

# Efficient Grocery Shopping Using Geolocation and Data Mining

Myrella Aliaga-Vasquez  
*Universidad Peruana de Ciencias Aplicadas (UPC)*  
 Lima, Perú  
 u201512857@upc.edu.pe

Reyna Bramon-Ayllon  
*Universidad Peruana de Ciencias Aplicadas (UPC)*  
 Lima, Perú  
 u201718779@upc.edu.pe

Willy Ugarte  
*Universidad Peruana de Ciencias Aplicadas (UPC)*  
 Lima, Perú  
 willy.ugarte@upc.pe

**Abstract**—In the current pandemic, people are looking to leave their houses less frequently to prevent getting infected, but the absence of an app that shows the necessary information before going to the supermarket forces people to look in different supermarkets for the products they want to buy, thus increasing their chances of catching the virus, not to mention the waste of money and time. DoremyS is an app that allows you to create shopping lists that indicate to the user which supermarket to visit to find every product in them; it uses Geolocation to recommend supermarkets that are near the user and Data Mining to recommend shopping lists based on the user's interests.

## I. INTRODUCTION

Due to the number of different supermarkets in the city, the price difference between them and the variety of products that they offer, customers need to travel from supermarket to supermarket to find the products they want to buy, whether it is because they cannot find them<sup>1</sup> or because they want to look for more suitable prices before making any purchase.

However, because of the pandemic, people don't want to leave the safety of their homes and expose themselves to the virus unless it's necessary. Thus, an application that not only shows the products available in each supermarket but compares the prices between them and shows the customer which supermarket close to them offers everything they need is a necessity.

Currently, there are similar international applications in other countries that offer customers ways to find the products they need to buy. One popular example is Radarprice in Spain that allows customers to scan the products' barcodes and compare the prices of them in different shops in the country<sup>2</sup>. However, the reviews of the app claim that it doesn't use Geolocation to know where you are for showing the products in Madrid; furthermore, it is not accurate because the prices are entered by the customers. Another example is Aisle411<sup>3</sup>, an American app that allows buyers to know the aisle in the store where they can find the product, but it only works in some stores in the United States, and you must manually enter the

store you are visiting. On top of that, there are not apps like these in South America, or any app that Peruvian citizens can use for this purpose.

Most of these apps are no longer available because they failed to meet the customers' expectations: they complained about the size of the database, the layout and design. In addition to these, none of the apps currently in the market nor the obsolete apps use Data Mining to make use of their clients' interaction with the app.

Furthermore, with most of the existing solutions, the customer needs to look for each product individually and compare the prices manually after finding the supermarkets that offer them; moreover, the customer has to look for their desired products one by one, which isn't a problem when you're only thinking of buying 5 or less products, but in the current context, people are looking to grocery shop for a week or even a month, and this proves to be inconvenient.

In this work, we develop a mobile application called DoremyS that considers not only price comparison, but also where to find all the products in a shopping list. In addition, our biggest differentiation from other solutions is the use of Data Mining to predict the user's shopping behavior with the help of a supervised algorithm.

The solution's system has two components: the smartphone interface app and the backend server module. Users (customers) can search for products. The server handles the backend process and the storage of the database, including the storage of the users' accounts.

Our search engine allows users to search for products based on their location via global positioning system (GPS). It also allows them to look for products in a specific supermarket, or in a particular brand/size and/or category. The results are shown in a map that allows the user to choose the supermarket closest to them.

Furthermore, many similar solutions get complaints about the size of their databases: users can't find the products they're looking for either because the app does not have the information of many supermarkets in the city or because they don't update their database. To solve this problem, the solution will also allow supermarkets' administrators to track items, so they can update the prices and delete items that are not available anymore.

<sup>1</sup>"Survey: Nearly all consumers frustrated when shopping in-store"

<sup>2</sup>Radarprice

<sup>3</sup>Aisle411 Shops

Our contributions are summarized as follow:

- We develop a mobile application that allows users to search for products of different categories.
- We consider the fact that users sometimes look for an extensive list of products and use a location-based supermarket search to determine which supermarket close to them offers the products they are looking for.
- We use data mining to recommend products to the user based on the frequency they look for these.

This paper is organized has follows. In Section II, we talk about other similar solutions. In Section III, the most important concepts are explained and the general architecture of the solution, as well as a detailed breakdown of how it works. In Section IV, we explain the results of the experiments along with the accuracy of the algorithm used and finally the conclusions.

