Neoj4 and SARMIX Model for Optimizing Product Placement and Predicting the Shortest Shopping Path

Preethu Sudharma
Computer Science and DT, ACE, UEL
University Way, London, UK
u2292241@uel.ac.uk

Rasha Hafidh
Computer Science and DT, ACE, UEL
University Way, London, UK
r.hafidh2@uel.ac.uk

Mhd Saeed Sharif
Computer Science and DT, ACE, UEL
University Way, London, UK
s.sharif@uel.ac.uk

Wael Elmedany
College of Information Technology
University of Bahrain, Kingdom of Bahrain
welmedany@uob.edu.bh

***Abstract*— Product placement of top-selling items in highly visible aisles inside supermarkets plays a crucial role in enhancing customer shopping experience. Moreover, it is important for retailers to assure that their customers can effortlessly navigate the store and locate the items they are searching for in a timely manner. The research proposes a novel and effective approach that combines two methods; the SARIMAX model for forecasting sales of each product based on historical data; by using the predicted result, placing the most demanding item in highly visible aisles. And the use of Graph Database Management Systems (GDBMS) such as Neo4j to find the shortest path for consumers to navigate throughout the store to finish the shopping as per their shopping list. By leveraging the power of data analytics and machine learning, retailers can make data-driven decisions that result in improved sales and customer satisfaction. Retailers investing in these technologies and strategies will likely see a significant increase in customer satisfaction and sales.**

Keywords—Efficient product placement, shortest path, Neo4j graph database, sales forecasting, SARIMAX model, customer behavior

I. INTRODUCTION

The use of the SARIMAX model to analyze historical sales data and forecast future sales details, retailers can create an efficient product placement strategy. Neo4j [1] can be used to find the shortest path between the store's aisles, which customers should take to reach the product without any uncertainty. The aim of this research is to assist supermarkets in optimizing product placement within a new shop by utilizing the provided store layout and offering the quickest route for customers to locate the items on their list. These are the critical areas of research in the field of retail management. There are no customer-centric applications available in the retail industry to enhance the shopping experience by considering the customer's valuable time. Hence, the proposed system will have a profound impact on the retail business and can boost the shopping experience for customers. As well as this system addresses the issue of not being able to remember the shopping list while at the store as customers can add items to their shopping list on the mobile app at any time. Likewise, providing the best route to complete shopping by taking the purchase list into account will minimize the effort required to navigate through the store to complete the shopping. Focusing on customer satisfaction is paramount because it is strongly linked to the success and sustainability of a business. Satisfied customers are more likely to choose the same shop as well as recommend the business to others. The contribution of the proposed application combines the SARIMAX model and Neo4j graph database, which provides several benefits such as: *Reduce shopping time:* This application reduces the time spent in the store by looking around in the shop to find the intended product. *Purchase list recallability:* Address the issue of failing to recall the shopping list while shopping. *Enhance customer experience*: An application that prioritizes the customer's needs by helping them to find the product more easily, saving them valuable time. *Improved product placement efficiency:* This strategy increases product placement efficiency which increases sales. *Real-world application*: The proposed application can be applied to all types of shops and warehouses. *Personalized shopping experience:* Customers will feel that the retailer is catering to their specific needs, as they provide the paths to follow during shopping. This can save customers time, which promotes customer loyalty.

II. IMPORTANCE OF PRODUCT PLACEMENT

The first and foremost advantage of placing the products in an optimized way is to increase sales. Allocating products appropriately in aisles is paramount and can have a significant impact on the retailing industry [2]. The customers can notice consumables that are placed in highly visible areas. Placing the highly demanded item in priority areas of supermarkets will maximize sales. A nice allocation of products on shelves increases extra consumption [3]. The sales could downturn if the customer fails to find the intended goods. Research by Weimar, Daniel, et al [4] indicates that the effect of relocation of products to visible areas increased sales from 79.7% to 478%. In addition to boosting sales, retailers can enhance the shopping experience of their valuable customers. By following a systematic approach to product allocation, retailers can improve consumer satisfaction[5]. However, most stores have many goods but limited space. According to [2], placing items on suitable shelves helps retailers optimize the use of space.

