EASY BUY

Annapurna Ananya Annadatha, Indhu Priya Reddem, Manasa Bobba, Monica Lakshmi Mandapati
Computer Engineering Department
San José State University (SJSU)
San José, CA, USA
Email: {annapurnaananya.annadatha, indhupriya.reddem, manasa.bobba, monicalakshmi.mandapati}@sjsu.edu

***Abstract*—The COVID-19 pandemic has accelerated the growth of e-commerce websites, making them the fastest growing business in the world. However, to stay competitive, it is essential to provide customers with an exceptional user experience. This project proposes building a full stack web application that integrates a machine learning model to recommend products based on customer preferences. By using content-based and collaborative filtering techniques, the model considers similar products and similar users' interests to provide a list of top recommendations based on product ratings. Customers can rate and review products, and can also receive notifications when their interested product is back in stock. To better understand their customers' needs and likes, retailers can use a dashboard to view the velocity of their products and receive insights into their inventory management. A marketing metric will be calculated to help retailers align their inventory with market demand, thus helping them gain profit. By using collaborative filtering, content-based filtering, and inventory management, this project aims to create a unique and exceptional e-commerce experience for both customers and retailers. This project aims to leverage machine learning and inventory management techniques to improve the e-commerce experience for both customers and retailers. The proposed web application integrates recommendation systems, product reviews, and inventory management to help retailers make data-driven decisions and improve their profit margins. By enhancing the user experience, this project hopes to contribute to the growth of the e-commerce industry.**

***Index Terms*—content based; collaborative filtering; e-commerce; inventory management; machine learning; marketing metric**

I. Introduction

Electronic commerce (E-commerce) has revolutionized the way businesses are conducted, making it possible to trade products and services online. E-commerce relies on information technology, particularly Electronic Data Interchange (EDI), to enable businesses to conduct transactions with customers over the internet. In an E-commerce website, the vendor directly sells products or services to customers through a digital shopping cart or basket system, which allows payment through credit cards, debit cards, or Electronic Fund Transfer (EFT) payments. The internet and smartphones have become an integral part of modern life, and with the widespread internet penetration, E-commerce has grown rapidly. Customers can buy products and services from a website by selecting the listed products online. However, the availability of numerous E-commerce websites makes it difficult for customers to select the best product, leading to confusion and a loss of interest. Therefore, recommendation systems play a crucial role in providing customers with the best user experience.

Recommendation systems yield accurate results and save customers time by making their shopping experience quicker. This paper explores different recommendation techniques that can provide accurate recommendations, such as collaborative filtering and content-based filtering. We aim to integrate these machine learning integration's into the full stack web application. Furthermore, digital platforms are helping businesses to better connect with their customers, making the supply chain leaner and smarter. This, in turn, reduces waste and supports green businesses. In this paper, we propose an E-commerce system that provides accurate recommendations to customers while reducing waste and supporting green businesses. This paper aims to explore different recommendation techniques and integrating them to provide accurate recommendations to customers in an E-commerce system. With the widespread internet penetration and the availability of smartphones, E-commerce has become an important tool for shopping, learning, communicating, and obtaining services. By reducing waste and supporting green businesses, we hope to contribute to the growth of the E-commerce industry while providing the best user experience to customers and retailers.

II. Problem Statement / Project Architecture

An E-commerce site can make a customer buy any product at any point of time and place. The purpose of the application is to examine the personal interests of users and recommend the products that are relevant to their interests. There will be many options available in different stores for a single product which can make the customer confused to select the best one of it. Our project makes sure the site suggests the products which are of similar interest to the customer and suggests the best reviewed items in the product. Easy Buy is an e-commerce web application which recommends products by considering customer preferences and based on reviews and ratings given by other customers.

The Architecture diagram as shown in Fig. 1. illustrates the high-level architecture of our project, which includes four key components: Front-end, Back-end, Recommendation system, and Database system. Each of these components plays an essential role in the overall functionality of the application.