## II. RELATED WORKS

In [1], the authors proposed a solution called Shop-Lister that allows users to create grocery shopping lists and find nearby supermarkets that offer them, as well as recommended items they can add to their lists. They proposed using Geolocation and Data Mining to recommend the items based on the brands in the lists. However, unlike our solution, it does not take into consideration the activity of the user within the app and recommends the items at the same rate to everyone, and it was prototyped only for Sri Lankan supermarkets.

In [2], the authors created “The Smart Shopping List”, an app that uses Geolocation and Data Mining with the objective of helping users find products in grocery stores and create shopping lists. They proposed using Association Rule Mining to recommend items based on the existing products in the lists. They also offer a function to share lists between users. The Smart Shopping List was also created only for Sri Lankan supermarkets, and it also lacks analysis of the users’ activity within the app.

In [3], the authors develop a mobile app for tourists that shows them personalized suggestions for their trips based on their location. This allows them to arrange their trips more efficiently and plan their visits to historical places in advance. The system combines hybrid filtering technology with the ant colony optimization (ACO) algorithm to make more efficient customized tourism recommendations [3]. Our solution differentiates from theirs because in their model users cannot create a list with the attractions shown in the app, they have to answer some questions and the app recommends them different itineraries with different trips, they choose one and it shows them the route for the selected trip. With our solution users can create their own lists freely and even edit the ones the app makes for them.

In [4], the authors use the Google Maps API to show patients the nearest health care facility in a walking distance using a composite algorithm that consists of three algorithms namely, Information Retrieval R-Tree (IR2) algorithm for filtering process, K-Nearest Neighbour (KNN) query Technique for path and distance verification and Ahuja-Dijkstra

algorithm to find optimized path. Their application reads the user’s location in real time to calculate the estimated time to cover the distance to the nearest IR facility using only Google Maps API [4]. Therefore, they do not apply any data mining technique. In [5], the authors created an application that reads the user’s location in real-time to calculate the estimated time a distance to the nearest IR facility using only Google Maps API. The main difference with our proposal is that they do not use Data Mining for recommendations based on user’s behavior, since their only focus is to show the patients the closest place to them, and they can achieve this only by knowing their current location and the location of these facilities.

In [6], the authors develop a mobile application that proposes personalized paths for their users, for example, if a user indicates that they use a wheelchair, the system recommends a path that avoids the architectural barriers that could present to said user [6]. Our main differentiation from this solution is that in their app the users have to enter, for example, that they use a wheelchair in order to get personalized paths, whereas in our app these features can be learned just by the use of the app while searching and browsing.

## III. MATERIAL AND METHOD

In this section of the paper, we will take a look at the definition of the main concepts of the project. Knowing these concepts will help having a better understanding of the proposed solution and the way it works.

### A. Preliminary Concepts

**Definition 1** (Geolocation). *A way of knowing the geographical location of something or someone automatically from certain coordinates. It refers to the location of the place on the map automatically, considering the digital cartography, taking the coordinates as references.*

This term has become quite famous with the evolution of mobile phones, since they have GPS technology and with the help of satellites they can provide us with an exact location anywhere in the world [7]. The most relevant georeferencing technologies are the following [8], [9]:

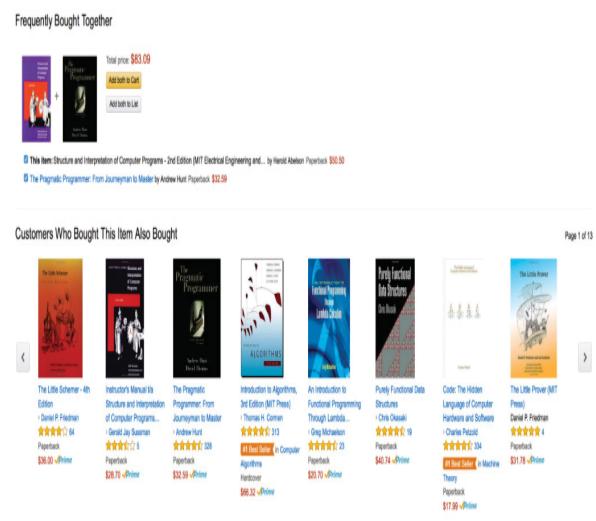
- GPS. By using the GPS satellite network.
- Wireless Wi-Fi networks.
- Mobile networks.
- IP address.

