University of Arkansas – CSCE Department
Capstone I – Final Proposal – Fall 2021

Walmart: Predictive Planning, Ordering, and Monitoring

Kyle Orman, George Romano, Abigail Tee, Joshua Thornburgh, and Margaret Turner

Abstract

Our main objective is to create a program that assists Walmart associates when ordering store supplies. To help associates order supplies more efficiently, we will be using machine learning (ML) that forecasts what items need to be ordered and the quantity needed to be purchased. This will automate the process of ordering these supplies, so associates do not have to do it manually. Associates will still have to verify that orders are correct and have the option of removing or adding to them as the need arises. Overall, this will increase efficiency, minimize human error, and avoid costly emergency supply orders when associates are ordering store supplies.

1.0 Problem

Current Walmart associates order store supplies manually by planning what they need ahead of time on a week-to-week basis. This includes handheld scanners, office equipment, plastic bags, light bulbs, software, servers, etc. They also pick the amount of each item when going through the shopping catalogue.

By planning these supplies manually week-to-week, it raises opportunity for mistakes and reduces the efficiency of the associates by spending more time on supply planning instead of their main duties. If the supplies needed were to be pre-generated, then this could improve efficiency and reduce mistakes for employees because it reduces the steps needed for assembling a supply order.

If a solution were to never be in place, then continuous mistakes are likely to be made by associates in stores. Some associates can be pressed for time and place their orders too fast and cause the orders to be inaccurate. This can also lead to associates not having their required supplies,

such as a light bulb missing in a section of the store, or a few associates not having a handheld scanner. These mistakes are human error and resort to more expensive emergency ordering.

A solution to this problem has been implemented before and failed. This is likely due to the fact that they do not keep stock numbers on store items not available for purchase.

2.0 Objective

The objective of this project is to help associates order supplies more efficiently by forecasting what items need to be ordered and the quantity of them. Currently employees must evaluate the need for store supplies and manually purchase the needed supplies through a catalog available to Walmart retail store locations. The goal is to capture the demand signal of supplies as well as foresee possible delays on the orders in order to predictively create business supply purchase orders with minimal intervention on behalf of the employees. It is based on trends, patterns, historical purchases, and logistics along with the increased demand of holiday and seasonal purchasing.

To achieve our objective, we will be implementing a machine learning model in Python based on historical Walmart retail location business supply order purchase data. The machine learning model will produce a predictive algorithm to forecast needed supplies. Through a simple web interface that allows a user to input the store number and employee credentials, a supply order will be populated with needed supplies that will be manually verified by a store employee to complete the purchase order.

3.0 Background

3.1 Key Concepts

For the front-end development, we will build the graphical user interface of a website through the use of HTML and JavaScript. Hyper Text Markup Language (HTML) is the backbone of any website development process. JavaScript is an event-based imperative programming language that is used to transform an HTML page into a dynamic interface. These implementations will allow the user to interact with the website.

The back-end development will focus on how our website functions. It will lay the foundational code that will enable the website to process the actions of the user on the front-end and deliver the correct information in return. The technology of the back-end is a combination of servers, applications, and databases. Our programming in this area could include writing APIs, creating libraries, working on data architecture, and writing code to interact with our database.

Lastly, we will use machine learning to implement our project. Machine learning is the study of computer algorithms that can automatically improve through experience and the use of data, which will be provided to us.

3.2 Related Work

Many companies are using machine learning for automated ordering. [4] In an IRJET article, the proposed design of accomplishing an inventory management system was tagging the warehouse components with a RFID (Radio-Frequency Identification) tag. The data these tags hold is then backed up to the cloud for future use. From there, you can see what is and is not in stock. This implementation method will be different than ours as we will not have stock numbers for the items being reordered.

[1] Another article looks at artificial intelligence for inventory management. It mentions that Amazon implemented artificial intelligence throughout their inventory operations. The article then mentions two key implementations of artificial intelligence for inventory. These are Demand Prediction for Inventory Management and Reinforcement Learning systems for full-inventory management. The method that would be a better approach for what we are doing is Demand Prediction for Inventory Management. The general idea of this method is to build a time series prediction model that can estimate what demand will be like for the coming days across all items in your inventory. This is what we are wanting to accomplish.