The Front-end component is built using ReactJS, a popular open-source JavaScript library for building user interfaces. ReactJS allows for the creation of dynamic and interactive web pages, providing a rich user experience. Also used React Bootstrap for styling the frontend application. The Front-end will be deployed on AWS EC2, a scalable and reliable cloud computing service that allows for easy deployment and management of web applications. Integrated AWS S3 to upload the product images into it.

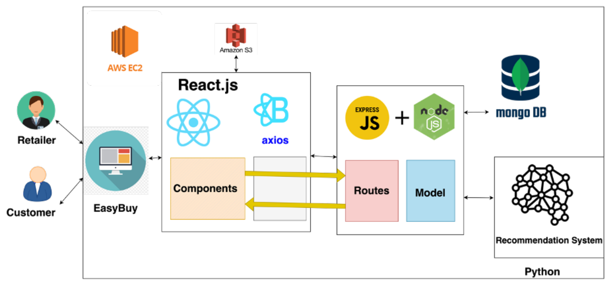


Fig. 1. Project Architecture.

The Back-end component of the application is developed using NodeJS, a powerful JavaScript run-time environment. NodeJS provides an efficient and scalable server-side platform for building web applications. The Back-end will also be deployed on AWS EC2, ensuring that the application can handle a high volume of traffic and remain responsive even during periods of peak usage.

The Recommendation system component of the application will be developed using Python. Recommendation systems rely heavily on data processing to identify patterns and make predictions. Python has a rich set of libraries, such as Pandas, NumPy, and Scikit-learn, that are specifically designed for data manipulation and analysis. These libraries make it easy to preprocess and clean large datasets, which is essential in building accurate recommendation models.

The Python recommendation system and backend are integrated using the Node spawn child processes. The childProcess.spawn() method spawns the child process asynchronously, without blocking the Node.js event loop. This method fetches the recommendation results into the backend Node application.

Finally, the Database system component of the application will be built using MongoDB, a popular NoSQL document-based database. MongoDB is known for its ability to handle large volumes of unstructured data and its scalability. The database will be used to store the data generated by the application, including user preferences and activity data.

III. Method(s) / System Design

The application consists of two types of users: Customers and Retailers.

User authentication: The user should authenticate by providing his details in the Login/Sign-Up page to access the services. JWT Web tokens are used to securely verify the user.

Product search: Upon landing on the Dashboard, the customer gets to search for the product based on the name, category and retailer. The searched products can be filtered based on price, ratings and can be sorting based on latest arrivals and price range.

Cart Management: Customer can add the products to cart and continue shopping. You can update, add and remove products from the cart.

Order Management: Customer can place the order after entering the shipping details, payment information and gets to view the summary of products present in the order. The Retailer can see the order details once placed and can deliver the order.

Product Management: The Retailer can Add/Update and Delete the Products. The customer can show his interest if the product is out of stock. So that he will receive the email notification once the product is stocked up. We are using Mailgun service to send the notification to the enrolled users.

Product Insights: The product insights are displayed on the Retailer Dashboard. We show the top 5 selling products this week and also the product velocity for the retailer to better understand his inventory management. Dashboard has a sales velocity chart which shows the details of sales of the categories based on the price range, this gives the retailer a better understanding of the market. A sales chart of the retailer is displayed to understand his profits and loss.

Product Recommendations: We show the top Retailers information and product recommendations on the Customer Dashboard. Recommendation systems are an essential component of many platforms and services, as they help users discover relevant and personalized content. These systems utilize sophisticated algorithms to filter and organize vast amounts of information and provide recommendations based on individual user preferences, interests, and behaviors. There are three main types of recommendation systems: content-based, collaborative filtering, hybrid recommendation systems and we have incorporated both content-based and collaborative filtering approaches to make predictions.

Content Based Filtering: Content-based filtering algorithms use information about the attributes of the items being recommended (such as product descriptions) to make recommendations that are like items that the user has already liked. In the below figure, the user purchases a white brand1 shirt, so the content based system recommends white shirt and a brand1 shirt.