**Example 1** (National Society for the Prevention of Cruelty to Children (NSPCC)). *The NSPCC is a United Kingdom based charity that aims “to end cruelty to children”. They use the Google Maps API to allow their users to locate themselves on the map as well as other people that are raising funds for the charity as pictured in Fig. 1a.*

**Definition 2** (Data Mining [10]). *Data mining refers to the act of analyzing data from various databases using Machine Learning and Machine Learning methods. The collected data can be pre-processed to turn it into meaningful information.*



(a) NSPCC Map by Google.



(b) Amazon recommendation system.

Fig. 1. Examples of Geolocalisation and Recommendation systems

*Helping to analyze the data in various dimensions. It also classifies and identifies the existing relationships that were determined. It is the method of identifying models along with various parameters in large relational databases. Data mining is mainly used in daily activities that have a strong influence on retail or wholesale [10].*

**Example 2** (Amazon). Many E-commerce companies use Data Mining to recommend products to their customers based on their interaction with their sites. Amazon is one of the most famous examples of this with their “People who viewed that product, also liked this” recommendation system, as shown in Fig. 1b. Like mentioned before, Amazon uses data gathered from their users while they browse to build and fine-tune its recommendation engine<sup>4</sup>. This benefits Amazon because the better they know their users, the better they can persuade them to buy similar products that they believe they might like.

**Definition 3** (Web Scraping [11]). Web scraping is a relatively new method of collecting data online. The term describes the automated process of accessing websites and downloading specific information, such as prices, for each one. By allowing the creation of large, customized data sets at low cost, web scraping is already applied for scientific and commercial purposes in many areas, such as marketing, industrial organizations or inflation measurement [12], [11].

**Example 3** (E-commerce). Web scraping is a useful tool to build marketing strategies for all businesses. Specifically in e-commerce, most of commerce require to know the prices that their competitors are offering in order to sell their products at a competitive price<sup>5</sup>. Web scraping allows them to extract all of this useful information in a simple and automated way.

<sup>4</sup>Amazon: Using Big Data to understand customers<sup>5</sup>Top 18 Web Scraper / Crawler Applications & Use Cases in 2021

**Definition 4** (Instance-based learning [13]). This type of learning is based on storing training examples and when classifying a new object, the most similar objects are extracted (the nearest neighbors) and their classification is used to classify the new object. This is a supervised classification method [13]. This algorithm is also known as K-NN (K - Nearest Neighbors). This algorithm works to predict by ranking the similarities between the items that the user interacts with most frequently. This means that it takes that object and looks for the closest similarities, which it takes and classifies to recommend it to the user [14].

**Example 4.** In Fig. 2, we can see instance X has more close neighbors of class "a" than class "b", so these are the data that we are going to obtain as a result of the execution of this algorithm [15].