III. FACTORS AFFECTING SALES

A pleasant store environment leads to higher customer purchase intention. The store layout, store size, and seasonal purchasing can affect the shop's sales. By placing the commodity in an optimized fashion, the vendor can enhance the shopping experience for their consumers[6]. Research by Haans et al. [7] and Fildes et al. [8] found that the importance of store layout and product placement is crucial in determining sales. These factors can create a sudden desire in consumers, which can lead to an impulsive purchase. Studies on customer impulse purchasing behavior suggest that retailers implement strategies that can influence customers. The range of products offered by a store can significantly impact customers' purchasing behavior [9]. Placing the products in a visually appealing way results in consumers spending more time in the shop and potentially making additional purchases. Moreover, knowledge of outlet availability and time constraints is important when shopping. Customers with more time can spend an extended period within the shop premises and are more likely to make unplanned purchases. On the contrary, those with insufficient time tend to spend less time shopping and purchasing fewer items [10]. Both factors can significantly affect unplanned purchases and brand switching due to difficulties in finding expected brands.

This study aids retailers in shaping their customers' experiences by understanding situational factors that affect sales. Retailers can, therefore, arrange and manage the store environment and operations to satisfy consumers. Furthermore, the lack of appropriate time for purchasing can have an adverse effect. Time pressure can badly affect customers' shopping satisfaction [6], [11]. If customers are unable to recall their purchase list due to the stress caused by time pressure, they may not be able to make the intended purchase. Memory plays a vital role when purchasing goods from an unfamiliar store rather than a regular one. The time required to locate a product may be longer in an unfamiliar store, which can discourage shoppers from returning to the store. According to[10], their study found that time pressure and a lack of familiarity with the store layout can cause customers to switch brands or products because they have difficulty finding the intended item.

IV. Sales Forecastingm

The long-established methods of setting sales goals are no longer effective in helping companies keep up with the competitive market. These methods have no insights into the purchasing patterns of customers. However, significant changes have occurred in the marketing industry due to advancements in machine learning. These advancements help the sales team create effective plans that will boost their business [12]. In this fast-paced life, every business needs to know what its clients expect ahead of time. To determine critical factors such as consumer purchasing patterns, target audiences, and forecast sales for upcoming years, establishments use sales forecasting techniques. These techniques involve forecasting future sales using historical data.

Machine learning in sales prediction allows businesses to understand customer behavior and preferences and make data-driven decisions to increase sales. The author, Pavlyshenko [13] state that forecasting future sales can help retailers plan their activities to improve their business, as accurately predicting sales is a crucial component in a company's strategic planning. Specifically, claims that several managerial decisions, such as product allocation, inventory pricing, ordering and management, largely rely on sales forecasts. Chris Chatfield [14], in his book named 'Time Series Forecasting' stated that time-series forecasting consists of a series of observations that are recorded sequentially over time. For instance, sales data of a specific product at successive months, the particular stock price at regular intervals, the number of daily visitors to a website and so on.

Sales, finance, transportation, healthcare, and weather are some of the common applications of time series forecasting. Forecasting methods can be categorized into judgemental, univariate and multivariate[15]. Commonly used univariate time series forecasting models are Auto Regressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA), seasonal auto regressive integrated moving average with exogenous factors (SARIMAX), Moving Average (MA) and Autoregressive (AR) [16]. The following subsection explains the different models in more details.

A. ARIMA

ARIMA is a widely used methodology for time series forecasting. Utilizing a combination of past values and errors in a linear manner. The authors of article [17], explored autoregressive and moving average components. AR determines how much the current value is influenced by past values while MA models the error term. ARIMA is labelled as ARIMA (p, d, q), non- seasonal (trend) parameters as follows:

p: Order of autoregressive components

q: Order of moving average components, this refers to the number of lagged errors that are utilised to forecast the value

d: Defines the number of times the data was differenced to achieve stationarity

The steps involved in the ARIMA model are preparation of data, identification of model, estimation and selection of parameters, validation of model and ultimately utilizing the model for prediction [18]. First, it is mandatory to identify whether the time series is stationarity. Most of the statistical techniques assume data is stationary and it cannot handle any form of non-stationarity, such as trends or seasonality. Predicting values becomes easier when the statistical properties, such as mean and autocorrelation structure, exhibit consistency over time. The initial step is to remove the non-stationarity and create a time series that is stationary. Differencing is applied to stabilize the variance. Applying high-order differencing can transform non-stationary data in to a stationary data. It is necessary to perform a differentiation on the time series d. Excessive differencing is likely to lead to an increase in standard deviation, so it is important to make the standard deviation as low as possible while performing differencing.

The next step after data transformation is model identification for determining the possible values for p, q and d of the AR and MA components. This can be clarified using the Augmented Dickey- Fuller (ADF) test, please refer to article [19] for in-depth explanation. The ADF test can determine if the data originates from a stationary process by testing for stationarity. On other hand, we can say model identification seeks the answer to whether data is stationary or not. To obtain first-order differencing it is better to begin with the smallest order ($d = 1$). First-order difference refers to the difference of data between the current and the immediately previous one. A positive outcome implies that the time series is not stationary, while a negative value indicates stationarity.

p<=0.05: Data is stationary. H1:A unit root is not present in the series.

p>0.05: Data is non-stationary. H0: The series exhibits a unit root.

