Group 2
AMS560/CSE542 Fall ‘23 Final Project
Grocery Shopping Optimization

Sai Swaroop Krishna Nellore, Sai Akhil Yerramsetty, Xander Barron

1. Introduction:

This report describes the development and functioning of an innovative retail application that aims to change traditional shopping into a more effective and customized experience. Our application provides a platform that blends convenience, customization, and choice in an effort to meet the changing needs of contemporary consumers.

1.1 Motivation for the Project:

Grocery shopping methods and options have become complex and confusing for some in an ever-changing world. Current options include traditional in-person shopping with aisles and checkouts, drive up pick up where employees must gather the requested groceries and deliver them to the customer's car, and delivery, where groceries are delivered directly to the customer's doorstop. The latter two options arose with the rise of contactless shopping as a result of the recent global pandemic, but have remained popular shopping methods. Is it true, however, that these methods are faster and less expensive for the consumer?

We hope to give the customer an informed option on how to buy their daily groceries at the best price and convenience because of the abundance of options available across stores. This is particularly significant because of how frequently consumers must buy these goods and how many options are available to them when doing so.

Finally, it is critical for retailers to forecast projected client density and purchasing habits. This has already been done, but only at an approximated level; there is no specific number, only past numbers and historical data. With our product, we can notify retailers about the exact

KPIs that our clients intend to use. This refers to the time they arrive/interact with the store, the number of things they intend to purchase, and how they intend to do it. This helps the store with any staffing or stocking concerns.

1.2 Project Goals

We set out to create a novel retail application to allow customers to choose their grocery purchase location and method in an informed manner, as well as giving them options to do so, streamlining the process.

In order to do this, we first must gather data on the stores, groceries and methods in question. From there an algorithm will give the customer options on how to best get their groceries based on their own personal preferences like time-cost priority, store preferences and method of purchase preferences.

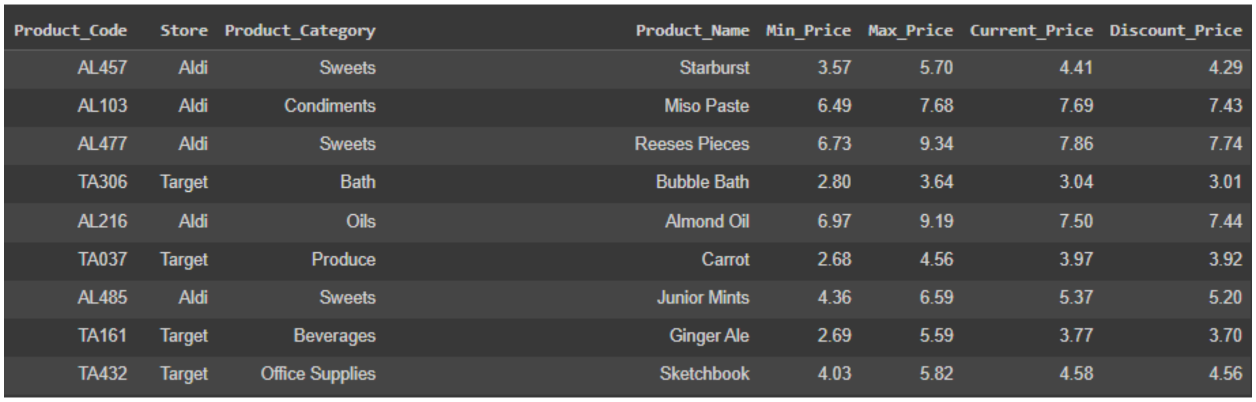
Lastly, in order to get retailers to buy into our model and give us updated and accurate information we will relay the customers' plans and preferences to the retailer so that they can plan accordingly.

2. Project Deliverables:

2.1 Grocery Data Points

For the foundation of our project, we utilized a pre-existing Australian grocery dataset encompassing data on various grocery items. We took a small sample from it and modified it to accommodate items from 3 different stores, namely, Walmart, Target, and Aldi. This dataset featured detailed information on 526 unique items from each store, meticulously categorized by both type and name. For example, the product "milk" lived in the product category of dairy along with "mozzarella cheese" and such. The categories included dairy, produce, meat, condiments, canned goods, beverages, frozen, oils, spices, health supplies, bath, baby care, flour, office supplies, sweets and grains and lentils. These are general weekly groceries that consumers would need when going grocery shopping. The frequency of purchasing some of these may be less than others, but all are still regularly bought.

The important data regarding the items were the prices of each item. For each item the data set had the minimum and maximum price that the product has ever been. Additionally, it had the current price and the price one would pay if they applied a discount to the item. This will be used to compare prices between stores and give consumers the best value price wise.