In the implementation of the predictive planning and ordering, Walmart representatives mentioned they wanted a "Did you forget" pop-up when finalizing the stock order. [3] An article describes this as a recommendation engine. This article defines a recommendation engine as, "information filtering tools that use algorithms and data to recommend that most relevant items to a particular user in a given context." This will be beneficial in implementation as it will decrease human error and increase order efficiency in checkout.

[2] An Unleased Software article, "Using Machine Learning in Inventory Management." discusses reducing forecasting errors. With machine learning technology, predictions can be made continuously using data to adjust forecasts to suit companies and account for more factors than typical forecasts. This is important to take into account when using machine learning because it can predict demand in the future and allow the correct quantity to be purchased in time of need before the time of need.

4.0 Design

4.1 Requirements, Use Cases, and Design Goals

Requirements –

1. Website frontend
    a. Hosted on Heroku
    b. Input fields
        i. Store number
            1. Restricts the ordering options to location
        ii. Employee login information
            1. Restricts the ordering budget by employee tier
2. Database backend
    a. A relational database

        i. PostgreSQL or equivalent
        ii. Two tables needed
            1. One table with supply description, ID, and price
            2. A second table with order history information including date, item ID quantity, store number, and a value to flag if the order was an emergency order
            3. The respective IDs will be primary keys within the relational database
    b. Historical supply ordering data to be provided by Walmart
        i. The data will be the basis for the machine learning model
3. Machine learning model
    a. We will be responsible for developing our own algorithm
    b. The ML model must take input
        i. Historical supply ordering data provided by Walmart
            1. The ML model will query the database containing the data item by item in order to formulate the predictive model output
    c. The ML model must provide output
        i. A predictive algorithm of supply needs
        ii. The supply needs will be manually verified by an employee

Use Cases –

The primary use case of this project is for restocking and reordering Walmart store locations with business supplies that employees regularly use. The website frontend interface will allow an employee to login using their store and employee credentials—the available budget and supplies to order will be restricted by location and employee tier. In addition to streamlining the supply reordering process and minimizing the amount of time that it takes for employees to evaluate and order needed supplies for day-to-day operation of Walmart retail locations, the tool we will develop will allow Walmart stores to avoid emergency supply orders for urgently needed supplies that come with increased cost.

There are two possibilities for expanding the use case once the base case is achieved. First, we can expand the use case to restock and reorder the entire store inventory including products and goods meant for sale to Walmart customers. A second possibility is that we can apply the machine learning model to assist with management of corporate office supplies—this would require a different dataset of historical supply orders but the same execution.

Design Goals –

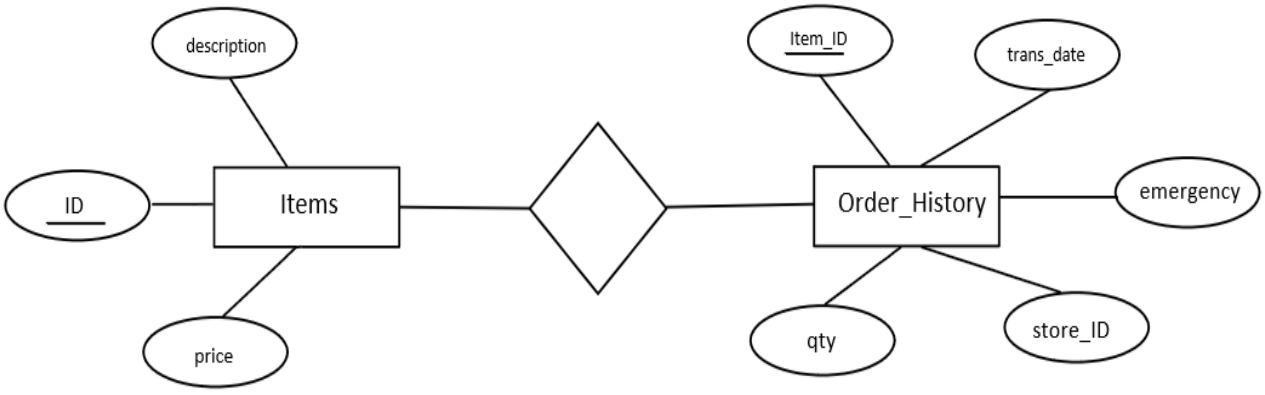
Our design goal is to ultimately develop a system that allows us to efficiently execute our primary use case. Past store supply order data will drive a machine learning model that yields a predictive algorithm that will be used for supply ordering and restocking. This includes a minimalist website frontend that accepts store and employee credentials and a database backend that stores the historical data provided by Walmart. The database backend will have historical supply order data provided by Walmart stored within an appropriate relational database.

