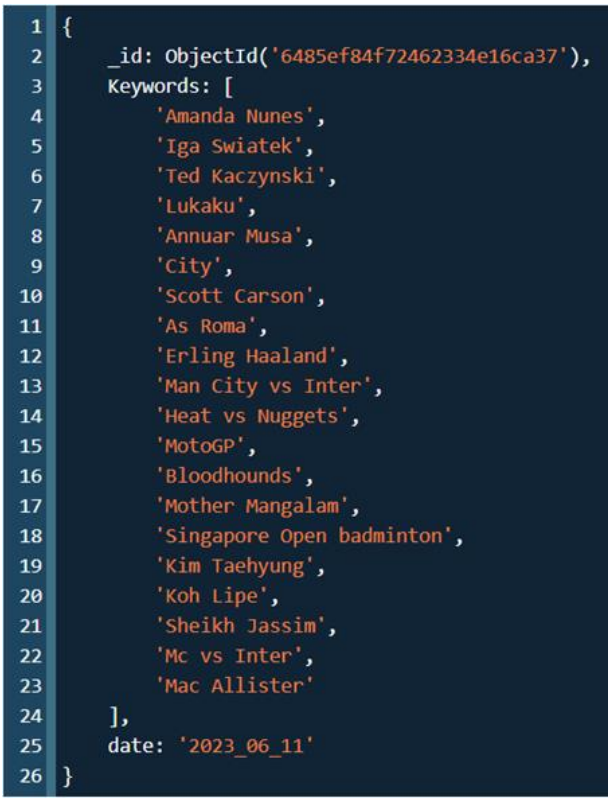


Figure 3.2: A sample of keywords data stored in MongoDB.

3.3 Data Pipeline

Apache Nifi is used to integrate the data from MySQL and MongoDB to Hive. There are two pipelines used in this project as there are two data sources. The first one is the MySQL to Hive data pipeline where the product data is first query from MySQL database then it is converted to csv and saved in HDFS. Then, the product data csv is loaded into Hive using LOAD DATA INTO TABLE query. As the product data is huge and contains a lot of data, it is more efficient to use LOAD DATA INTO TABLE to insert the product data into Hive, instead of INSERT INTO. Figure 3.3 shows the pipeline from MySQL to Hive.

The second pipeline is for trending data from MongoDB to Hive. The pipeline is scheduled to run daily. This pipeline will take the latest trending item from Google Trend from MongoDB and put it into Hive by converting the latest trending item in JSON format into HiveQL statement using INSERT INTO. Figure 3.4 shows the pipeline from MongoDB to Hive.