Product_Code	Store	Product_Category	Product_Name	Min_Price	Max_Price	Current_Price	Discount_Price
AL457	Aldi	Sweets	Starburst	3.57	5.70	4.41	4.29
AL103	Aldi	Condiments	Miso Paste	6.49	7.68	7.69	7.43
AL477	Aldi	Sweets	Reeses Pieces	6.73	9.34	7.86	7.74
TA306	Target	Bath	Bubble Bath	2.80	3.64	3.04	3.01
AL216	Aldi	Oils	Almond Oil	6.97	9.19	7.50	7.44
TA037	Target	Produce	Carrot	2.68	4.56	3.97	3.92
AL485	Aldi	Sweets	Junior Mints	4.36	6.59	5.37	5.20
TA161	Target	Beverages	Ginger Ale	2.69	5.59	3.77	3.70
TA432	Target	Office Supplies	Sketchbook	4.03	5.82	4.58	4.56

When it comes to providing consumers with accurate time estimates for acquiring food based on their chosen purchasing method, real-time prediction poses a challenge. Consequently, we resorted to establishing consistent, predetermined values for each shopping mode and retailer, along with fuel costs. This estimation process, though manual, was executed with realistic setting in mind. These values are subsequently integrated into our recommendation algorithm, further details of which are elaborated on in the following sections.

2.2 Algorithm and Recommendations

The heart of our smart grocery system lies in the recommendation algorithm. Below is an overview of the recommendation system:

2.2.1 Time and Cost Factors

- Time-related factors, such as aisle transition time, checkout time per item, and delivery time factors, are considered to optimize the shopping process.
- Cost-related factors, including pickup and delivery cost factors, mileage per gallon, and gallon cost, are taken into account.

2.2.2 Distance and Time Calculation

Google Maps API is used to calculate the distance and time from the customer's location to each store. This information is crucial for optimizing the delivery process.

2.2.3 Cart Price and Costs Calculation

The algorithm computes the prices for items in the user's cart from different stores, considering discounts. It also generates data structures to store cart details.

2.2.4 Mode of Purchase Calculation

Different modes of purchase (in-store, pickup, delivery) are evaluated based on individual cart prices and distance travel costs.

2.2.5 Time Calculation

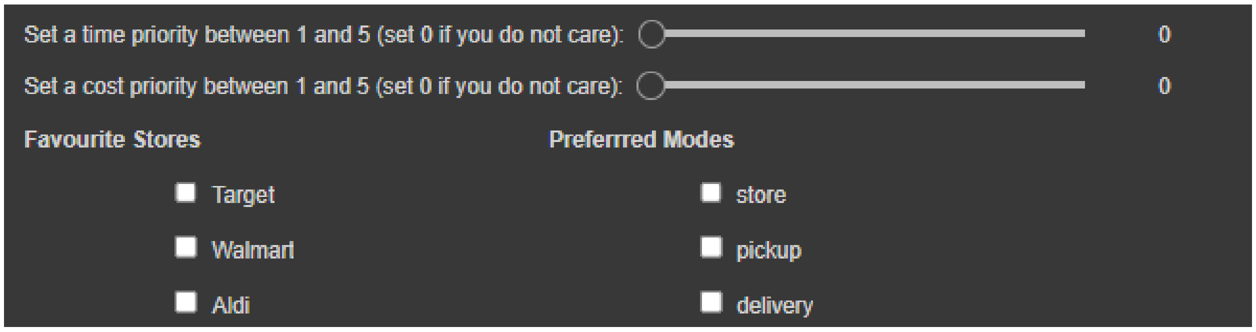
The algorithm calculates the time required for shopping in each store and for each mode, considering factors such as aisle transition time, checkout time, and payment time.

2.2.6 Normalization

Since we are using costs and times to make recommendations. To get reliable projections, we must normalize these numbers.

2.2.7 Recommendation Generation

The final recommendation is generated by combining normalized time and cost values. The algorithm allows for adjusting priorities between time and cost based on user preferences and also allows the user to choose their own preferences for stores and methods of purchase.



Set a time priority between 1 and 5 (set 0 if you do not care): 0

Set a cost priority between 1 and 5 (set 0 if you do not care): 0

Favourite Stores	Preferred Modes
☐ Target	☐ store
☐ Walmart	☐ pickup
☐ Aldi	☐ delivery

2.3 Prototype Development

Demo of this prototype: Smart Shoppers
Algorithm as shown in the demo: Group 2 Backend Logic.ipynb

In the development phase of our smart grocery project, we implemented a MERN stack application to enhance the grocery shopping experience for users. The application focuses on leveraging big data analytics to provide users with informed decision-making capabilities. Here is an overview of the prototype development:

2.3.1 Database Setup

We utilized MongoDB Atlas to establish a cluster for our database. The database includes collections for products from different stores, retailer information, and customer details.