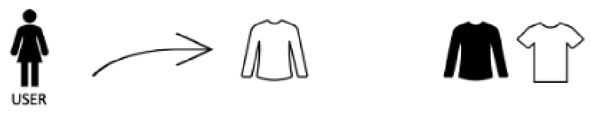


Fig. 2. Content based filtering.

To implement content-based filtering,

1) We utilized the description field of our product data. By employing TF-IDF (Term Frequency-Inverse Document Frequency) scoring, we can determine the relevance of different products to individual customers based on their previous orders.

$$\mathbf{W}_{(}x,y) = \mathbf{tf}_{(}x,y) * \log\left(\frac{N}{df_x}\right) \tag{1}$$

where,

$\mathbf{W}_{(}x,y)$ is the term x within document y,
$\mathbf{tf}_{(}x,y)$ is frequency of x in y,
df_x is number of documents containing x,
N is total number of documents

2) We determined and calculated the cosine similarity, then arranged them according to their similarity based on the input item. This scoring system allows us to generate personalized recommendations by suggesting products that align closely with the customer's interests and preferences.

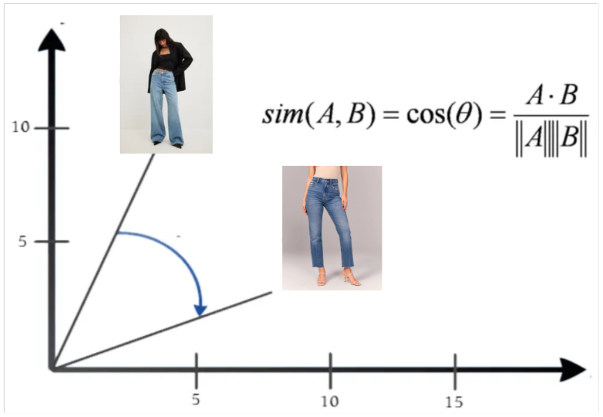


Fig. 3. Cosine Similarity.

Collaborative Filtering: The idea behind collaborative filtering is that if two users have similar preferences for certain items, then it is likely that they will have similar preferences for other items as well.

To implement collaborative filtering, we leverage the ratings data collected from our web application, which includes information such as the userID, productID, and corresponding ratings. Our approach involves using the KNNWithMeans algorithm, which allows us to achieve effective recommendations through collaborative filtering. The KNNWithMeans algorithm operates by identifying the k most similar users or items to a particular target user or item. It achieves this by comparing their rating patterns and preferences. Based on the average rating or score of these nearest neighbors, the algorithm generates recommendations for the target user or item. This information enables us to make recommendations that align closely with their interests, thereby enhancing their overall user experience on our web application.

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u,v) \cdot (r_{vi} - \mu_v)}{\sum_{v \in N_i^k(u)} \text{sim}(u,v)} \tag{2}$$

where,

k (int) – The (max) number of neighbors to take into account for aggregation.

min_k (int) – The minimum number of neighbors to take into account for aggregation.

$sim_options$ (dict) – A dictionary of options for the similarity measure...

verbose (bool) – Whether to print trace messages of bias estimation, similarity, etc. Default is True.

In the below figure, both User1 and User2 have similar interests, so the system recommends User1, product (c). Recommender systems face various challenges, and one of the primary hurdles is the cold start problem. The cold start problem occurs when there is insufficient information about users or products in the system, making it difficult to provide accurate recommendations. This issue poses a significant challenge, particularly for collaborative filtering-based recommender systems that heavily rely on historical user data to make predictions.

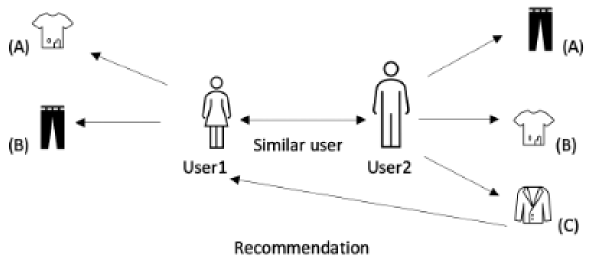


Fig. 4. Collaborative filtering.