By conducting the ADF test, we can proceed towards determining the suitable values for p in AR and q in MA. Ultimately, the selected model is fitted to the time series data. Once we obtain the forecast it will be easy to make the right decisions.

B. SARIMA

SARIMA is an extension of ARIMA where three more hyperparameters state the autoregression, moving average and differencing [20]. The main four components of SARIMA are;

- Seasonal Autoregression: This component models how the variable depends on its past values at the same time of year.
- Seasonal Integration: The component SI involves taking seasonal differences of the data in order to remove any trend or seasonal patterns.
- Autoregression: AR modelling the relationship between the variable and its lagged values at non-seasonal intervals.
- Moving Average: This component involves modelling the error term of the model as a linear combination of past error terms.

Configuration for SARIMA requires hyperparameter selection for both trend and seasonal components (SARIMA (p, d, q) (P, D, Q) m). Where (p, d, q) refers to the order of time series, the same order was used in ARIMA but does not consider seasonality. The seasonal order, (P, D, Q)m refers to components of time series. In order to attain stationarity for Seasonal ARIMA, adjustments are made for both trend and seasonality. Seasonal parameters are;

P: Seasonal auto-regression order

D: Seasonal differencing order

Q: Seasonal order of moving average components

M: Number of times steps in the seasonal cycle. P, Q and D are parameters that are affected by the value of M.

C. SARIMAX

SARIMAX [21], is the result of the implementation of SARIMA in which X suggests that implementation also favours exogenous variables which reduces error and improves accuracy. X can be external parallel time series data. According to [20], the SARIMAX technique employs a time-series methodology that integrates both seasonal and exogenous factors to reduce error values and improve the overall precision of the model. In addition, the SARIMAX predictive model has exhibited the capability to handle diverse types of historical data. As per the experimental results, the SARIMAX model demonstrates favourable performance in comparison to the simpler autoregressive integrated moving average based technique.

D. ARIMA, SARIMA, SARIMAX Comparison

The below table shows the major difference summarized between the three forecasting approaches as per the study by [14], [16], [21].

TABLE I. ARIMA, SARIMA AND SARIMAX COMPARISION

ARIMA	SARIMA	SARIMAX
Autoregressive Integrated Moving Average	Seasonal Autoregressive Integrated Moving Average	Seasonal Autoregressive Integrated Moving Average with Exogenous Variables
Predictions of future values are made by considering past values	The extension of ARIMA includes additional parameters to account for seasonal patterns in data	The extension of SARIMA allows the inclusion of exogenous variables, which can be used to capture external factors
ARIMA (p, d, q)	SARIMA (p, d, q) (P, D, Q) m	SARIMAX (p, d, q) (P, D, Q) m
AR, MA and I are the components	Include additional components SAR, SMA and SI	Includes additional variable X
Not able to handle seasonal data effectively	Specifically designed to handle seasonal time series data	An additional capability of including an exogenous variable that may affect the time series

V. SHORTEST PATH OPERATION

In the research [22], the authors noted that the shortest path refers to the path between two points such that the sum of the weight of its constituent edges is minimized. According to [23], customers in supermarkets can mainly come under two categories. One among them is those who shop without purpose and add items to their cart when they come across them. The second category consists of those who come with a specified shopping list, meaning they are shopping with a certain purpose. Finding the intended item can be quite challenging, especially if the shop is large. Therefore, it is important to find the best route for customers in order to save shopping time.

There exists a numerous number of algorithms to find the shortest path. Dijkstra's and Bellman-Ford algorithm [24], and graph databases are some of the techniques used to find the shortest path. The research [25], [26], highlights the emerging advantages of Graph Database management System (GDBMS) and how it is outperforming Relational Database Management System (RDBMS) . In a relational database management system, the connection between nodes is represented in tables that are connected by primary and foreign keys. Executing queries for graphs is complex since several table joins are required to calculate the shortest path between nodes. The performance of the process significantly reduces. GDBMS is capable of handling large amounts of graph data as data is structured in a graph-like format. The data model of the graph involves defining nodes, edges and attributes that describe the characteristics linked to the nodes or edges. It stores all the relationships directly in data allowing it to traverse through the graph structure efficiently.