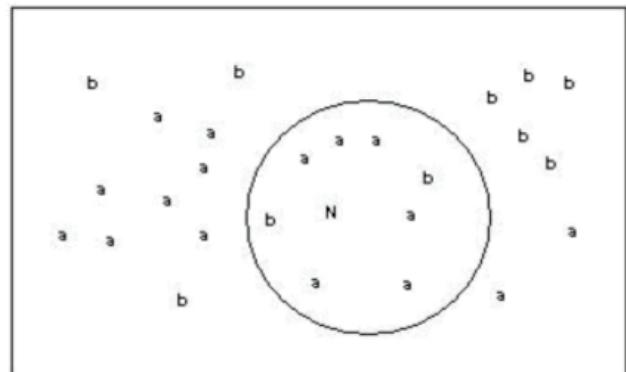


Fig. 2. Instance-based learning

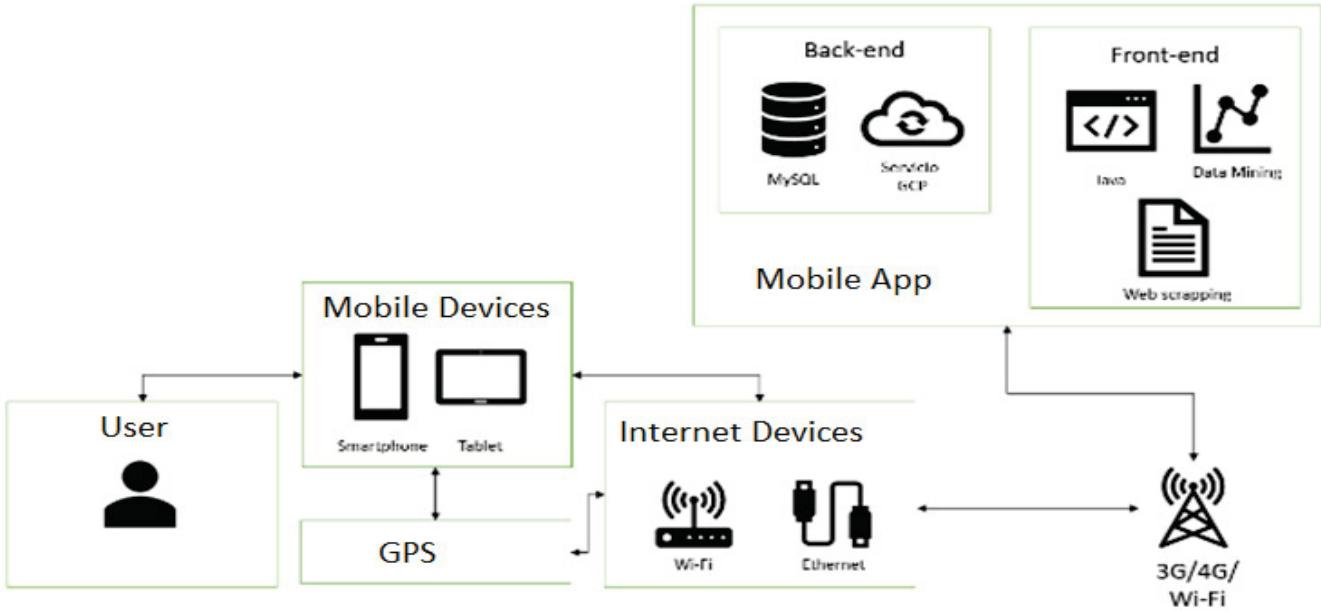


Fig. 3. Physical Architecture

### B. Proposed Approach

In this section we'll explain the components of the physical and logical architectures of our solution, as well as a detailed breakdown of how the solution works.

1) *Physical Architecture*: Fig. 3 shows the physical architecture of the proposed solution, the customer uses a smartphone or a tablet that has an internet connection thanks to Wi-Fi or an Ethernet cable and at the same time gets connected to Wi-Fi/3G/4G to show the app to the customer. As it can be seen, in the backend of the solution is everything related to the database, such as the MySQL server and Google Cloud Platform; the front-end includes the Java language, Data Mining and web scraping.

2) *Logical Architecture*: This architecture has three layers: the business layer, the application layer and the technology layer. The Business Layer allows us to model the operational organization of an enterprise<sup>6</sup>. Product Search starts with the user searching for a product, they apply filters to their search, then they select the product and compare prices from the different supermarkets that offer the same product to finally select the one that they want to buy from and see the route to it. Shopping List Making starts with the selection a product by clicking “Add it to a shopping list”, then the quantity of the products can be edited or deleted. Favorite List Making works in a similar way, in this case the users do not have to create the list because it’s created by extracting the most common purchased items from their historical data. Finally, Recommended List Making starts when the system analyzes the searches the users make and the frequency in order to recognize their favorite products and then recommend similar products to them.

<sup>6</sup>ArchiMate 3 - “8. Business Layer: ArchiMate R○ 3.1 Specification”

The Application Layer allows us to model the structure, behavior, and interaction of the applications of the enterprise<sup>7</sup>. It also shows that it stores every product search made by the user as well as the lists previously mentioned.

The Technology Layer allows us to model the structure and behavior of the technology infrastructure of the enterprise<sup>8</sup>. As shown, it is required to use a smartphone or a tablet to use the solution. There are three layers in the middle node: the Presentation one that includes Google Cloud Platform; the Logical one that includes Android Studio and the Data one that includes the MySQL server. These layers get connected to the DoremyS interface thanks to Google Cloud Platform. A backup was also incorporated for the solution that is stored in Google Cloud Platform.

3) *Recommended List Module*: The algorithm that we propose in our project to make the recommendations of the shopping lists is given as follows:

- Step 1: The user interacts with the products.
- Step 2: Each of these interactions is given a score in the range of 0 to 1, meaning that the more interactions the score will be the higher.
- Step 3: The products with the highest score are detected (the minimum score is 4 to make recommendations) and the most similar<sup>9</sup> products are searched (the Nearest Neighbors) and added to the recommended list.
- Step 4: The application will send notifications every time the list is updated.