IV. Evaluation Methodology / Materials

Our research problem focuses on examining customer satisfaction, retailer satisfaction, and recommendation accuracy. To address this problem, we have developed a comprehensive full-stack web application that prioritizes user-friendliness, ease of operation, accessibility, and adaptability. The primary goal is to ensure that users have a positive shopping experience on our platform.

The recommendation system within our web application is a crucial aspect that we have thoroughly evaluated using various algorithms. We have assessed the performance of different recommendation algorithms and determined the one with the highest average rating to be the most effective for our purposes. This selection process allows us to identify the best algorithm to provide accurate and relevant recommendations to our users.

To assess the overall effectiveness of our web application, we have established specific metrics to evaluate different aspects. These metrics include product fetching, cart management, successful order placement, and overall performance. By considering these metrics, we can gauge the success of our web application in meeting the objectives we have set.

Content-based filtering is a technique that makes recommendations based on the characteristics of the items themselves. The dataset consists of product names, and a TF-IDF matrix was created to represent the textual information. The dataset was split into training and testing sets using an 80:20 ratio, with a random state of 42. Pairwise cosine similarities were computed between items in the training and testing sets. The accuracy of the model was calculated as the percentage of accurate predictions based on the top similar items and the accuracy is 93.75%.

Collaborative filtering is a technique used in recommendation systems that makes recommendations based on the similarities and differences between users. The dataset was split into training and test sets using a 80:20 ratio. Different parameter combinations for the KNNWithMeans algorithm were tested, optimizing for the lowest RMSE. The best parameter pearson was selected based on lowest RMSE obtained on the test set. Cross-validation was performed on the entire dataset using the best parameters to estimate the average RMSE and the RMSE value is 1.44.

To gather data for training our recommendation model, we extensively reviewed reputable sources such as papers, presentations, blogs, videos, and websites, with a focus on recommendation systems. Most of these sources were obtained from the IEEE website, ensuring access to relevant and reliable information. To acquire the necessary dataset, we searched through Kaggle, a platform renowned for hosting diverse and high-quality datasets. After careful exploration, we selected the Amazon Review Data from a GitHub link as the most suitable dataset for our recommendation system. This dataset specifically includes ratings related to Amazon fashion, aligning perfectly with the recommendations we aim to provide on our website.

V. Results

The application contains different functionalities like SignIn, SignUp, search product, add product to the cart, add products to the inventory, send notification to the customers, place order.

Customer Dashboard: This is the landing page for EasyBuy Application. It has search bar, categories to select products from and information about top retailers. It has "You may also like" and "Customers also brought" sections. This two sections gets the recommendations from python.

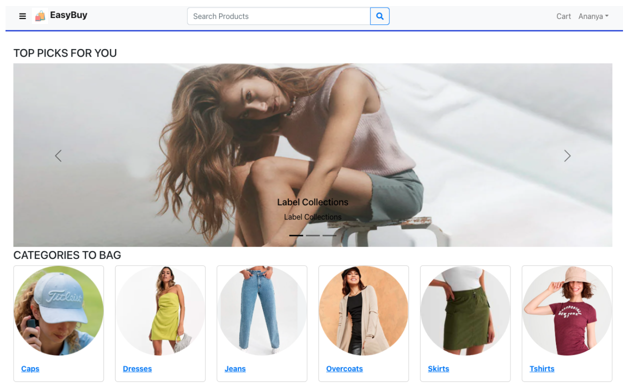


Fig. 5. Customer Dashboard.

Search Page: The Search Product page allows users to search for products based on keywords or filters and Sort them. Users can enter a keyword in the search box or use filters to narrow down their search results. Customer can filter based on categories, price range and average customer reviews. They can also sort based on Newest Arrivals, low to high, vice-versa and Avg. customer reviews.