2.3.2 Frontend Implementation (React)

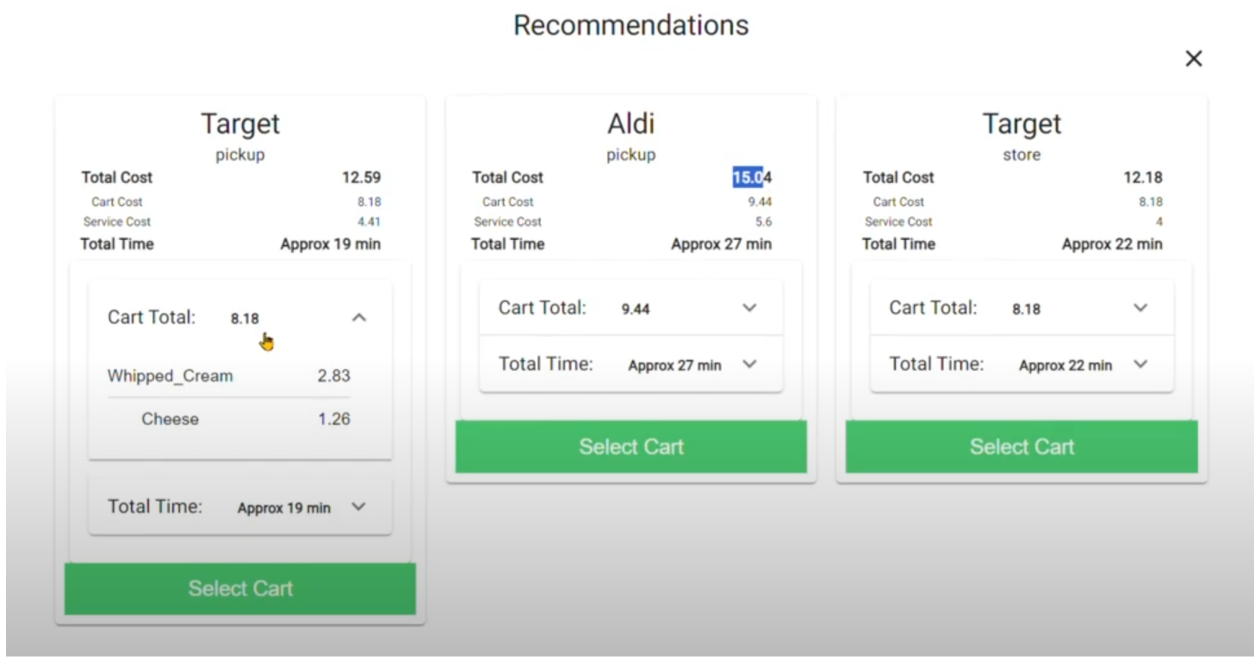
Components and Structure:
1. Functional Components: The frontend is built using React, utilizing functional components for modularity and reusability.
2. Redux State Management: Redux is employed to manage the application state, ensuring a centralized store for data such as user information, cart items, and application-wide states.
3. User Authentication: JWT authentication is implemented to secure user data and ensure authenticated access to personalized features.
4. E-commerce Website:
 - Product Pages: Users can browse products through home or category-specific pages.
 - Search Functionality: A search feature allows users to find products efficiently.
 - User Cart: An interactive cart system enables users to add and remove items seamlessly.

2.3.3 Backend Implementation (Express.js and Mongoose)

Models and APIs:
1. Express.js Backend: The backend is developed using Express.js, providing a robust and scalable server framework.
2. Mongoose Models: Models are defined for users, products, and orders using Mongoose, facilitating interaction with the MongoDB database.
3. Axios for Communication: Axios is used to establish communication between the frontend and backend, ensuring smooth data flow.

2.3.4 Recommendation System Integration:

1. Cart Management: The backend manages user carts, updates cart contents, and processes orders.
2. Recommendation Trigger: Once a user finalizes their cart, a recommendation is triggered by calling the backend. The backend algorithm then generates recommendations based on various factors.



Recommendations

Target pickup		Aldi pickup		Target store	
Total Cost	12.59	Total Cost	15.04	Total Cost	12.18
Cart Cost	8.18	Cart Cost	9.44	Cart Cost	8.18
Service Cost	4.41	Service Cost	5.6	Service Cost	4
Total Time	Approx 19 min	Total Time	Approx 27 min	Total Time	Approx 22 min
Cart Total:	8.18	Cart Total:	9.44	Cart Total:	8.18
Whipped_Cream	2.83				
Cheese	1.26				
Total Time:	Approx 19 min	Total Time:	Approx 27 min	Total Time:	Approx 22 min

