MDSAA

Master Degree Program in

Data Science and Advanced Analytics

Big Data Analytics

Recommender System for E-Commerce Retail

Daniel Caridade, number: 20211588
Gonçalo Teles, number: 20211684
Gonçalo Peres, number: 20211625
Guilherme Godinho, number: 20211552
Vinicius Pinto, number: 20211682

Group 04

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

June, 2025

1. DEVELOPMENTS SINCE THE DEFENCE

Since the project defence, Group 04 has made significant enhancements to extend and strengthen the original work. One of the main areas of improvement was in the modelling and pre-processing phase in which the data cleaning and preprocessing was tested, by implementing the recommender system with and without some of the steps to quantify its impact in the relevance of recommendations made evaluated based on the similarity scores.

Another major advancement was in the streaming component. Initially, the prototype simulated real-time interactions by using the last known transaction of a customer and pushing it into a stream, essentially mimicking micro-batches. This was replaced by a more realistic and interactive simulation. The new setup allows a user to enter the system by "logging in" with their customer ID, receiving recommendations based on their profile and current cart status, and then actively deciding what products to add—whether recommended or not. They can explore the catalogue, search by name, and proceed to check out or abandon the session, simulating a lead. This redesign creates a much more lifelike flow, turning the stream into a dynamic customer journey that better aligns with a real-world E-commerce behaviour.

Finally, the introduction of GraphFrames added a valuable new dimension to the analysis. By modelling the dataset as a product co-purchase network, it uncovered deeper relationships beyond simple similarity or frequency counts. Using PageRank and NetworkX, this approach identifies not just the most popular items, but those that serve as key anchors in the buying process. These insights provide meaningful guidance for marketing strategies, helping the E-commerce platform focus on products that influence purchasing behaviour and drive sales.

2. Introduction & Background

2.1. Problem Statement

In an increasingly digital world, businesses are constantly adapting to meet evolving consumer expectations. The COVID-19 pandemic acted as a catalyst for a massive shift toward E-Commerce, introducing millions of people to the convenience of purchasing goods from the comfort of their homes. This transformation changed the way customers interact with brands, compelling businesses to rethink their strategies in order to stay relevant and competitive.

This shift has led to an explosion in the volume, variety, and velocity of data being generated—from website clicks and search queries to purchase histories and customer reviews. While this Big Data holds immense potential, it also presents complex challenges: how can businesses effectively process and analyse such large-scale, unstructured information to uncover actionable insights?

To address this, our team has developed a Big Data-powered recommender system aimed at enhancing the online shopping experience. By analysing patterns in customer behaviour and preferences, the system delivers personalized product suggestions that not only simplify purchasing decisions but also foster a deeper connection between the customer and the brand. This tool was developed to enhance the customer journey by helping users find desired products more quickly and avoid forgetting items they may need, thereby improving satisfaction, fostering loyalty, and increasing revenue. It empowers businesses to deliver more personalized promotions while offering customers the intuitive, efficient shopping experience they expect in today's digital landscape.

2.2. Research Motivation

The primary motivation behind this research lies in the growing prominence of customer-centric strategies within the online retail sector. As the volume and variety of products available online continue to expand, consumers increasingly expect fast, seamless, and personalized shopping experiences. This complexity presents a valuable opportunity for businesses to adopt intelligent systems that simplify decision-making and enhance user satisfaction.

Our group was particularly drawn to recommender systems because they offer a practical and scalable way to anticipate customer needs—helping users find relevant products more efficiently while fostering a sense of personalization and trust. To ensure our work was grounded in real-world applicability, we analysed transaction data from an E-commerce platform using Big Data tools such as Apache Spark and Databricks. This project not only allowed us to apply theoretical knowledge in a business-oriented context, but also highlighted the vital role data analytics plays in shaping strategic decisions and driving value in modern online retail.