According to [27], the graph model can be considered as a collection of objects (such as people and places) and the relationships between them (such as friend or living). A network is formed in graph databases by utilizing these objects and their relationships. The definition of graph structure is as follows:

$$G= (V, E)$$

where V= {v1, v2,v3,...,vn} represents a set of vertices
and E= {e1, e2,e3,.....,en} represents set of edges

The Neo4j [28], has been used for finding the shortest path. Neo4j has been developed in Java programming language and is one of the graph databases that has been operational since 2003 onward, its open source which makes it widely available for developers and researchers. Neo4j use graph traversal queries, instead of using mathematical calculations. The Bidirectional Traversal algorithm [29], is a feature of Neo4j, its advantage is to increase the speed of calculations. Moreover, the Neo4j server can communicate with clients via Representational State Transfer (REST) Application Programming Interface (API).

The APOC library in Neo4j offers a graph traversal algorithm called "apoc.algo.dijkstra", which utilizes Dijkstra's algorithm to determine the shortest path between two nodes in a graph. It's written in Java and uses the graph data structures provided by Neo4j to perform its calculations. This algorithm considers the weights of the relationships between the nodes to find the most optimal path. According to [30], the Dijkstra's algorithm starts at a source node and explores its neighbors iteratively, updating the path and distance to each neighbor if a shorter path is discovered. The process continues until either the destination node is reached or there are no more nodes left to explore. Assuming that the edge weights are non-negative, the algorithm ensures that it has found the shortest path to the destination node.

VI. Research Methodology

The application was developed by utilizing of Python programming language, while sales data was processed with NumPy and Pandas libraries [31], a powerful data analysis toolkit. Within Python, the Pandas library offers a comprehensive set of data analysis tools. SARIMAX was used to develop predictive models. FastAPI [32] is a Python web framework that excels in performance when building APIs. Swift, a programming language developed by Apple Inc., is designed for easy readability and writing. UIKit, also developed by Apple Inc., is a primary framework used for creating graphical user interfaces for iOS. Developers can use Interface Builder, a component of Xcode, to design and visually layout user interfaces. MySQL is utilized for storing product location and user details, as well as fetching store layout information. Neo4j was used for finding the shortest path. Imputing missing values with mean or median is not required in "GrocerySales.csv" as it has no missing values.

To increase readability, the unnecessary columns such as Order Id, Customer Id, City, Region, and Sales was dropped. Two column names, Profit and Subcategory, was renamed to Sales and Product respectively. A label encoder is used to convert a categorical value (Product) to numerical variables. The next step involved splitting the month and year from the order date, as prediction is based on a monthly basis. The year is required for grouping the product based on month and year.

A. High Level Diagram of Proposed System

As can be seen from Fig. 1, a proposed system high level diagram (HLD), has an interface designed for mobile devices through which store managers and customers can access the service. Using the mobile application, the store manager will input future dates, which the FastAPI engine will analyze to predict highly demanded items. These items can then be appropriately placed in the aisles. After receiving the shopping list through the customer application, the proposed system will deliver the most efficient route for customers to take based on the items they need.

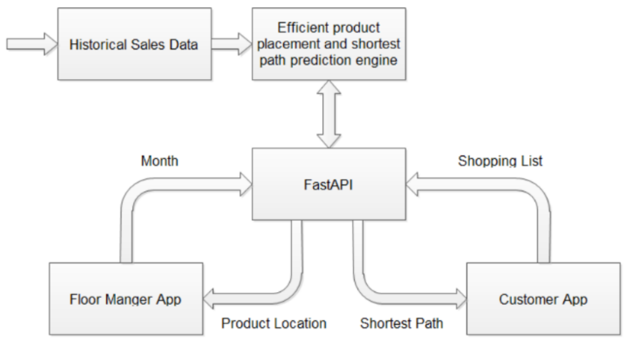

Fig. 1: HLD of the proposed system.

B. Data Flow Diagram of Proposed System

The Data Flow Diagram (DFD), Fig 2. is a graphical representation of the proposed system, showing data flows through different stages. Different symbols are utilized in DFDs to represent distinct entities, including data sources, processes, data stores, and data flows. Description of DFD steps:

1. *Data Set:* The proposed system is a generic application, so it can be trained with any kind of sales dataset having the date and product columns. The chosen dataset "GrocerySales.csv", is only for demo purposes. Was collected from the supermarket sales data for year 2022 from Kaggle. The column includes customer Id, category, subcategory, city, date, region, sales, and profit.