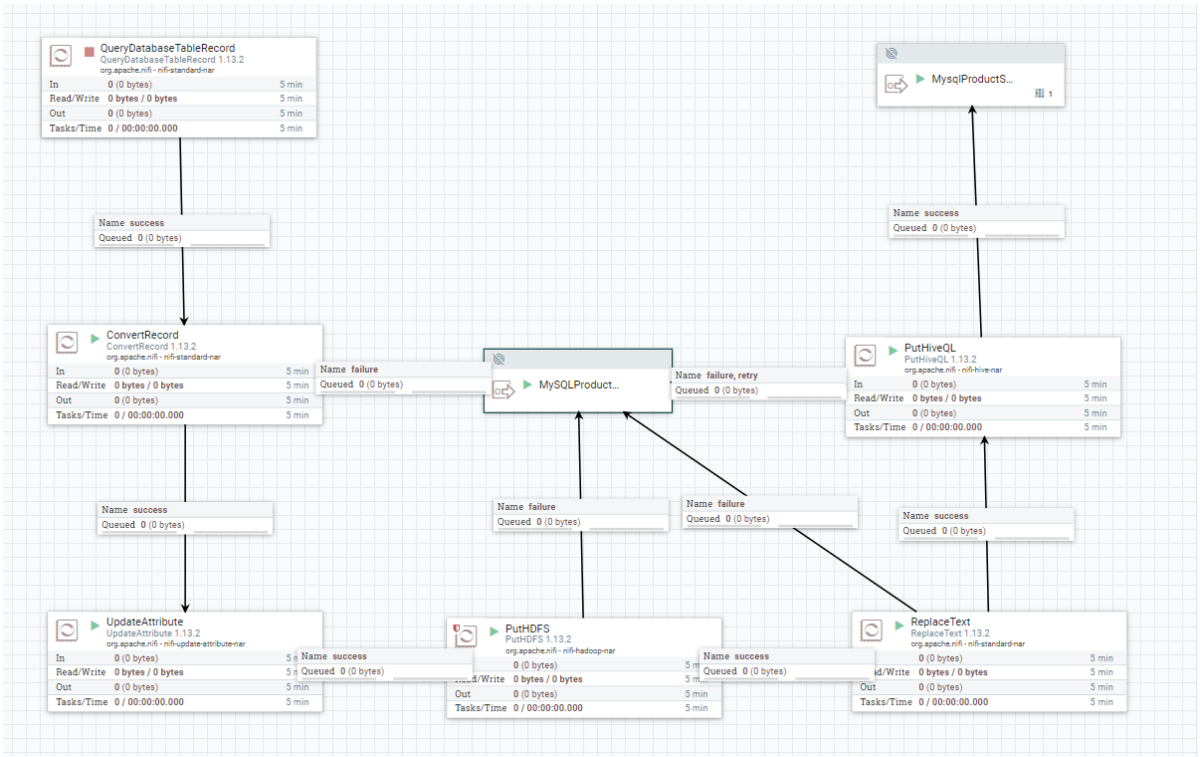


Figure 3.3 Data pipeline from MySQL to Hive

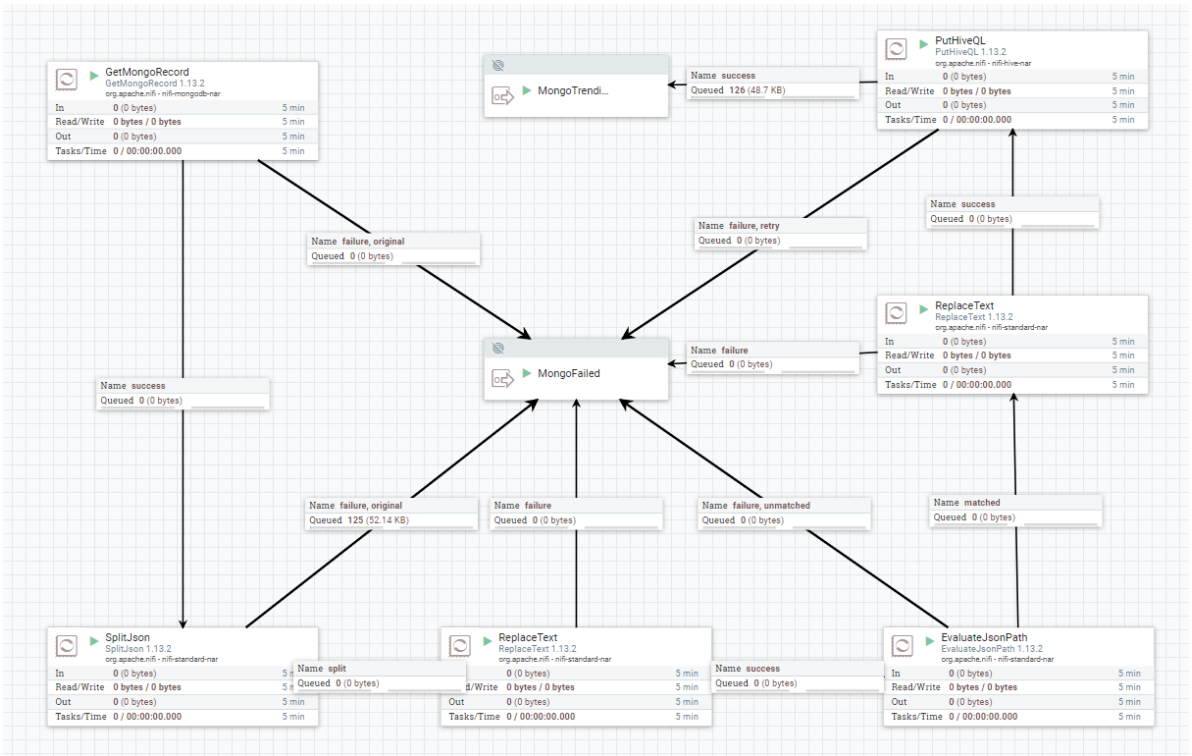


Figure 3.4: Data pipeline from MongoDB to Hive

3.4 Result – The data application

3.4.1 Streamlit

The data web application, product recommendation system was developed with Streamlit as shown in Figure 3.5. The final web application can be accessed with following URL: https://group3.streamlit.app . The user can use this website to know current trends in the left sidebar. Besides that, the users can select how many suggested products that users would like to see based on the trends. On the right side, the product appears based on the trends. This web application provides detailed information about the product. For example, this web application provides a category for each product, the original price and discount price, and a product review. The users can select this product, which will be linked to the Amazon page. Our group used the same web application