3. Data Collection & Preprocessing

3.1. Data Sources

The dataset used in this project was obtained from a London-based online retail store and is publicly available on Kaggle *(Kaggle, n.d.)*. This store has been selling gifts and homewares for adults and children through its website since 2007 and made publicly available the transaction records from December 2018 to December 2019 for academic and analytical use by the broader data science community. For this project, the data was loaded into a Databricks environment and processed using Apache Spark, an open-source analytics engine designed for large-scale data processing.

3.2. Data Characteristics

The dataset was provided in a structured tabular format, which facilitated seamless integration into the Databricks environment. It contains 535,350 rows and 8 features, each representing key attributes of individual transactions. A summary of these features is presented in Appendix 1.

An initial exploratory data analysis revealed several insights that guided the development of our recommender system. Notably, while **the majority of the company's customers are based in the UK (91%)**, the **highest spenders** are located **in the Netherlands (£238,645.71)** and **Ireland (£127,741.91)**, as shown in Appendix 2. Additionally, Popcorn Holder was identified as the most purchased item, appearing in numerous transactions—making it a strong candidate for product recommendations. Lastly, sales peaked in November 2018, driven by a successful period of customer acquisition.

3.3. Data Cleaning and Preprocessing

To make our recommender system useful and relevant, we first had to solve several issues in the raw dataset. The goal was to make sure the patterns uncovered actually represented how customers behave. To reach that point, 5 steps were performed, namely:

1. **Remove duplicated rows**: Some transactions listed the same product twice with the same quantity. Since we already have a *Quantity* feature, this was clearly an error that would have skewed purchase patterns.
2. **Filtered out cancelled transactions**: Cancelled transactions were removed, as including them reduced the relevance of the product recommendations by lowering similarity values.
3. **Remove leads in the dataset:** Some transactions lacked a customer ID and were removed, as most of them were also cancelled, mostly representing leads on the data.
4. **Transforming variables into integers for fast processing:** *TransactionNo* and *CustomerNo* were originally stored as strings and were converted into integers to improve processing efficiency, as integers are processed faster than strings.
5. **Standardizing names of countries:** Outdated country names like 'EIRE' (Ireland) and 'RSA' (South Africa) were standardized to simplify analysis and better understand customer origins.

What sets this apart from a typical cleaning pipeline is how we tuned the process around recommendation relevance. Instead of blindly keeping every record, we focused on what helps the system *actually recommend better*—and dropped what didn't.

Once data was cleaned, our team moved into feature engineering for clustering. These features (listed in Appendix 3) were crafted to reflect customer purchasing behaviour, and then three preprocessing stages were applied using a pipeline to ensure consistent, reusable, and scalable transformations—essential in distributed systems like Spark.

1. **Vector Assembler:** Used to combine features into a single vector, as Spark algorithms require vectorized input.
2. **Robust Scaler:** Traditional scaling methods are sensitive to outliers, which are common in sales data. To address this, Robust Scaler was applied, which uses medians and interquartile ranges for a more reliable and realistic scaling.
3. **Normalization**: To make clustering more meaningful, we applied Databricks' Normalizer to ensure each feature vector had unit norm. Testing showed that this approach outperformed unnormalized data by allowing algorithms like K-means to focus on customers' purchasing patterns rather than the scale of their spending, resulting in more insightful groupings.

This cleaning and preprocessing approach builds on previous work with this dataset and in Big Data Analytics by combining scalable, distributed processing with robust feature transformations to handle large, complex transactional data. Uniquely, each preprocessing step was iteratively tested to ensure it contributed to improving the recommender system's performance, creating a feedback loop that maximizes recommendation relevance for the customers.

4. METHODOLOGY & TOOLS

4.1. MACHINE LEARNING TECHNIQUES

Our recommender system relied on two key machine learning techniques: customer segmentation and product co-purchase modelling. For customer segmentation, the relevance of features (in Appendix 3) were evaluated using Spearman correlation and Mutual Information—an approach that goes beyond simple feature selection by capturing both linear and non-linear relationships, which is crucial in complex Big Data environments. This allowed us to exclude redundant features and focus on the most informative ones: *Monetary, AverageBasketSize, Frequency, AverageDaysBetweenTransactions, and RepeatedPurchaseRatio*.