2. *Data Preprocessing:* In data preprocessing the raw data is transformed into a form suitable for models of machine learning.

   a) Model training: SARIMAX time series forecasting technique was used for prediction by using historical data. Once the model is trained, it becomes capable of forecasting sales for a particular timeframe that lies ahead.
   b) Store layout: Retrieving the store layout information inputted by the store manager from the Neo4j database.

c) Product placement: The placement of the products was be based on the knowledge acquired through the forecasting process.
d) Product list: The FastAPI will fetch all the available products from MySQL database.
e) Shortest path: Dijkstra's algorithm in Neo4j, along with custom sorting, can was usedto find the shortest path for the items on the customer's shopping list. The goal is to minimize the distance travelled by starting at the entry and picking up items along the way until reaching the checkout.
f) Neo4j: A Neo4j graph database was used to store user details and layout information.
g) MySQL: It stored product details and their respective shelves number

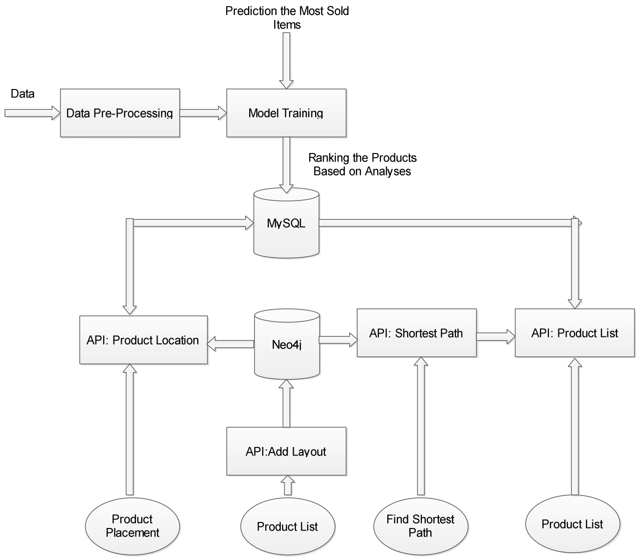

Fig. 2: DFD of the proposed system.

VII. Case Scenario

The Fig 3. shows the store layout information of the supermarket, which can serve as a reference for the demo of the proposed system. The physical arrangement of the store helps customers find what they are looking for quickly. The layout of the store can be divided into three areas: the entrance, center store, and checkout area.

a) Entrance: The entrance is the first point of contact for customers.

b) Centre store: The center store area is where items are displayed.

c) Checkout: Finally, customers can pay for their items at the checkout

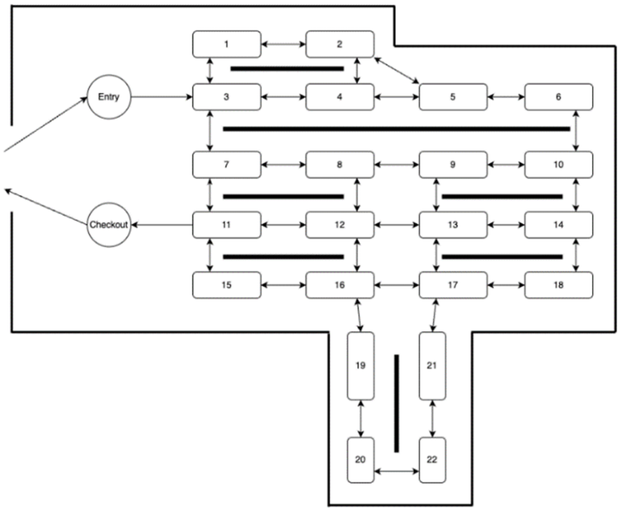

Fig. 3: Case scenario proposed layout of a supermarket.

VIII. Implementation

Neo4j database is used for the implementation of the proposed application. Neo4j is an efficient graph database that can be used to model complex relationships. It is particularly well-suited for applications that involve product placement and shopping path optimization. One of the key advantages of using Neo4j for product placement is that it allows retailers to model the relationships between shelves in the store. Similarly, Neo4j can be used to identify the shortest path for shopping by analyzing the physical location of the store. By creating a graph-based model, as can be seen from Fig 3, it helps to find the most efficient route for customers as they move through the store. In this case, aisles can be represented as nodes in the graph, and the connections between the aisles can be represented as edges. To run the shortest path algorithm in Neo4j, a Cypher query is used. The query specifies the start and end nodes for the path. The algorithm searches the graph for the shortest path using a variant of Dijkstra's algorithm. Neo4j returns the shortest path between those nodes and the weight/distance between the given start and end node.

a) Collect Sales Data

As the first step of implementation, previous year's sales data was chosen for a supermarket from the Kaggle community (GrocerySales.csv). The entities used such as Order ID, Customer ID, Category, City, Region, and Sales.