Product Page: The page displays the information of the product such as name, retailer details, price, availabilty, ratings and reviews. It also provides the feasibility for the user to add his rating and review of the product.

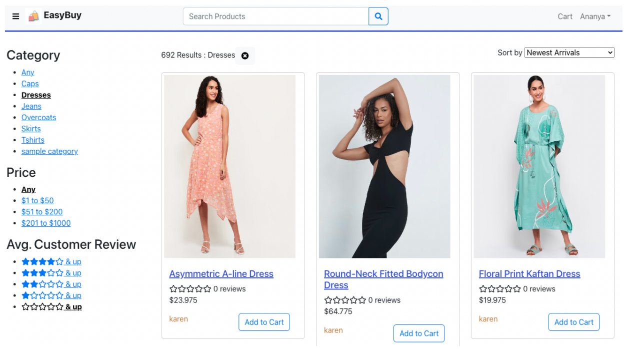
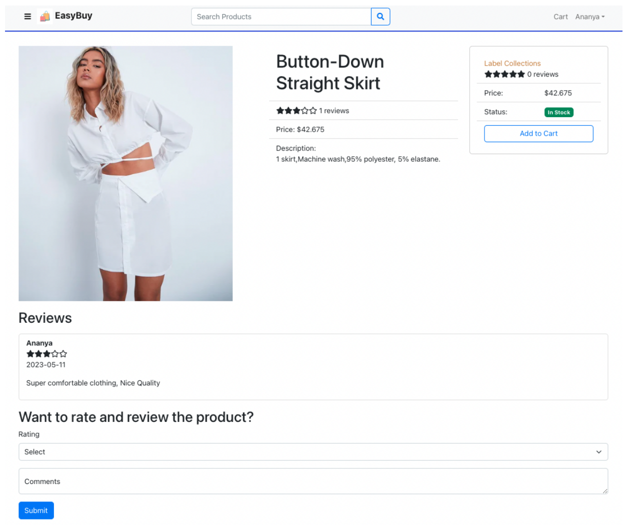


Fig. 6. Search.



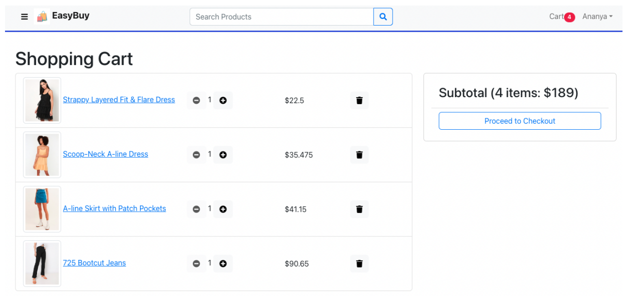
Description:
1 skirt,Machine wash,95% polyester, 5% elastane.

2023-05-11

Super comfortable clothing, Nice Quality

Fig. 7. Product.

Cart Page: The Fig.7 shows the cart screen where users can manage the products that are added to cart. User can add, update and delete items from cart.



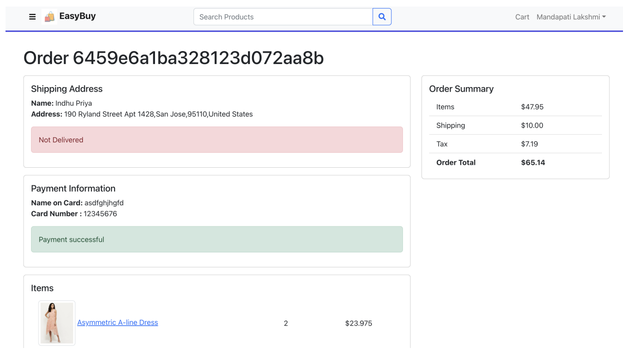
Strappy Layered Fit & Flare Dress	1	$22.5	
Scoop-Neck A-line Dress	1	$35.475	
A-line Skirt with Patch Pockets	1	$41.15	
725 Bootcut Jeans	1	$90.65	

Subtotal (4 items: $189)

Fig. 8. Cart.

