College of Science and Engineering's BS in Data Science

DS299 Final Capstone Project Report

Final Capstone Project Report

Designing
Recommendation Systems for Smartphones Using Algorithmic Approaches

Author: Elen Petrosyan

Coordinator: Habet Madoyan

Supervisor: Arman Asryan

Spring 2024

College of Science and Engineering's BS in Data Science

DS299 Final Capstone Project Report

Abstract

As mobile technology continues to grow and advance in Armenia, people are often overwhelmed by the never ending array of options, showcasing the need for effective recommendation systems to simplify the decision-making process. Among the most prominent stores, Mobile Centre stands out as the biggest most recognized retailer throughout the country. Mobile Centre provides opportunities for customers seeking the latest smartphones and mobile accessories. Nevertheless, given the variety of products available, helping consumers find the best smartphones that meets their expectations and needs still is a big challenge. This paper is to suggest ways and implement algorithmic approaches for the recommendation process at Mobile Centre through advanced recommendation engines. Using approaches, these engines will guide customers toward more accurate, need-based recommendations. Some metrics will match smartphones based on similar features, while others will group devices into meaningful clusters, making it easier for customers to explore relevant personal preferences. Moreover, by implementing the suggested recommendation engines further, Mobile Centre will significantly enhance its recommendation process, giving a seamless customer experience. This approach will lead to improved customer satisfaction and loyalty, increasing Mobile Centre’s status as Armenia's top smartphone retailer and also overall improving customer satisfaction with their choices around Armenia.

College of Science and Engineering's BS in Data Science

DS299 Final Capstone Project Report

College of Science and Engineering's BS in Data Science

DS299 Final Capstone Project Report

Introduction

In nowadays digital era, there is an abundance of product choice preferences in any sphere in the market, especially the world of technology. This phenomenon makes people get lost in the abundance of products out there. It is crucial to create a helping tool and design recommender systems to make lives of people easier and increase user satisfaction. Recommender systems have become essential tools that have had a seamless integration into our virtual interactions and daily experiences. Shopping decisions. Entertainment. Selection of movies. Discovering music. Travel guides. You name it… Now imagine discovering the full potential of your smartphone experience with a recommender system that is built based on the specific platform, and knows precisely which product will fit the best for you before you even do. That is exactly what the ambition of our Capstone Project is: Designing Recommendation Systems for Smartphones using algorithmic approaches. This project has a goal to design different recommendation engines for smartphones using various algorithms. We have focused on data from a store based in Armenia, helping it give customers personalised decision-making opportunities and serve as guides as they make purchases.

Our approaches include collaborative filtering via one recommendation engine based on cosine similarity to match products by comparing key features, another using hierarchical clustering to categorise similar smartphones into clusters, and collaborative filtering via a K-Nearest Neighbors KNN algorithm, which will analyse the patterns of users ratings to make smartphones suggestions.

Introduction to Data

We wanted to build recommendation systems for Mobile Centre, the largest mobile phone retailer in Armenia, from scratch without requesting data from any company. We decided to scrape the data from the website using coding. Instead of implementing pre prepared data, which would possibly limit our project's workload, scope and control, we decided to gather the data independently. Thus, using this approach assured that our project was built from the bottom to the top. This approach allowed us to tailor the data extraction to specific needs and better understand every detail of the data we would use in our recommendation engines.

To do this, we used Python and specifically the ‘BeautifulSoup’ library, which is considered a powerful tool for Web Scraping. This allowed us to extract data directly from Mobile Centre's website. Our data collection process began with accessing the main product listing page at Mobile Centre and by sending a ‘get’ request with appropriate headers to mimic the browser user, ensuring that our scraper was treated like a normal user.

From the main listing, we have successfully extracted individual product links for each smartphone using the appropriate HTML classes associated with the products. Each smartphone link was then accessed to scrape detailed information about each device and to gather detailed information about each smartphone. The main data attributes we collected include brand data, rating, price, year, and further features.

Finally, after completing the data collection, we structured the already existing data and saved it into a CSV file naming it ‘mobilecenter_data.csv’. Thus, this file serves as the basement for our further analysis, giving us access and ability to work with the data as we develop and work on our recommendation engines.