Next, the optimal number of clusters was determined using the silhouette score and elbow methods, with four clusters selected for K-Means. K-Means was selected for its native Spark implementation, enabling scalable and distributed processing ideal for Big Data. In contrast, other methods required manual implementation via pandas Data Frames, which is inefficient and impractical for large-scale, distributed environments. The resulting clusters revealed distinct customer behaviours:

- **Segment 0: New Routine Clients** — These customers make frequent purchases of similar products but with smaller baskets and lower overall spending. Their consistent buying and shorter time between transactions suggest they are new but quickly forming loyal habits.
- **Segment 1: Premium Novelty Seekers** — This group buys less frequently but focuses on high-value, often premium products. Despite buying moderate basket sizes, their behaviour shows a willingness to explore and spend on exclusive items, making them key for targeted premium promotions.
- **Segment 2: Bargain Enthusiasts** — These customers shop frequently with full baskets, but mostly choose lower-priced products. Their diverse purchases show they seek deals and value, making them a price-sensitive segment that responds well to discounts and drives volume through frequent transactions.

- **Segment 3: Uninterested Clients** — This segment includes infrequent shoppers making small, low-value purchases. Though they form a large part of the customer base, their engagement is limited, highlighting opportunities to reactivate or better serve them.

Visualizing these segments with t-SNE (Appendix 4) showed clear boundaries between groups, while they were not very far apart from each other, confirming that our clustering effectively captures meaningful and actionable customer distinctions.

The product co-purchase modelling enhances the recommender system by guiding customers throughout their shopping journey. As items are added to the cart, the system suggests complementary products based on past purchases and popular combinations, using **Jaccard similarity**—a method that compares overlaps in purchase history across users. At checkout, it shifts to a “Did You Forget?” recommender, recommending products based on recent transactions and similar customers, encouraging larger baskets without overwhelming the shopper.

What sets this approach apart is its seamless integration with customer behaviour and clustering insights. By combining real-time cart analysis with cluster-based product similarities, we deliver smarter, more context-aware recommendations that feel natural and relevant. This dynamic interplay between shopping actions and data-driven suggestions is especially key in Big Data environments, where balancing speed, personalization, and scalability is challenging but critical for effective recommendation systems.

4.2. Querying & Storage

All data processing and storage was managed using Databricks File System (DBFS) on the Community Edition, which allows only one active cluster at a time. Despite this limitation, DBFS provided an environment for handling large transactional datasets, which integrated with Apache Spark, enabling fast querying and distributed computing to meet the high-performance demands of the recommender system.

4.3. Visualization

To support analysis and interpretation, various visualizations were created using both Spark, Pandas and Plotly. While Spark was used for summaries of the data, Pandas and Plotly were used for more detailed, interactive plots once data was aggregated. Additionally, NetworkX was employed to visualize the product co-purchase network derived from GraphFrames and PageRank, helping to highlight top influential products. These visual tools played a key role in making complex patterns more intuitive and actionable.

4.4. Streaming

To create a more dynamic and interactive shopping experience, a real-time streaming setup was implemented using Databricks, to simulate how the recommender system works when a customer enters the website to make a purchase. As users interact with the system—adding products to their cart—those actions are instantly captured and written to a Delta table using the Databricks File System (DBFS). This stream of cart events allows the recommender system to immediately respond, updating product suggestions based on the latest activity.

What makes this approach stand out is its interactivity. Instead of relying on static data or batch processing like many Python-based solutions on other works, this setup responds to user actions as they happen. When a product is added, the system instantly recalculates suggestions using Spark, helping users discover complementary items at the moment. This kind of *live* intelligence adds a level of personalization and fluidity that traditional recommendation systems often lack. By simulating the customer journey with real-time inputs and integrating it into the recommendation engine, the experience becomes more natural and engaging—bringing machine learning closer to how real shopping behaviour unfolds. Examples of this can be seen in Appendix 5 and 6.