2.4 Retailer End View

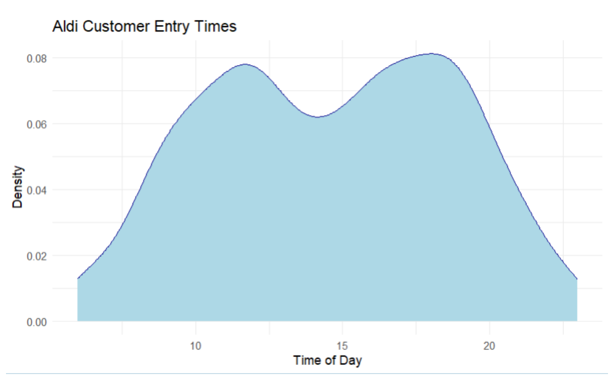
The last goal of our project was to relay this information to the retailers. However, the current setup of our project can’t create a sense of volume of orders of customers. As such, we had to create some sort of volume and usage of our prototype. We did this with random numbers pulled from two normal distributions with varying means and standard deviations denoting the times that customers will interact with the stores.

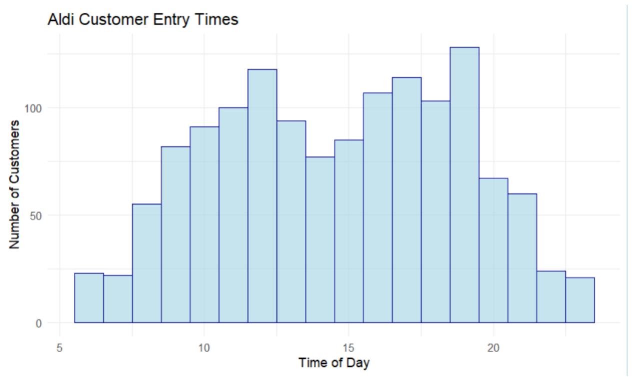
2.4.1 Distributions of customers

We chose two normal distributions as stores normally have two large waves of customers. One during the beginning of the afternoon and another one right after work hours (at around 6:00 PM). Adding these two distributions together create a bimodal distribution that can aid stores in visualizing their customer traffic. This is especially important as stores can then more wisely staff their locations in more cost effective ways. This can be more easily seen by truncating each hour of the distribution into a density, showing how many customers will be there and thus how many employees will be needed for said customers.

2.4.2 Distributions of item purchases

The last part of this was to display what customers would purchase. This was chosen using a gamma distribution to choose the quantity of items that a customer would want to purchase. Gamma was chosen since we expect a large amount of customers wanting a small amount of items and not every possible item that the store can offer. From here, this information can be relayed to stores to inform them of what items customers are looking to buy. This helps with stocking, item placement and pricing if the stores wanted to delve deeper into that themselves. Examples of such for the store Aldi are imaged below:





2.4.3 Reason for this feature

Our project goals included helping the consumers make informed choices and did not necessarily include aiding stores. However, we feel this component is important as it allows stores to sort of buy into our idea of aiding customers. With more up to date and accurate information on in store, pickup and delivery time we're able to refine our process and project, giving consumers more incentive to use our product. This feeds into a growth loop where we can add more stores with more current information, refining, adding onto and improving our data thus helping the customers.

3. Overall Benefits:

1. Convenience and Efficiency:
   - Users gain insights into product availability, prices, and optimal shopping methods.
2. Cost and Time Savings:
   - The system calculates potential cost and time savings, considering discounts, methods of purchase and commuting times and expenses.
3. Personalization:
   - Recommendations are tailored to user preferences, enhancing the overall shopping experience.

4. Timelines and Group Contributions:

In general we did work as a group unit. However there were key parts headed by different members while integration between them was worked on by all. In particular,

1. Akhil and Xander headed data gathering - 2 Weeks
2. Swaroop headed the frontend implementation. - 3 Weeks
3. Akhil headed the recommendation algorithm. - 3 Weeks
4. Akhil and Swaroop headed Backend implementation. - 4 Weeks
5. Xander headed the retailer's end view. - 3 Weeks

5. Future Scope:

The future of this project has a clear path - use real time data. As mentioned in the beginning, we took a lot of this data from preexisting, historic data. This is not the ideal of this project. Ideally we would get data of item prices, traffic data, user location in real time, feed it back into our system, and give out recommendations with said information.

Additionally, we could look towards upscaling the level of complexity and considerations that the algorithm looks at. For example, some customers prefer more organic wholesalers, or some don’t prefer certain stores for personal reasons/history, or others prefer only local stores. These are factors that definitely could be added but would require more data collection, customer feedback and integration and as such is kept as a future topic.

Lastly, we would definitely need to create a comprehensive dashboard for the retailers to use and visualize the data instead of just sending them graphs, data and spreadsheets for them to do the work. Instead we would hope to add a sort of UI, similar to that of the customers but for the retailers to utilize and learn from. This will further aid in our goal to add more stores and gather more data from said stores for our platform.