	Name	Price	Rating
0	Realme Note 50 4GB/128GB (Sky Blue)	39,900դր.	4
1	Realme Note 50 4GB/128GB (Midnight Black)	39,900դր.	4
2	Realme Note 50 3GB/64GB (Sky Blue)	33,900դր.	4
3	Realme Note 50 3GB/64GB (Midnight Black)	33,900դր.	4
4	Realme C67 8GB/256GB (Sunny Oasis)	81,900դր.	3
...	...	...	...
425	NOKIA 230 (Black)	32,900դր.	4
426	NOKIA 230 (Silver)	32,900դր.	4
427	NOKIA 3310 (Dark Blue)	30,900դր.	3
428	NOKIA 3310 (Grey)	30,900դր.	4
429	NOKIA 3310 (Red)	30,900դր.	4

Literature Review

For a more detailed and thorough understanding of content-based recommendation systems, we also reviewed additional literature and previous implementation. According to Analytics Vidhya, content-based systems utilise item features such as categories, brands, or user-specific metadata to make recommendations. As we also have user data, we opted for content-based recommendation systems. These systems perform particularly well when detailed descriptive data about the items is available. They primarily recommend items by understanding the content of the products relative to the preferences of users.

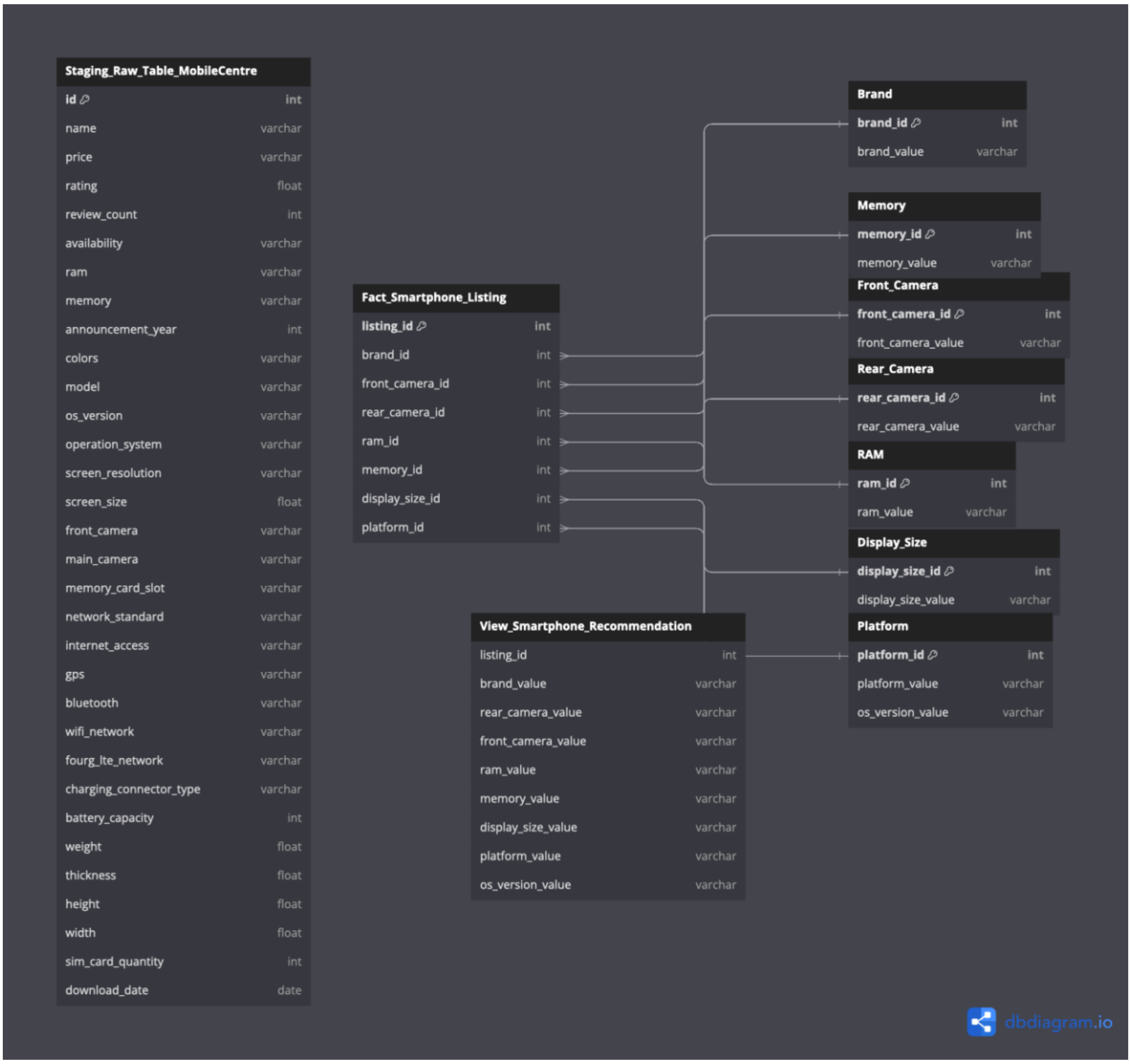
The methodology includes creating feature vectors from item attributes, which can then be used to calculate similarities between items. For example, if a user likes a particular smartphone model due to particular features like camera quality or battery life, the system will recommend other smartphones with similar specifications. This approach is ideal for situations where user data is sparse or when new items are added to the catalogue, as it doesn't rely solely on user-item interactions (Analytics Vidhya).