4.5. Graph Frames

To uncover deeper relationships between products, a product co-purchase network was built using **GraphFrames** on Databricks, where each product is a node and edges connect products bought together in the same transaction. Utilizing **PageRank**, the most influential products—those frequently bought alongside many others—were identified, with key rings standing out as the most influential products, especially those representing initials like **Z, Y, and V**. This suggests strong personalization patterns—likely buyers choosing initials for themselves or as small gifts. These insights make key rings ideal for homepage features, checkout prompts, or personalized bundles.

This graph-based approach goes beyond previous works that often stop at similarity scores. By modelling the product ecosystem as a network, it becomes easier to identify not just what's popular—but what's **central** to buying behaviour. These insights help surface bundling opportunities, such as pairing lower-ranking items like Tea related products with more influential products, to boost purchases of those products and manage better stocks.

5. Conclusion

This project showcases the value of combining multiple analytics techniques—segmentation, co-purchase modelling, and feature selection—within a scalable Big Data framework. Using Apache Spark on Databricks, we built a system capable of delivering real-time, personalized recommendations for E-Commerce, while maintaining computation performance and scalability across large datasets by utilizing a distributed computing solution.

During the development of this solution, several challenges were addressed. First, the dataset had very few purchases for each customer making methods like smart baskets unviable, leading us to adopt co-purchase modelling enhanced by customer clustering. Secondly, while Databricks Community limitations are well known, they posed an extra challenge for streaming. Our initial plan to build a web interface wasn't feasible, so we developed an interactive in-platform solution instead, preserving distributed performance. Additionally, extracting meaningful insights from product relationships was difficult when relying only on co-occurrence. GraphFrames with PageRank proved essential, revealing influential items that basic counts could not surface.

Ultimately, the project delivers a practical and extensible approach for recommendation systems in Big Data environments, illustrating how Spark and Databricks can orchestrate complex pipelines that are scalable, interactive, and insight-driven.

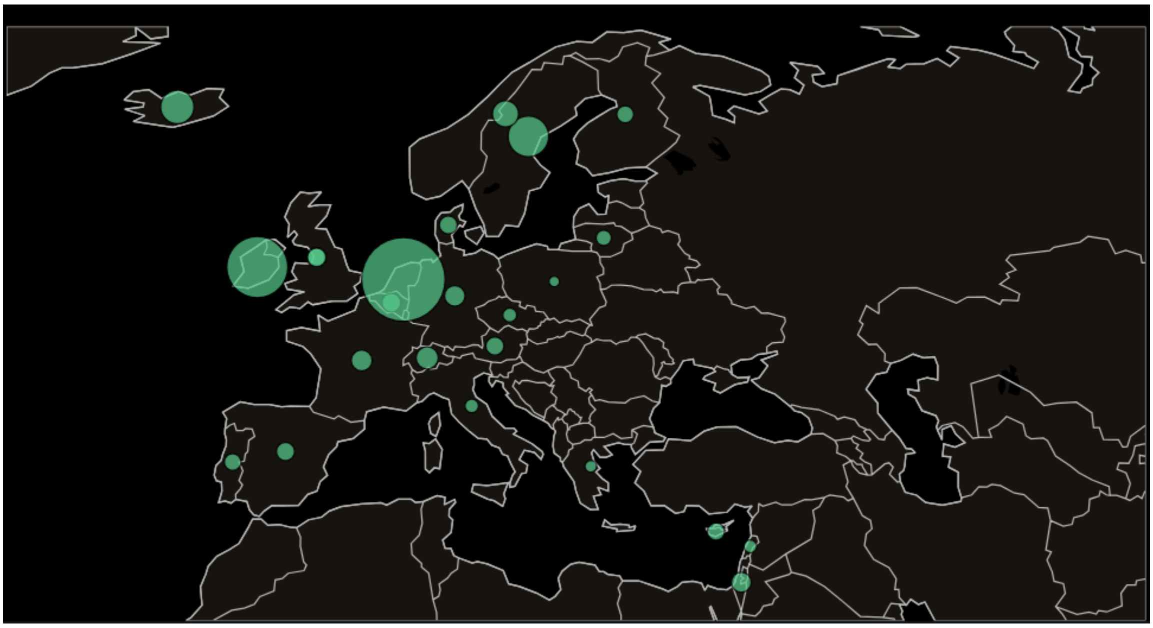
6. References

Kaggle (n.d.). Retrieved 4 June 2025, from https://www.kaggle.com/datasets/gabrielramos87/an-online-shop-business