4) *How it works*: Now, we explain step by step how the solution works: the user first needs to login with their account

<sup>7</sup>ArchiMate 2 - “10. Technology Layer: ArchiMate R○ 3.1 Specification”

<sup>8</sup>ArchiMate 1 - “9. Application Layer: ArchiMate R○ 3.1 Specification”

<sup>9</sup>Similarity is measured with Euclidean distance.

(a) Homepage of the solution.

(b) Searching filters.

(c) Details of a product.

(d) No results.

(e) “Compare prices” feature.

(f) Supermarkets displayed in the map.

(g) Details of a shopping list created by the user.

(h) Supermarket that offers the products of the list displayed in the map.

(i) Recommended list.

Fig. 4. Overview of the App screens

before using our app. Then, the user will be redirected to the homepage shown in Figure 4a, where all the products from every Peruvian supermarket will be displayed thanks to the web scrapping tool called ParseHub, for extracting all the

product's information from the supermarket's web pages. The users can also use keywords and other filter information for searching (Figure 4b). The searching results will be displayed, and the user can choose between viewing the details of the

product (Fig. 4c) or adding it to a shopping list. However, if the app finds no match for the input keywords, it will display a message that lets the users know the product is not available, as shown in Fig. 4d.

As shown in Fig. 4c, there is a feature that allows users to compare the prices of the same product in different supermarkets so they can choose the best price. By clicking on this “Compare prices” button it shows the different supermarkets that offer the product and how much more or less they charge for them, as shown in Fig. 4e. Furthermore, the user can click on the button on the bottom right labelled “Map” and see which supermarket close to them has the product they searched for. Given that we’re actively using Geolocation, the app will always suggest the nearest supermarket to the user as the first option, and the rest will be listed in order of proximity. Fig. 4f shows the location of the user and the location of the supermarkets as well as the fastest route to each one of them. As mentioned before, users can add products to a shopping list they create, these lists show the amount of products in them and the total cost of the products, as shown in Fig. 4g. Moreover, if they click on the “Maps” button they are shown which supermarket close to them has every product of said list, as shown in Fig. 4h.

Our bigger differentiation from similar solutions is our recommendation system. It works by analyzing the user behavior from their previous searches and the shopping lists they create to learn about the products they are more interested in or the ones that they buy the most based on the frequency of the searches. With all that information, the system creates a personalized list that contains products that the user buys frequently and other products that it thinks they could be interested in. This list is 100% customizable, which means the user can delete items from the list, add more to it and edit the amount of the items. It also shows the amount of products and the total cost of the list and the supermarket that offers all of them as shown in Fig. 4i.

#### IV. RESULTS AND DISCUSSIONS

Now, we explain how we validated the functionality of our proposal, the development environment and the data collection.

##### A. Experimental protocol

The application is developed in Android Studio using the Java language. The MySQL database and the PHP connection files are hosted on a GCP (Google Cloud Platform) server. The image used contains LAMP (Linux, Apache, MySQL and PHP) and a Debian operating system. For authentication we use the firebase service that offers various login methods that are quite simple to implement.

##### B. Results

We conducted two acceptance tests with 50 users who used a smartphone with the Android operating system. For data capture, Google Forms was used on these two occasions, before performing the tests and after, with the aim of obtaining

various data, including if they exceed their budgets, the effectiveness of their purchases and feedback on the interface. The indicators defined for the validation process are the following:

- 1) **Average purchase time:** The average purchase time is calculated using the following formula:

$$APT = 1 - \frac{PTS}{TPT} \quad (1)$$

Where PTS stands for Purchase Time using the Solution and TPT stands for Traditional Purchase Time.

According to Alania, in 2018 users took an average of 40 minutes to make their purchases<sup>10</sup>, with our solution users took 30.5 minutes, that is, the time purchase decreased by 23.75%.