Furthermore, GraphAware discusses the integration of graph databases in building sophisticated content-based recommender systems. By modelling both items and their features as nodes in a graph, it is possible to create a dynamic and scalable recommendation engine. This approach allows for rich, interconnected data representations that can enhance the accuracy of recommendations through deeper insights into the relationships between different data entities (GraphAware).

Applied Tools

DB Diagram

We have used dbdiagram.io tool - an extensive tool to visualize and structure our database architecture effectively. This tool has proved itself for visually designing our database schema, enabling us to better organise our data and make tailored decisions for the upcoming steps and how to effectively make use of our data in the development process. By providing a clear visual representation, dbdiagram.io has helped us to have a better understanding of the relational structures and dependencies, which is a key in scaling the project and further development and enhancing its functionality. We have structured our database into various tables to handle different aspects of the data we collected from the Mobile Centre.



Our primary table, Staging_Raw_Table_MobileCentre, acts as the initial repository for all raw data scraped from the website. This table is designed to hold comprehensive details about each smartphone, including technical specifications and descriptive data such as name, price, and availability. Key identifiers and characteristics such as RAM, memory, camera specifications, and more are also included, which are important for our recommendation systems.

We have then created dimension tables for attributes like Brand, Rear_Camera, Front_Camera, RAM, Memory, Display_Size, and Platform. These tables store unique identifiers and descriptive values for their columns and with this approach we decrease redundancy, while increasing data integrity, which is essential for real-time recommendation systems.

The data from our staging table is processed and transformed into these dimension tables, from where it is then inserted into a Fact_Smartphone_Listing table. This fact table links all dimension tables using foreign keys, which not only preserves the relational integrity but also optimises the data retrieving process for the future recommendation engines.

As for the final step, we have visualised a View_Smartphone_Recommendation table. This view collects and gathers data from the fact table and all related dimension tables, presenting a comprehensive perspective of smartphone options based on user preferences. This setup ensures that our project can scale efficiently as the dataset grows and as the recommendation algorithms become more complex.

As we look to enhance and scale our project, the clarity provided by this tool will be pivotal in integrating more features and handling larger datasets, ultimately leading to a more dynamic and responsive recommendation system.

PostgreSQL, DataGrip, Visual Studio Code

To ensure our recommendation system can scale effectively further we decided to implement PostgreSQL as our primary database management system. With its powerful capabilities for dealing with complex queries and managing large datasets, PostgreSQL was a leading choice for our project. To facilitate our schema creation, data management, and SQL queries, we relied on DataGrip, an IDE. This tool provided a user-friendly environment where we could easily design our database schema, write queries, and visualise data relationships, making it essential to maintaining data integrity and smooth project development.

Visual Studio Code served as our primary Python development environment, allowing us to write and execute Python scripts seamlessly. By using Python's set of libraries, we could process data, manipulate it as necessary, and perform the relevant transformations before pushing it into our PostgreSQL database. In our Python script, we provided the connection credentials by setting the server name, username, password, port, and database name with the PostgreSql. Additionally, our Python scripts were designed to fetch data directly from the Mobile Centre website, cleanse and normalise it, then upload it into the tables in our PostgreSQL database accordingly. We tried to ensure that data could be pushed to the database consistently and accurately. Once the data was in PostgreSQL, we could easily fetch it using Python and SQL queries. This was an important step for the data retrieval and crucial for providing the recommendation engines, which required efficient access to relevant smartphone features.