designed as Group 2. However, there are some different features of the web application. For example, our group was using a different database from group 2. Therefore, the is a different product suggested for the users. Besides that, different amounts of suggested products are available for user selection compared to group 2.

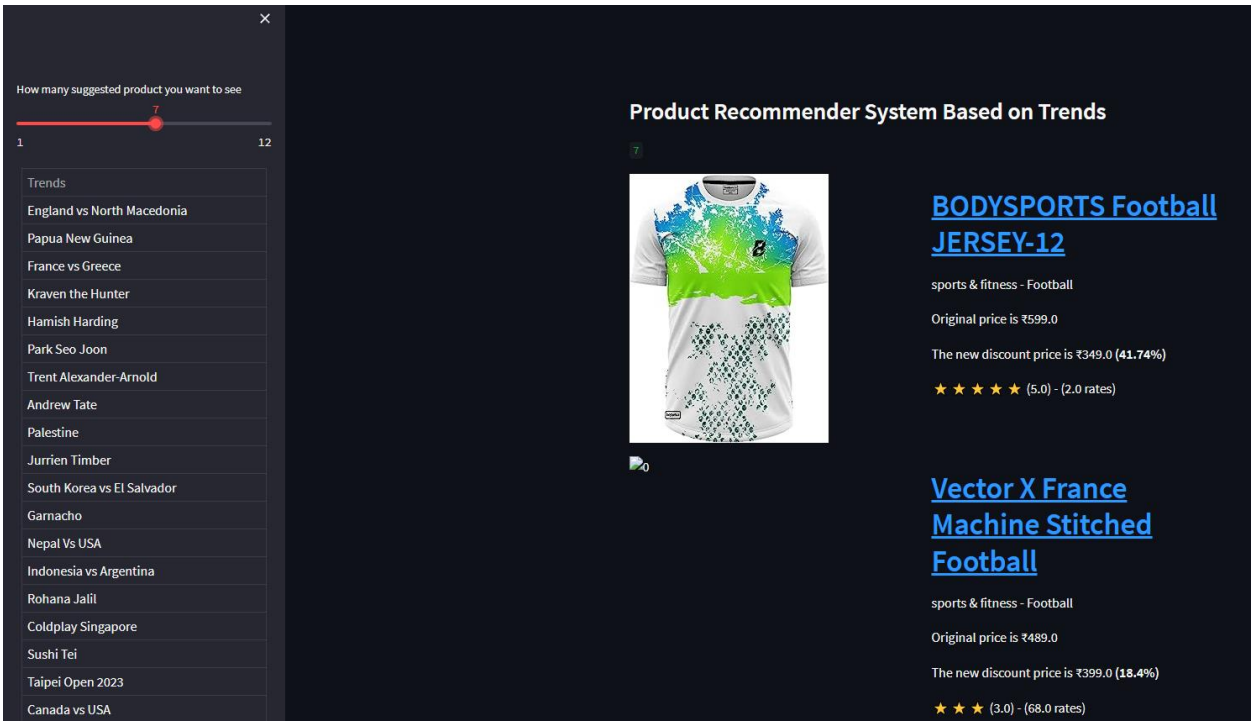


Figure 3.5: A screenshot of the Streamlit web application.

3.5 Comparison of MySQL and MongoDB as Data Source

MySQL and MongoDB are both popular and widely used database systems, but they have fundamental differences in their data models and usage scenarios. MySQL is a relational database management system (RDBMS) that follows a structured, table-based data model with predefined schemas. It is highly suitable for applications that require complex transactions, strict data integrity, and adherence to the relational data model. MySQL supports SQL (Structured Query Language) for querying and manipulating data.