- 2) **Excess in budget:** In Fig. 5a we can see that 95% (with 40% stating that they sometimes exceed their budget, 35% stating they exceeded it many times and 20% stating they exceed it almost every time) of users spent more than budgeted on more than one occasion when they went grocery shopping in a traditional way, after using the application we can see that 100% of users declared that they spent the exact amount the app showed them. This means the solution helped them to spend less than their budget for their grocery shopping.
- 3) **Purchase effectiveness:** In Fig. 5b we can see that 95% (with 60% stating that they have left an important product out many times, 25% stating they have few times and 10% stating they always do it) of the users did not buy a necessary product from their shopping list because it was out of stock or because of the high price. After using the application we can see that 100% of the users found all of the products they were looking for, since the app showed them that they were available in their supermarket of choice. This means the purchase effectiveness increased.
- 4) **User satisfaction:** In Fig. 5c we can see that 25% of users are satisfied and 75% very satisfied with the usability of DoremyS. In the case of the interface, 40% are satisfied and 55% are very satisfied, with only 5% neutral about it. In the case of efficiency, 25% are satisfied and 75% very satisfied. Finally, 95% are willing to use the solution if it were to be launched on the market (with 55% very willing and 40% willing), with only 5% neutral about it.

For this indicator, we validated the proposed supervised algorithm (Instance-based learning or K-NN) with three experiments simulating the activity of three users with different activity levels. The first user was barely active during the week, used the solution 10 minutes once a week because he was only looking for one specific product; the second user was slightly more active, used the solution 25 minutes twice a week and the third user was really active, used the solution 20 minutes each two days. The way it works is that it populates

<sup>10</sup>“Aplicació n de la teoría de colas en la atención de clientes en los cajeros de supermercados vivanda tienda de Benavides – Lima” (in Spanish)