4.2 High Level Architecture

A high-level point of view we will have three major components, the database, the front end, and the machine learning (ML). Currently we do not know what the data looks like, but we can predict it will have a name, description, price, quantity to a box, and an item ID. The item ID will be our key for the product table with another connect table full of the order history, this would be what we use to predict future ordering.

A design focus for this project is to auto-generate a list of items to be ordered for the store to function. Initially, we will design for single store use and can expand to a more generalized model that could include regional attributes. Before ordering happens, the user will review and confirm that the order is correct and adjust the order if the order appears incorrect. A mock webpage of the ordering system will be crafted to better showcase the ML model. An email could be generated that notifies the persons in charge of ordering requirements once the allotted time for ordering comes.

Some major design hurdles that will come up include implementing a webpage that can query a database with the security behind it to be useable on a commercial level. Another major obstacle will be implementing a machine learning model that can accurately predict what the company is needing. Setting up the database is a trivial task. A relational model to describe the tables is as follows:



The image above shows that we will have two tables. One table will be for an item's information. This will include attributes such as ID, price, and a description of the item. The second table will include order history. This table is connected to the items table by the foreign key Item_ID. Attributes for order history are item ID, transaction date, emergency (was the order an emergency), quantity on the order, and store id that the items were shipped to. When querying the database, we will look at an items ID and the pull the subsequent rows from order history. The ML model will then calculate what items may be nearly out of stock.

Once a week the model will examine every item in our database and compile statistics on what needs to be ordered. That data compiled will then be sent to the user who finalizes the ordering. Upon receipt, the user checks for accurate numbers and adjusts to the store's needs. After adjustments, the orders may be sent to be approved and the transaction is complete. One such hurdle will be implementing this with zero stock numbers. A problem that we will run into is not

having an accurate initial state. Since the user is able to tweak an order as they are derived, this should auto correct itself.

Not much technology will be needed because most of what we are doing is software focused. From previous projects, experience with Heroku will be valuable in hosting the webpage. For the database hosting, our team is thinking on using the University's Oracle DB. If that is not an option, we will then result to using a Raspberry Pi.

4.3 Risks

Risk	Risk Reduction
Potential SQL injections	Using prepared statements
Dropping the DB	Backing up regularly
Potential Overordering	Webpage frontend that associates can use to confirm orders

4.4 Tasks

Understand the given data

- We are currently waiting for the 6 months of data from to determine how to classify items. (Do we need to consider tools separate from items that are used and thrown away? Like toilet paper and paper towels.)

Researching Machine learning

- Since our team has limited to no experience with machine learning, we will be researching what makes an accurate ML model. Are there items that have relations? Or does it only seem like they do?
- A longer time period has been given to this as it is the area that will have the largest weight for the project.

Designing the DB to pull from

- Considering the large scale of data that we will be receiving; we will need to design our database with quick queries in mind. A cluster design would be quick, but would require a lot of maintenance. A sorted design with binary search should be more than sufficient.

Designing the frontend that the user interacts with

- We will design a mock front end that show what will show which items may need to be ordered. The user will still need to confirm before the system orders.

Designing the algorithm to predict orders

- Once the research is done into ML, we will begin designing the algorithm to predict future needs. While in the design phase, we will select a small subset of data to tweak the system. Without the data, it is hard to say what we do and do not have.

Implement designs

- Once the database structure is complete, front end designed, and the ML model tweaked, we can then begin setting everything up.
- The database will need to be filled with the given data and tested for accuracy.
- The frontend will need to be implemented in such a way that form and function are considered. A goal for this is swiftness.
- Once the DB and webpage are complete, the ML model can then be connected and then begins the testing.

Testing

- Since we are only receiving 6 months of data, one consideration is the use of only 4-5 months in the database to simulate generated data. One thing to watch for is tweaking the ML model to output expected data instead of wanted data.
- To curb any need overordering, one concept we will consider is under performing in the prediction. This should keep costs down and still have the capabilities of being tweaked through the user page for final ordering.