7. Appendix

Appendix 1: Dataset Feature Overview

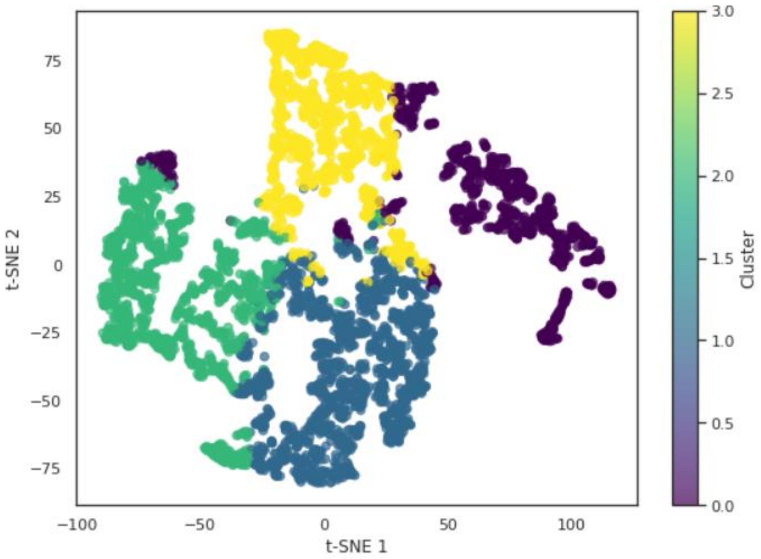
Feature	Description
TransactionNo	Unique identifier for each transaction.
Date	Date on which the transaction occurred.
ProductNo	Unique identifier for each product.
ProductName	Name of the purchased product.
Price	Price per unit of the product in British pounds (£).
Quantity	Number of units purchased (negative values indicate cancellations).
CustomerNo	Unique identifier for each customer.
Country	Country of residence of the customer.



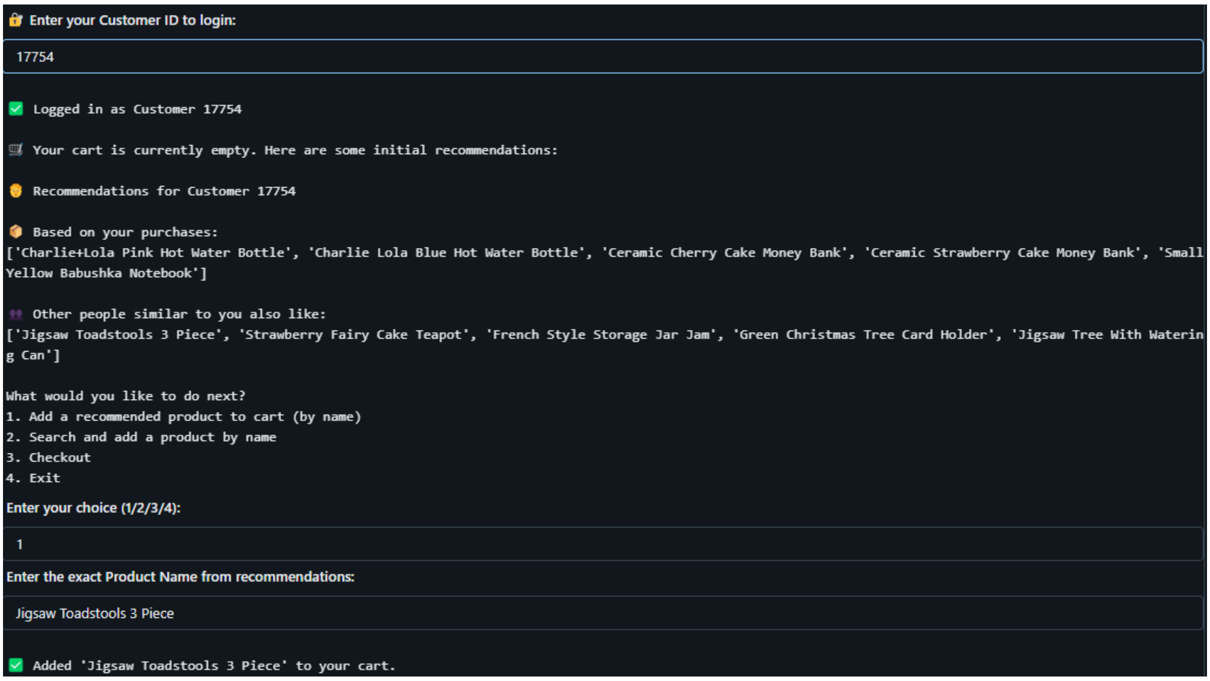
Appendix 2: Average Money Spent by Customer per country