b) Model training and prediction

A cloud-based development environment called Google Collaboratory is used for training the model. It is owned by Google, which offers a variety of features, such as free access to computing resources like CPU, GPU, and TPU. Moreover, it has built-in support for data science libraries. Colab notebooks are stored on Google Drive. It has popular libraries, such as EnsorFlow, Keras, PyTorch, and Scikit-learn, for building and training models, as well as Pandas, NumPy, and Matplotlib libraries for data analysis. To prepare the sales data for modelling, the following was carried out:

- Column rename*:* Rename the columns “Profit” as “Sales” and “Subcategory” as “Product” for readability.
- Check for missing values*:* There is a function named isnull() in Python that is used to detect missing values in pandas. The “GrocerySales.csv” does not have any missing values, as isnull() returns zero for all the columns.
- Transforming data: Transforming the Order Date into date format with to_datetime, which is a method in the panda's library that can convert a string into a DateTime object. The LabelEncoder is a utility class provided by the scikit-learn library that converts categorical values into numerical values. Each product name is assigned a unique integer value.
- Grouping data: The Groupby operation is performed on the specified columns: Product_id, Month and Year. The resulting groups are then aggregated by summing the values in the Sales column for each group.
- Data visualization: Matplotlib is the software used in the process of creating visual representations to help understand data and insights.
- Building model: Once data preparation is completed, the SARIMAX model is built using Python and the statsmodels library. After creating the prediction model, the model file for each product is saved as a pickle file. Additionally, a CSV is generated for mapping the products and their related model files, along with the date of sales. There is a purpose for taking the last sales date because each product has different datasets. The prediction date starts after the last sales data. Then, the shortest path algorithm is used to identify the products in the store layout based on customer shopping lists, which are stored in a Neo4j graph database.
- Sales forecast*:* Once the model is built, use it to forecast product demand for the upcoming months. The prediction is loaded using Python with fast API, and it saves the data to a MySQL database for quick reloading. Then, the prediction results are used to place products efficiently inside the store. The diagram below shows the forecast for upcoming years.

c) Mobile application

- The mobile application is developed in iOS, using the Swift programming language and the Xcode development environment. Developing an iOS application typically involves the following steps:
- Define the app: Start by defining the purpose and identify the target audience.
- Plan the app: Plan the functionality, user interface, and features of the app.
- Xcode configuration: Install Xcode and set up a new project.
- User interface: Use Xcode's Interface Builder to design the app's user interface, which includes screens, buttons, labels, and other elements.
- Implementation of functionality: Use Swift to write code that implements the functionality, including responding to user interactions such as user log-in, adding layout and product placement location and finding the shortest path.

IX. Results

A. Adding Purchase List

Once the customer logs in to the application, it will display the list of all products available in the store. The customer can add items to their shopping list by clicking the "Add" button located on the right side of the product name. Additionally, they can remove products from the purchase list by clicking the same button again. By clicking the button on the top left, the customer can see the shortest path to complete their shopping.

```
async def get_prd_list():
   q1 = "SELECT shelf,prd_name FROM
product_tbl;"

   cursor = mydb.cursor()
   cursor.execute(q1)
   res = cursor.fetchall()
   prd_arr = []

   for i, item in enumerate(res):
       shelf = item[0]
       prd_name = item[1]

prd_arr.append({"shelf":shelf,"prd_name":p
rd_name})
   if len(prd_arr) == 0:
       row = {"status": False, "message":
"Wrong username or password"}
       json_data = json.dumps(row)
       return Response(content=json_data,
headers=headers)
   else:
       data = {}
       data["status"] = True
       data["data"] = prd_arr
       json_data = json.dumps(data)
       return Response(content=json_data,
headers=headers)
```

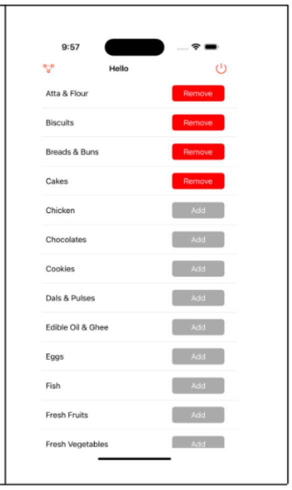

Fig. 4: Code and UI for adding purchased list.

B. *Shortest Path Identification*

The application would identify the starting point and ending point for the route, and then use the built-in Dijkstra's algorithm, “apoc.algo.dijkstra”, to calculate the shortest path between the two points. The algorithm considers the weights assigned to the edges in the graph to determine the most efficient route for the customer. Afterward, the application would display the suggested route to the customer, which may include information about the location of each item on their purchase list in each aisle. This feature aims to make it easier for the customer to find what they need.