Documentation

- Once everything is functioning, we will then compile a formal document that outlines how the functions operate. This will include expected input/output and possible tweaks.
- These documents will include the PHP, HTML, JavaScript, Python, and SQL code and queries that make the model operational.

4.5 Schedule

Tasks	**Dates**
1. Understand the given data	1. 11/9-11/28
2. Research ML	2. 12/20-1/24
3. Design Webpage, DB, ML	3. 1/18-1/24
4. Implement Webpage, DB	4. 1/24-2/18
a. Implement Initial ML model b. Refine model c. Use current data to confirm viability	a.1/24-2/7 b.2/7-2/28 c.2/28-3/7
5. Test DB with Webpage	5. 2/10-2/24
6. Test ML with DB and Webpage	6. 3/1-3/14
7. Documentation	7. 3/25-4/28

4.6 Deliverables

- Design Document: Contains a listing of each major software component and any diagrams.
    - Early Database design
    - The evolution of the ML model
    - Design of the web front
- Database schema and initial data.
    - The schema is for a SQL database.
    - Initial data is six months of data provided by Walmart for ML model building.
- Web site code: including frontend and connected backend/database interface code.
    - These will include the PHP, HTML, and JavaScript code for the webpages.
- Machine Learning code and analysis.
    - The language of choice for our ML model will be in python.
- Final Report
    - A detailed analysis and breakdown of the ML model, how queries are made to the DB, and any problems that arose.

5.0 Key Personnel

Kyle Orman – Orman is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Foundations I and II, Programming Paradigms and Software Engineering which are relevant to the software design aspects of this project. His professional experience as a broadcast engineer will be useful for creative problem solving and logistics tasks. During this project he will be responsible for machine learning research and implementation.

Josh Thornburgh – Thornburgh is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed the following relevant courses: Database Management Systems, Computer Security, Cryptography, Information Retrieval, Programming Paradigms, Algorithms, Software Engineering, and Programming Foundations I and II. During this project he will be responsible for machine learning research and implementation.

Abigail Tee – Tee is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed Programming Foundations I and II, Programming Paradigms, Database Management Systems, and Software Engineering. During this project she will be responsible for front-end development.

George Romano – Romano is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses such as Programming Foundations I and II, Programming Paradigms, Database Management Systems, Software Engineering, Mobile Programming, etc. He will be working on front-end development for the project.

Margaret Turner – Turner is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant courses. Relevant courses she has completed are as follows: Database Management Systems, Programming Paradigms, Information Retrieval, and Software Engineering. She worked an Information Technology internship the summer of 2021 at Texas AirSystems. She will be working on the front-end and back-end development.

Dipika Mohapatra – Our industry point of contact. She works for Walmart as a software developer.

Aneshkumar Tadi and Prasoon Anand –Walmart technical architect and Walmart tech lead in Bangalore respectively. They will assist our project by providing technical support and the data we will use to populate our database.

6.0 Facilities and Equipment

The scope of this project is primarily focused on software design and implementation. We will not need particular facilities and equipment with the exception of personal devices to research and code with. We will use Heroku or a similar service to host our web application with a PostgreSQL database containing our historical supply order data. Everyone in the group must have access to a computer with an internet connection so they can participate in the research, design, and coding of the project. We will be using database containing real world supply chain data and ML model developed in python to produce the predictive algorithm that will be utilized to forecast business supply orders for Walmart store locations. We will utilize the task management software Trello to assign individual tasks and keep to our proposed schedule. No particular physical location or facility is required to achieve the goals of this project.

7.0 References

[1] AI, Remi. “Artificial Intelligence for Inventory Management.” *Medium*, Medium, 25 Sept. 2019, https://medium.com/@RemiStudios/artificial-intelligence-for-inventory-management-c8a9c0c2a694.

[2] Chan, Melanie. “Using Machine Learning in Inventory Management.” *Unleashed Software*, 20 Apr. 2021, https://www.unleashedsoftware.com/blog/using-machine-learning-inventory-management.

[3] Gaspar, By: Huba, et al. “10 Product Recommendation Techniques to Improve UX & Conversions.” *CXL*, 25 Sept. 2020, https://cxl.com/blog/product-recommendations/.

[4] Tambe, Prof. Dr. “Review of Inventory Management System for Warehouse.” *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, no. 6, 2019, pp. 1912–1915., https://doi.org/10.22214/ijraset.2019.6320.