Appendix 3: Features Created for Clustering

Feature	Description
DaysAsCustomer	Number of days between the customer's first and last purchase.
Recency	Days since the customer's most recent transaction. Lower values mean more recent activity.
Frequency	Total number of unique transactions made by the customer.
Monetary	Total money spent by the customer.
AverageDaysBetweenTransactions	Average number of days between transactions. Indicates engagement consistency.
RepeatedPurchaseRatio	Ratio of distinct transactions to distinct products. Higher values suggest repeat buying behaviour.
HHI (Herfindahl-Hirschman Index)	Indicates product purchase concentration. Higher values mean less variety.
AverageBasketSize	Average quantity of items per transaction.
AverageBasketValue	Average amount spent per transaction.
EngagementScore	Weighted metric combining frequency, monetary, recency, and consistency.

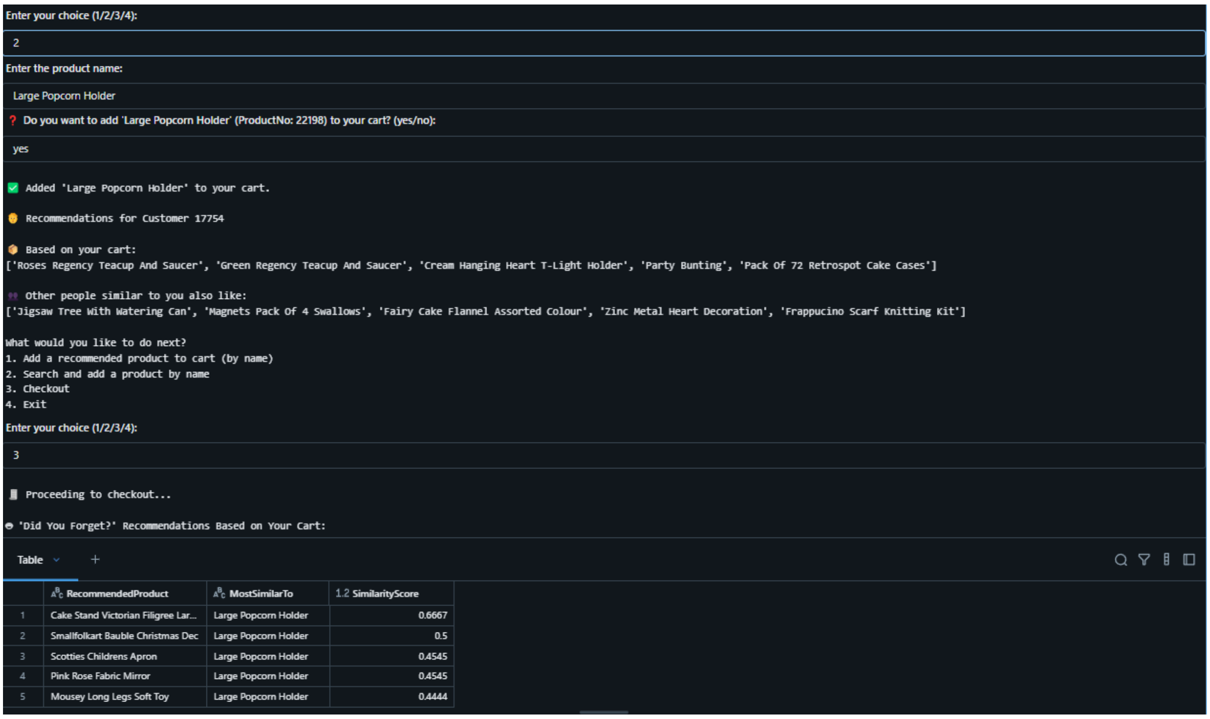


Appendix 4: 2D t-SNE Visualization of Customer Segments



```
🔐 Enter your Customer ID to login:
17754

✅ Logged in as Customer 17754

🛒 Your cart is currently empty. Here are some initial recommendations:

👤 Recommendations for Customer 17754

📦 Based on your purchases:
['Charlie+Lola Pink Hot Water Bottle', 'Charlie Lola Blue Hot Water Bottle', 'Ceramic Cherry Cake Money Bank', 'Ceramic Strawberry Cake Money Bank', 'Small
Yellow Babushka Notebook']

👥 Other people similar to you also like:
['Jigsaw Toadstools 3 Piece', 'Strawberry Fairy Cake Teapot', 'French Style Storage Jar Jam', 'Green Christmas Tree Card Holder', 'Jigsaw Tree With Waterin
g Can']

What would you like to do next?