The implementation of this approach can provide useful insights into how data-driven techniques can be used to optimize product placement strategies in supermarkets. The findings can help retailers enhance their customer experience and sales by placing products in the most effective locations in the store layout. The implementation involves several steps: data collection and preprocessing, model training and prediction, product placement, and shortest path strategy.

```
def get_shortestPath(self, start,target):
  query = "match (s:shelves {name:"
  query += str(start)
  query += "}),(e:shelves {name:"
  query += str(target)
  query += "}) call
apoc.algo.dijkstra(s,e,'connected','weigh
t') YIELD path, weight unwind nodes(path)
"as node return node.name as node_name,
weight"

  with self.driver.session() as session:
      result = session.run(query)
      weight = 0
      path = []
      for record in result:

path.append(record["node_name"])
          weight = record["weight"]

      return {"weight":weight,
"path":path}
```

Fig. 5: Code and UI for shortest path identification.

X. Conclusion

Product placement in a supermarket plays a crucial role in enhancing customer satisfaction. The goal of this research is to develop an efficient practice for product placement based on prediction results and give the shortest path to complete the shopping based on the customer's shopping list. A solution that combines two methods: the SARIMAX model for making an efficient product placement strategy by forecasting monthly sales of each product based on historical data and finding the shortest path for consumers to navigate throughout the store to finish the shopping as per their shopping list using Neo4j graph database. Overall, this research provided a novel and effective approach for efficient product placement in a supermarket, which lead to improved customer satisfaction as customers will feel that the retailer is catering to their specific needs, as they provide the paths to follow during shopping. This can save customers time, which promotes customer loyalty. The study has contributed to the field of retail analytics by demonstrating the effectiveness of time series forecasting models.

References

[1] B. M. Sasaki, J. Chao, R. Howard, B. M. Sasaki, J. Chao, and R. Howard, "Graph Databases for Beginners Graph Databases For Beginners," *EBook*, 2018.

[2] R. Ali, "The Impact of Shelf Position and Space Allocation on Customer Purchase Decision: The Mediating Role of Customer Brand Awareness," Jun. 2020.

[3] G. Aloysius and D. Binu, "An approach to products placement in supermarkets using PrefixSpan algorithm," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 25, no. 1, pp. 77–87, 2013, doi: 10.1016/j.jksuci.2012.07.001.

[4] D. Weimar, C. Deutscher, and R. Decker, "The sales effect of in-store product displays: The special case of total product relocation," *J. Bus. Retail Manag. Res.*, vol. 15, no. 01, pp. 1–14, 2020, doi: 10.24052/jbrmr/v15is01/art-01.

[5] D. Landa-Silva, F. Marikar, and K. Le, "Heuristic Approach for Automated Shelf Space Allocation," in *Proceedings of the 2009 ACM Symposium on Applied Computing*, 2009, pp. 922–928, doi: 10.1145/1529282.1529482.

[6] F. de Oliveira Santini, W. J. Ladeira, D. C. Pinto, M. M. Herter, A. S. Mattila, and M. G. Perin, "Retail crowding: meta-analysis of contextual and cultural moderators," *Mark. Intell. Plan.*, vol. 40, no. 1, pp. 57–71, Jan. 2022, doi: 10.1108/MIP-03-2021-0076.

[7] H. Haans and E. Gijsbrechts, "'One-deal-fits-all?' On Category Sales Promotion Effectiveness in Smaller versus Larger Supermarkets," *J. Retail.*, vol. 87, no. 4, pp. 427–443, 2011, doi: https://doi.org/10.1016/j.jretai.2011.05.001.

[8] S. K. R. F. Shaohui Ma, "M P RA Munich Personal RePEc Archive Trust, Reciprocity and Rules," no. 89356, p. 77, 2019, [Online]. Available: https://mpra.ub.uni-muenchen.de/89356/1/MPRA_paper_89356.pdf.

[9] P. Igbongidi, "Retail Store Merchandise Assortment and Display and their Influence on Consumer Impulse Buying Behaviour in North-West Nigeria Retail Store Merchandise Assortment and Display and their Influence on Consumer Impulse Buying Behaviour in North-West Nigeria," Jan. 2015.

[10] E. Nilsson, T. Gärling, and A. Marell, "Effects of time pressure, type of shopping, and store attributes on consumers' satisfaction with grocery shopping," *Int. Rev. Retail. Distrib. Consum. Res.*, vol. 27, no. 4, pp. 334–351, Aug. 2017, doi: 10.1080/09593969.2017.1309674.

[11] R. Mehta, "Understanding perceived retail crowding: A critical review and research agenda," *J. Retail. Consum. Serv.*, vol. 20, no. 6, pp. 642–649, 2013, doi: https://doi.org/10.1016/j.jretconser.2013.06.002.