Order Summary Screen: The Fig.8 shows the summary of the products, shipping address, payment information and total price of the order. It also shows the payment status and delivery status of the order.

Recommendations: Upon placing the order, the python files are run and content-based, collaborative filtering recommendations are generated. This recommendations will be fetched onto the customer dashboard.



EasyBuy | Search Products | Cart | Mandapati Lakshmi

Order 6459e6a1ba328123d072aa8b

Shipping Address

Name: Indhu Priya
Address: 190 Ryland Street Apt 1428,San Jose,95110,United States

Not Delivered

Payment Information

Name on Card: asdfghjhgfd
Card Number : 12345676

Payment successful

Items

Asymmetric A-line Dress 2 $23.975

Order Summary

Items	$47.95
Shipping	$10.00
Tax	$7.19
Order Total	**$65.14**

Fig. 9. Order Summary.

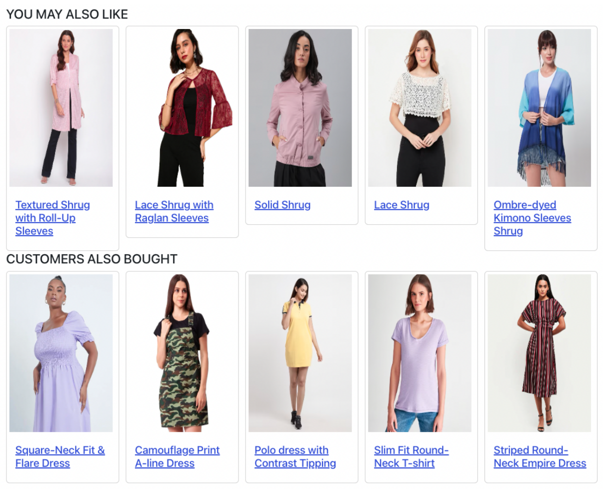
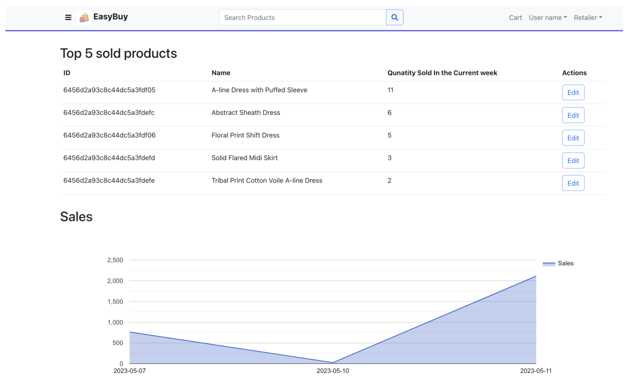

YOU MAY ALSO LIKE
Textured Shrug with Roll-Up Sleeves
Lace Shrug with Raglan Sleeves
Solid Shrug
Lace Shrug
Ombre-dyed Kimono Sleeves Shrug
CUSTOMERS ALSO BOUGHT
Square-Neck Fit & Flare Dress
Camouflage Print A-line Dress
Polo dress with Contrast Tipping
Slim Fit Round-Neck T-shirt
Striped Round-Neck Empire Dress

Fig. 10. Recommendations.

Retailer Dashboard: The Retailer Dashboard shows product and sales insights to the retailer. The first section shows the top 5 selling products of the current week along with its velocity. The second section shows the sales chart velocity which gives current market insights.



EasyBuy | Search Products | Cart | User name | Retailer

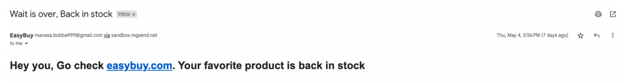
Top 5 sold products

ID	Name	Qunatity Sold In the Current week	Actions
6456d2a93c8c44dc5a3fdf05	A-line Dress with Puffed Sleeve	11	Edit
6456d2a93c8c44dc5a3fdefc	Abstract Sheath Dress	6	Edit
6456d2a93c8c44dc5a3fdf06	Floral Print Shift Dress	5	Edit
6456d2a93c8c44dc5a3fdefd	Solid Flared Midi Skirt	3	Edit
6456d2a93c8c44dc5a3fdefe	Tribal Print Cotton Voile A-line Dress	2	Edit

Sales

Fig. 11. Retailer Dashboard.