On the other hand, MongoDB is a NoSQL database that utilizes a flexible, document-based data model. It allows for the storage of data in JSON-like documents, without enforcing strict schemas. This flexibility enables developers to work with dynamic and nested data structures, making MongoDB ideal for handling unstructured or semi-structured data. It also offers features like sharding and replication, which enable horizontal scalability across clusters of machines.

In terms of scalability, MySQL is known for vertical scalability, meaning it can handle increasing loads by upgrading hardware resources such as CPU, memory, and storage. It is well-suited for applications with a fixed amount of data and moderate workloads. However, MySQL has limitations when it comes to scaling horizontally across multiple machines.

On the other hand, MongoDB is designed for horizontal scalability, allowing data to be distributed and stored across multiple servers or clusters. It can handle high volumes of data and support large-scale, distributed applications. MongoDB's sharding capabilities allow for the distribution of data across shards, providing excellent scalability for write-intensive workloads.

Choosing between MySQL and MongoDB as a data source depends on the specific requirements of the application. If a well-defined schema and require complex transactions is needed, MySQL may be the better choice. However, if unstructured or rapidly evolving data is needed, and requires horizontal scalability and flexibility, MongoDB might be the more suitable option. In this project, MySQL is stored a well-defined product data, while MongoDB is used to store current trending item which is rapidly evolving data.

4 Conclusions and Future Work

This project has developed a product recommendation system with a full data pipeline from data sources to data warehouse. The project has been split into five parts, consisting of data source, data integration, data warehouse, analytics environment, and visualization & deployment. MySQL and MongoDB are used to store product data and trending items to simulate the data source in the data pipeline. Apache Nifi is used to integrate data from multiple sources into data warehouses. Moreover, Hive is used as data warehouses in this project, running on top of Apache Hadoop. In this project, JupyterLab is used as the coding or analytics environment with python to write the pipeline to generate product recommendation based on current trending item. Lastly, Streamlit is used to visualize the recommended product and present it as a web application.

In consideration of future work, there are several enhancements that can be made to improve the system. Firstly, the result can be included on the homepage of ecommerce websites instead of Streamlit as Streamlit is only for demonstration purposes only. Additionally, instead of using Amazon products, it can use real ecommerce website products as some of the products in the Amazon dataset are unavailable anymore. This can prevent the product image being empty during the demonstration.

5 References

Knotzer, Nicolas, "Product Recommendations in E-Commerce Retailing Applications". https://library.oapen.org/bitstream/handle/20.500.12657/26821/1/1003224.pdf

Lee (2014), "The Impact of Recommender Systems on Consumers: Study of Sales Volume and Diversity". https://riskcenter.wharton.upenn.edu/wp-content/uploads/2014/07/Lee.pdf

Lee, D. & Hosanagar, K., 2017, "How Do Recommender Systems Affect Sales Diversity? A Cross-Category Investigation via Randomized Field Experiment". https://repository.upenn.edu/cgi/viewcontent.cgi?article=1341&context=marketing_papers

Rai, B. K. (2023). IOT based humidity and temperature control system for Smart Warehouse. *Gazi University Journal of Science*, *36*(1), 173–188. https://doi.org/10.35378/gujs.993959

Sarangpure, N., Dhamde, V., Roge, A., Doye, J., Patle, S., & Tamboli, S. (2023). Automating the machine learning process using pycaret and Streamlit. *2023 2nd International Conference for Innovation in Technology (INOCON)*. https://doi.org/10.1109/inocon57975.2023.10101357

OpenAI API. OpenAI Platform. (n.d.). https://platform.openai.com/docs/introduction/overview

Paiva, S. (2013). A fuzzy algorithm for optimizing semantic documental searches. *Procedia Technology*, *9*, 1–10. https://doi.org/10.1016/j.protcy.2013.12.001

V. I. Levenshtein. (1965). Binary codes with correction of dropouts, insertions and substitutions of symbols. *Dokl. Academy of Sciences of the USSR*, *163*(4), 845–848.

Parab, L. (2023, March 26). *Amazon Products Sales Dataset 2023*. Kaggle. https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset?select=All%2BElectronics.csv

OpenAI. (2023). *Chatgpt*. ChatGPT. https://openai.com/chatgpt