The integration of PostgreSQL, DataGrip, and Visual Studio Code gave a scalable architecture for our project. This setup ensures that our recommendation system will continue to be efficient and adaptable, even as the data volume increases, while also allowing for quick iteration and future feature development.

Recommendation Engines

For my capstone project, we developed three distinct recommendation engines to enhance the purchasing experience at Mobile Centre by providing tailored smartphone suggestions. Each recommender engine utilises a unique algorithmic approach, ensuring robustness in our recommendations. Each recommendation approach is presented below with an explanation of their effectiveness.

Cosine Similarity Engine Recommendation Engines

First approach is cosine similarity. This recommender engine analyses the features of smartphones as vectors in a multidimensional space, by calculating the cosine of the angle between these vectors, it then determines how similar two smartphones are. The smaller the angle, the higher the similarity. This method is particularly effective for identifying products with similar specifications in a straightforward, mathematical manner. It's excellent for customers who are interested in finding phones with closely matching features to a phone they already like.

```
Top 5 similar products to 'Samsung Galaxy A55 256GB (Yellow)':
Samsung Galaxy A55 256GB (Yellow): Similarity Score = 1.0000000000000002
Samsung Galaxy A55 256GB (Blue Black): Similarity Score = 0.5610193985140073
Samsung Galaxy A55 256GB (Blue Black): Similarity Score = 0.5610193985140073
Samsung Galaxy A55 256GB (Light Blue): Similarity Score = 0.5440721290671786
Samsung Galaxy A55 256GB (Light Blue): Similarity Score = 0.5440721290671786
                                             + Code      + Markdown
```

Based on cosine similarity recommendation engine

K-Nearest Neighbors (KNN) Approach

The next recommender engine is KNN engine which classifies smartphones based on a similarity metric, typically Euclidean distance between feature points. It looks at the 'k' closest neighbours to a query phone and suggests those as the top recommendations. This method is well adapted to diverse datasets and can effectively handle a variety of product features, making it suitable for a dynamic inventory like that of Mobile Centre.

```
Top 5 recommendations similar to 'Samsung Galaxy A55 256GB (Yellow)':
Realme Note 50 4GB/128GB (Sky Blue)
Realme Note 50 4GB/128GB (Midnight Black)
Realme Note 50 3GB/64GB (Sky Blue)
Realme Note 50 3GB/64GB (Midnight Black)
Realme C67 8GB/256GB (Black Rock)
```

Based on KNN recommendation engine

Hierarchical Clustering

The last, but not least, recommendation engine is hierarchical clustering approach. This method groups smartphones into clusters based on their attributes without needing predefined categories. Thus, phones within the same cluster are more similar to each other than to those in other clusters. This approach is useful for exploring different phone types and discovering new models that might not be immediately obvious from specifications alone.

```
Top 5 recommendations similar to 'Samsung Galaxy A55 256GB (Yellow)':
Samsung Galaxy A55 256GB (Light Blue)
Samsung Galaxy A55 256GB (Yellow)
Samsung Galaxy A55 256GB (Light Blue)
Samsung Galaxy A55 256GB (Yellow)
Samsung Galaxy A55 256GB (Blue Black)
```

Based on hierarchical clustering recommendation engine

The selection process of these three recommendation engines allows us to see different customer preferences and purchasing patterns while offering deeper insights into the varying recommendation methods. When it comes to comparing the recommendations of top 5 recommended smartphones, we could notice that Cosine Similarity and KNN frequently suggested similar devices due to their more stringent matching techniques. On the other hand, Hierarchical Clustering provided greater diversity, suggesting options that were relevant but at the same time distinct from the input model.

In conclusion, these engines collectively deliver an enriched recommendation system that enhances user satisfaction by offering multiple perspectives on potential smartphone choices. The system not only streamlines decision-making but also adapts to various customer needs, in its case improving the shopping experience at Mobile Centre.

College of Science and Engineering's BS in Data Science

DS299 Final Capstone Project Report