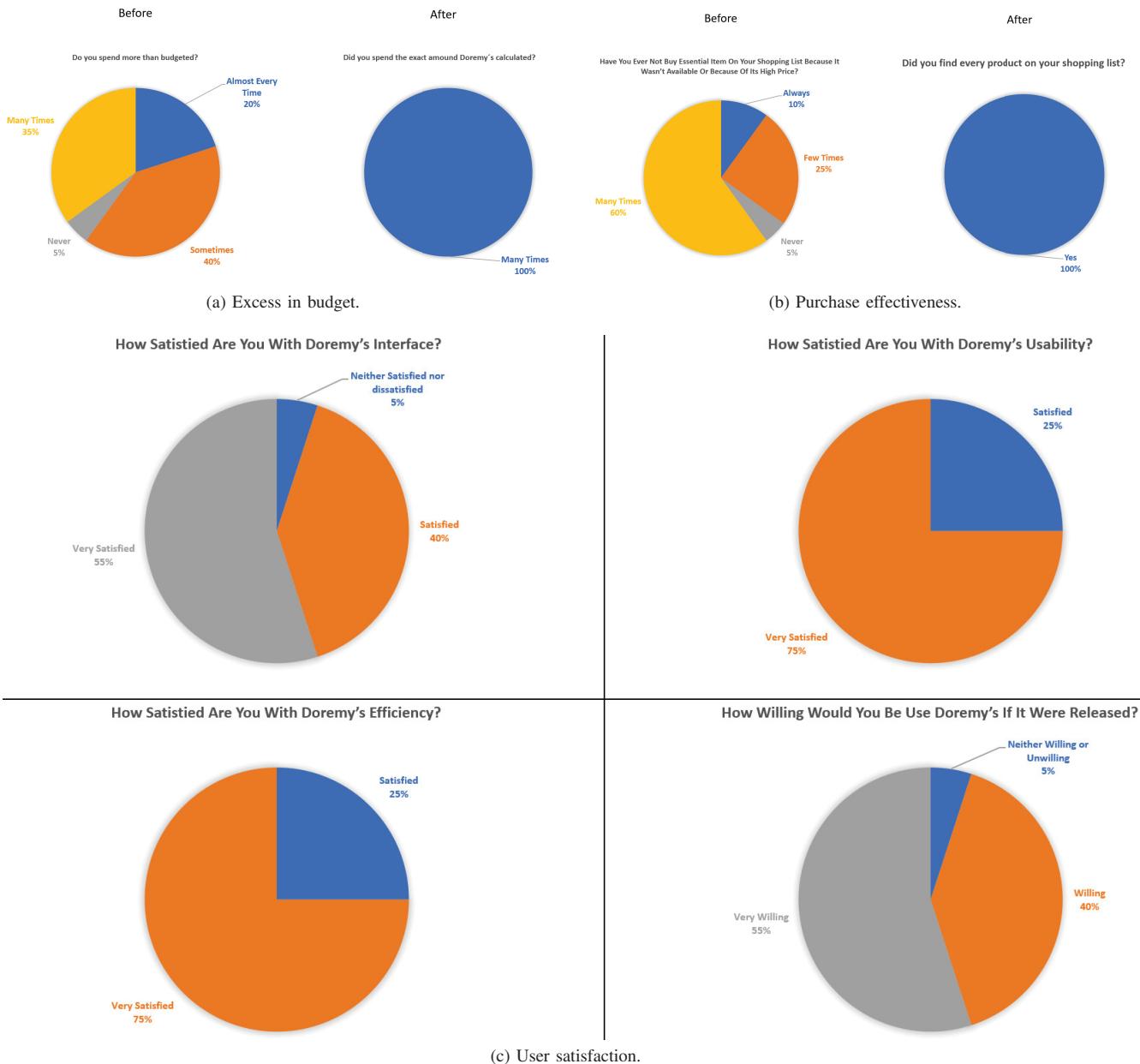


Fig. 5. Comparison of results

a recommended list every time the users uses the app and it takes the objects the user frequently interacts with and uses them to find more similar objects. For example, if an user always buys milk, the recommended list will always have milk and, additionally, cheese because it's a dairy product. The results are shown in Table Ia.

- 5) **Category variety:** For this indicator, we conducted three experiments simulating the activity of three users with different activity levels. The results are shown in Table Ia.

The first column refers to each of the users in the

experiment, the second one to the minutes they spend using the app and the last column refers to the number of categories they get on their recommended list. DoremyS currently works with 5 categories: Breakfast, Groceries, Fruits & Vegetables, Milk and Meats & Chicken. As it can be seen, the more active the user is the more categories he gets in his Recommended List. Supervised algorithms used what they learned from relationships found in previous data sets to predict the output values for new ones. Since the recommended list is always being populated, the algorithm is constantly

TABLE I. RESULTS AND CONFUSION MATRIX

User	Minutes/Week	Number of Categories
Barely Active User	10	1
Slightly More Active User	50	2
Extremely Active User	80	5

(a) Results of the experiment.

		Predicted Results		
		Class 1	Class 2	Class 3
<i>Actual Results</i>	Class 1	9	2	2
	Class 2	1	7	0
	Class 3	0	3	11
		10	12	13

(b) Confusion Matrix.

improving and making more accurate predictions from previous lists.

Each one of these results show that the proposed solution improved grocery shopping for all of the users in the following ways: 1) it allowed them to save time since it took them a shorter amount of time to purchase, 2) it allowed them to buy every product they saw necessary, and 3) it allowed not to exceed their budget.

*Accuracy of the algorithm:* We used a 3x3 confusion matrix to test the accuracy of the algorithm used. A Confusion Matrix is a visual evaluation tool used in machine learning. The columns of a Confusion Matrix represent the prediction class results, and the rows represent the actual class results. It enumerates all possible cases of a classification problem.

Furthermore, it was proved that the feature Recommended List is a useful tool for customers because it is created according to their activity levels.

Table Ib shows that along the x-axis are listed the true class results (Actual Results) and along the y-axis are the k-nearest neighbors class predictions (Predicted Results). Along the diagonal are the correct classifications or True Positives, and all the other cells are misclassifications. To find out the overall accuracy we need to use the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

For values in Table Ib,  $\text{Accuracy} = \frac{9+7+11}{9+2+2+1+7+3+11} = \frac{27}{35} = .7714$ . The overall accuracy gives us an insight on how often our classifier is right. This is the most important metric to prove our algorithm works, since it means that out of 100 times, 77.14 times the algorithm shows the user the right amount of categories in their Recommended Lists according to the activity within the app. It proves that for an extremely active user, the Recommended List would have more categories than the one of a barely active user. This helps prove that the app will show the user an accurate list that accommodates to their likes the vast majority of times.

## V. CONCLUSIONS

With our solution we achieved a 23.75% reduction in the average shopping time, as well as 95% of users that are willing to use the application if it is released to the market in the future. These metrics are important because a reduction in the average time shows that the application serves its purpose of making grocery shopping more efficient and the results also show that it has a high acceptance from the users. Furthermore, we conducted an experiment to prove that the Recommended List feature works according to each user, which means that every user gets a list that works for them based on their activity levels. Finally, we proved that the algorithm has an overall accuracy of 77.14%. These results show that the solution helps users to stay within their budget and not leave any important product out of their purchase.