Availabilty Email: The customer recieved the email when his favorited product is back in stock. The below image shows the sample email a customer receives.



Wait is over, Back in stock Inbox ×

EasyBuy manasa.bobba999@gmail.com via sandbox.mgsend.net
to me

Thu, May 4, 5:56 PM (7 days ago)

Hey you, Go check easybuy.com. Your favorite product is back in stock

Fig. 12. Email Notification.

Important API's: All the API's are tested and are performing successfully as expected.

Type	API	Inputs
POST	/signUp	User details
GET	/signIn	Username, Password
POST	/addProduct	Product details
PUT	/uplaodImage	Image file inserted into S3
PUT	/updateProduct	updated Product details
POST	/deleteProduct	Product Id
POST	/addReview	User rating, review
POST	/createOrder	Order details
PUT	/deliverOrder	order ID
DELETE	/deleteOrder	order ID

VI. Discussion

In this study, we have focused on the significance of recommendation systems in the context of E-commerce.Our research explored various recommendation techniques, with a specific emphasis on collaborative filtering and content-based filtering.In the process of integrating the TensorFlow.js model with the Node.js application, we encountered several significant difficulties, particularly during the initial phases. To begin the integration process, we developed a Keras model and saved it as an h5 file, a common format for storing Keras models. However, the transition from the h5 format to a TensorFlow.js model proved to be problematic. We utilized the TensorFlow.js converter to convert the h5 file into a format compatible with TensorFlow.js. Subsequently, we loaded the converted model into the Node.js application using JavaScript, following standard practices for integrating TensorFlow.js models. we faced problems while executing the model, which led to inaccurate predictions and could not gain meaningful insights from the model output.

Future research can extend our current system by incorporating advanced techniques, such as deep learning for image and text analysis, to provide more accurate product recommendations. Additionally, integrating a sentiment analysis module can help retailers gauge customer sentiment and further enhance their products and services. These potential advancements would strengthen the capabilities of our system and contribute to ongoing developments in the e-commerce domain. In addition to our current approach, we can incorporate customer input based on their interactions with the application as a valuable enhancement. By tracking user behavior, such as the number of clicks, we can gather additional data to further enhance the recommender system's capabilities. This approach allows us to gather more nuanced insights into customer preferences and tailor recommendations to their individual needs. By integrating user interaction data into the recommender system, we can capture valuable signals about customer interests and preferences. For example, if a customer frequently clicks on products from a specific category or shows a preference for certain features, we can leverage this information to refine and personalize their recommendations. This not only enhances the accuracy of the recommendations but also enables us to provide more meaningful insights to the customers.

VII. Related Works

Recommendation system is an essential part of this project. It helps both the consumers and the retailers. It helps buyers to find products much more easily based on the recommendations. There are different types of recommender systems such as item-to-item, attribute-based.If the system requires minimal input or little user action it is considered automatic. Recommendations are personalized based on the user's interests such as preferred color, movie genre and music group [1], [2], [3]. These recommendations are straightforward because they need minimal effort to produce and are considered automatic because no user input is required. Since they can be used by a range of people, these recommendations are not merely temporary. In recommendations that are generated based on qualities, items can be described using a variety of features and attributes. Since the user must specifically look for a specific kind of product to use as the basis for the recommendations, this method is regarded as manual. Depending on how long the system keeps user choices in memory, these recommendations may be temporary or permanent. Item-to-item correlation recommenders make recommendations for items based on other items the user has expressed interest in. On e-commerce websites, where new products are recommended depending on what customers have in their shopping baskets, these suggestions are regularly found. These suggestions are manual since the user must have a full shopping basket, and they expire quickly because the user does not. In this system, association rules are most frequently applied. The people-to-people correlation algorithm identifies similarities between the active user and other system users and suggests things that other consumers have previously bought or evaluated. The best method in the system is collaborative filtering. This technique is very manual because it depends on people having previously bought or rated things. Depending on the system's design, the recommendations might continue [1], [2]. [4]

Based on the presence of other products, association rules are utilized to recommend products. When two things are purchased simultaneously, the inclusion of one item in a transaction can be utilized to determine that the inclusion of the second product is also present. When recommending products to new users who want to buy them, this is incredibly helpful [1], [3], [5].