1. Add a recommended product to cart (by name)
2. Search and add a product by name
3. Checkout
4. Exit
Enter your choice (1/2/3/4):
1
Enter the exact Product Name from recommendations:
Jigsaw Toadstools 3 Piece

✅ Added 'Jigsaw Toadstools 3 Piece' to your cart.
```

Appendix 5: Real-Time Streaming at Login with Initial Product Recommendations



```
Enter your choice (1/2/3/4):
2
Enter the product name:
Large Popcorn Holder
❓ Do you want to add 'Large Popcorn Holder' (ProductNo: 22198) to your cart? (yes/no):
yes

✅ Added 'Large Popcorn Holder' to your cart.

👤 Recommendations for Customer 17754

📦 Based on your cart:
['Roses Regency Teacup And Saucer', 'Green Regency Teacup And Saucer', 'Cream Hanging Heart T-Light Holder', 'Party Bunting', 'Pack Of 72 Retrospot Cake Cases']

👥 Other people similar to you also like:
['Jigsaw Tree With Watering Can', 'Magnets Pack Of 4 Swallows', 'Fairy Cake Flannel Assorted Colour', 'Zinc Metal Heart Decoration', 'Frappucino Scarf Knitting Kit']

What would you like to do next?
1. Add a recommended product to cart (by name)
2. Search and add a product by name
3. Checkout
4. Exit
Enter your choice (1/2/3/4):
3

🧾 Proceeding to checkout...

● 'Did You Forget?' Recommendations Based on Your Cart:
```

	RecommendedProduct	MostSimilarTo	SimilarityScore
1	Cake Stand Victorian Filigree Lar...	Large Popcorn Holder	0.6667
2	Smallfolkart Bauble Christmas Dec	Large Popcorn Holder	0.5
3	Scotties Childrens Apron	Large Popcorn Holder	0.4545
4	Pink Rose Fabric Mirror	Large Popcorn Holder	0.4545
5	Mousey Long Legs Soft Toy	Large Popcorn Holder	0.4444

Appendix 6: Real-Time Product Search, Cart Updates, and Checkout Suggestions