It is important to note that due to the pandemic there have been some limitations with the validations. First, we could not hold a face-to-face focus group and instead had to hold it via Google Meets. In addition, we had to do this validation with a smaller group that we had planned, with only 20 people. One of the unanswered questions that could be explored in the future is if the solution helps users save money and the amount saved, since the app shows users how much they can save with individual products thanks to the “Compare prices” feature implemented, but the results only show that they payed exactly what the app told them they were going to pay, it doesn’t show how much money they save with each purchase.

Lastly, one way we would like to expand the project in the future is by adding more premises for each supermarket. As of now, we only have one per supermarket and only Tottus in Lima has more than 31 premises. This would help many more users to find their products in a fast and efficient way. Furthermore, store the data with blockchain technology [16], [17] or gather more information with chatbots [18].

## REFERENCES

- [1] A. Firoz and G. Ratnayaka, “Shoplister – a grocery list management application,” in *ICIP*, 2020.
- [2] W. A. Harsha Jayawilal and S. Premeratne, “The smart shopping list: An effective mobile solution for grocery list-creation process,” in *IEEE (MICC)*, 2017.

- [3] C. Chen and J. Tsai, "Determinants of behavioral intention to use the personalized location-based mobile tourism application: An empirical study by integrating TAM with ISSM," *Future Gener. Comput. Syst.*, vol. 96, 2019.
- [4] R. Ganesan and V. V. Chamundeeswari, "Composite algorithm for pervasive healthcare system - a solution to find optimized route for closest available health care facilities," *Multim. Tools Appl.*, vol. 79, 2020.
- [5] J. E. Mandel, L. Morel-Ovalle, F. E. Boas, E. Ziv, H. Yarmohammadi, A. Deipolyi, H. R. Mohabir, and J. P. Erinjeri, "Optimizing travel time to outpatient interventional radiology procedures in a multi-site hospital system using a google maps application," *J. Digit. Imaging*, vol. 31, no. 5, 2018.
- [6] D. Fogli, A. Arenghi, and F. Gentilin, "A universal design approach to wayfinding and navigation," *Multim. Tools Appl.*, vol. 79, 2020.
- [7] "Geolocation," in *Encyclopedia of GIS*. Springer, 2017, p. 694.
- [8] Z. Rashid, C. S. Mon, and R. Kolandaisamy, "Proposing a development of geolocation mobile application for airport pickup of international students pickup," in *ACM ICSCA*, 2019.
- [9] N. D. Priandani, H. Tolle, A. G. Hapsani, and L. Fanani, "Malang historical tourism guide mobile application based on geolocation," in *ACM ICSCA*, 2017.
- [10] R. Kaur and S. Singh, "A survey of data mining and social network analysis based anomaly detection techniques," *Egyptian Informatics Journal*, vol. 17, 2016.
- [11] H. M. Kienle, D. M. Germán, and H. A. Müller, "Legal concerns of web site reverse engineering," in *IEEE WSE*, 2004.
- [12] A. Cavallo and R. Rigobon, "The billion prices project: Using online prices for measurement and research," *The Journal of Economic Perspectives*, vol. 30, 2016.
- [13] M. Haddad, V. K. Ghassab, F. Najar, and N. Bouguila, "Instance-based learning for human action recognition," in *IEEE SMC*, 2020.
- [14] S. G. K. Patro, B. K. Mishra, S. K. Panda, R. Kumar, H. V. Long, D. Taniar, and I. Priyadarshini, "A hybrid action-related k-nearest neighbour (HAR-KNN) approach for recommendation systems," *IEEE Access*, vol. 8, 2020.
- [15] Y. Ruan, Y. Xiao, Z. Hao, and B. Liu, "A nearest-neighbor search model for distance metric learning," *Inf. Sci.*, vol. 552, 2021.
- [16] J. J. Cueva-Sánchez, A. J. Coyco-Ordemar, and W. Ugarte, "A blockchain-based technological solution to ensure data transparency of the wood supply chain," in *2020 IEEE ANDESCON*, 2020, pp. 1-6.
- [17] J. C. Arroyo-Mariños, K. M. Mejia-Valle, and W. Ugarte, "Technological model for the protection of genetic information using blockchain technology in the private health sector," in *ICT4AWE*. SCITEPRESS, 2021, pp. 171-178.
- [18] J. M. Solis-Quispe, K. M. Quico-Cautí, and W. Ugarte, "Chatbot to simplify customer interaction in e-commerce channels of retail companies," in *ICITS (1)*, ser. Advances in Intelligent Systems and Computing, vol. 1330. Springer, 2021, pp. 561-570.