[12] R. Praveen, D. P. Kumar, A. P. Sam, and G. S. Sundari, "Sales prediction using machine learning," *AIP Conf. Proc.*, vol. 2444, no. June, pp. 3619–3625, 2022, doi: 10.1063/5.0078390.

[13] B. M. Pavlyshenko, "Machine-learning models for sales time series forecasting," *Data*, vol. 4, no. 1, pp. 1–11, 2019, doi: 10.3390/data4010015.

[14] C. Chatfield, *Time-Series Forecasting*. CRC Press, 2000.

[15] C. Chatfield, "What is the 'best' method of forecasting?," *J. Appl. Stat.*, vol. 15, no. 1, pp. 19–38, Jan. 1988, doi: 10.1080/02664768800000003.

[16] D. C. Fahad Radhi Alharbi, "A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach," *Inventions*, vol. 7, no. 94, 2022.

[17] J. Fattah, L. Ezzine, Z. Aman, H. El Moussami, and A. Lachhab, "Forecasting of demand using ARIMA model," *Int. J. Eng. Bus. Manag.*, vol. 10, pp. 1–9, 2018, doi: 10.1177/1847979018808673.

[18] R. Zhang, H. Song, Q. Chen, Y. Wang, S. Wang, and Y. Li, "Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China," *PLoS One*, vol. 17, no. 1 January 2022, 2022, doi: 10.1371/journal.pone.0262009.

[19] E. Herranz, "Unit root tests," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 9, no. 3, pp. 111–139, 2017, doi: 10.1002/wics.1396.

[20] H. Chhabra, "A Comparative Study of Arima and Sarima Models to Forecast Lockdowns Due to Covid-19," *Res. Sq.*, 2022.

[21] F. Alharbi and D. Csala, "A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach," *Inventions*, vol. 7, p. 94, Oct. 2022, doi: 10.3390/inventions7040094.

[22] R. S. Kumar, C. Rani, and P. G. Kumar, "Predicting shortest path for goods delivery to fair price shops in India," *2017 4th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2017*, 2017, doi: 10.1109/ICACCS.2017.8014631.

[23] X. Chen, Y. Li, and T. Hu, "Solving the supermarket shopping route planning problem based on genetic algorithm," *2015 IEEE/ACIS 14th Int. Conf. Comput. Inf. Sci. ICIS 2015 - Proc.*, pp. 529–533, 2015, doi: 10.1109/ICIS.2015.7166649.

[24] F. Mukhlif and A. Saif, "Comparative Study On Bellman-Ford And Dijkstra Algorithms," *Int. Conf. Commun. Electr. Comput. Networks (ICCECN 2020)*, no. April, pp. 1–2, 2020.

[25] G. Jin and S. Salihoglu, "Making RDBMSs efficient on graph workloads through predefined joins," *Proc. VLDB Endow.*, vol. 15, no. 5, pp. 1011–1023, 2022, doi: 10.14778/3510397.3510400.

[26] Y. Cheng, P. Ding, T. Wang, W. Lu, and X. Du, "Which Category Is Better: Benchmarking Relational and Graph Database

Management Systems," *Data Sci. Eng.*, vol. 4, no. 4, pp. 309–322, 2019, doi: 10.1007/s41019-019-00110-3.

[27] M. Miler, D. Medak, and D. Odobašić, "The shortest path algorithm performance comparison in graph and relational database on a transportation network," *Promet - Traffic - Traffico*, vol. 26, no. 1, pp. 75–82, 2014, doi: 10.7307/ptt.v26i1.1268.

[28] A. Sacco, "Neo4j: Graph Database," *Begin. Spring Data*, no. January, pp. 231–247, 2023, doi: 10.1007/978-1-4842-8764-4_9.

[29] W. Pijls, "A new bidirectional algorithm for shortest paths," *Intelligence*, no. July, pp. 1–6, 2008.

[30] J. C. Dela Cruz, G. V Magwili, J. P. E. Mundo, G. P. B. Gregorio, M. L. L. Lamoca, and J. A. Villaseñor, "Items-mapping and route optimization in a grocery store using Dijkstra's, Bellman-Ford and Floyd-Warshall Algorithms," in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 243–246, doi: 10.1109/TENCON.2016.7847998.

[31] W. McKinney and P. D. Team, "Pandas - Powerful Python Data Analysis Toolkit," *Pandas - Powerful Python Data Anal. Toolkit*,

[32] T. Point, T. Point, and T. Point, "FastAPI," *Organ. Behav.*, pp. 1–10, 2014.