Collaborative filtering is the most successful technology used in recommender systems and it is the most widely used on the internet. Customer information, aggregated reviews, and ratings from all consumers are used in the collaborative filtering approach to provide suggestions. The advantage of this strategy is that it evaluates current, active consumers who share the same tastes and traits as the current client to develop recommendations. To determine the suggestions, the collaborative filtering model, which is heuristic-based or memory-based, considers rating information, whether the product has been purchased or not, and the amount of time spent viewing the products [1], [3], [6], [7].

Content-based filtering is based on the ability to analyze products and find similarities with active users to recommend products. This method, unlike collaborative filtering or association rules, does not require an active database of purchase history. It is based on data retrieval, analysis, and filtering. This method is most used in places where content can be read or analyzed, such as news articles, movies, and anything with metadata attached. It makes recommendations based on items the user has previously viewed. Nearest neighbor or clustering algorithms can be used to recommend other articles to the active user based on these labels and user preferences. However, this method is challenged by new users who have limited information and a limited number of labels. Commonly used algorithms include k-nearest neighbor, clustering, Bayesian, and artificial neural networks [1], [3].

Hybrid solutions have been proposed to not have problems that exist in both the above filtering systems. Solutions include implementing both filtering systems separately and combining the results, incorporating characteristics of content-based filtering to collaborative filtering systems, and developing new algorithms that incorporate the techniques of both systems. Combining different recommender system approaches entails developing two distinct recommender systems based on collaborative and content-based approaches. The recommendations can be generated separately and then combined linearly [1], [8].

VIII. Conclusions

We have achieved developing a Full Stack application integrated with Machine Learning recommendations, providing customers and retailers with an unparalleled shopping experience.. Our team utilized React Bootstrap to ensure aesthetically pleasing styling and a user-friendly interface, resulting in a shopping experience that is smooth and seamless for customers. Our application includes a comprehensive product management system, intuitive cart management, and efficient order management, all with the added value of personalized recommendations for each individual user. In addition, we have incorporated features such as email notifications and S3 image storage, adding further convenience and value to the application.

Our application also provides insightful product analytics, aiding retailers in managing their inventory efficiently. Retailers are able to make informed decisions based on detailed product insights, enabling them to optimize their inventory and enhance their overall business operations. The integration of Machine Learning recommendations and our robust feature set creates an unparalleled shopping experience, improving the lives of our users and enhancing the success of retailers.

Acknowledgment

We would like to thank Professor Dr. Wencen Wu for her invaluable guidance, support, and feedback throughout the duration of this research project. Her expertise and knowledge in the field have played a pivotal role in shaping our ideas and enhancing the overall quality of our work. Additionally, we would like to express our gratitude to San Jose State University for providing us with the necessary resources and facilities to conduct this research.

References

[1] S. Sivapalan, A. Sadeghian, H. Rahnama, and A. M. Madni, "Recommender systems in e-commerce," *2014 World Automation Congress (WAC)*, pp. 179–184, 2014.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[3] P. R. S. Namitha Ann Regi, "A survey on recommendation techniques in e-commerce," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY*, vol. 2, December 2013.

[4] T. Sugata and C. Yang, "Leaf app: Leaf recognition with deep convolutional neural networks," *IOP Conference Series: Materials Science and Engineering*, vol. 273, p. 012004, 11 2017.

[5] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM Conference on Electronic Commerce*, ser. EC '99. New York, NY, USA: Association for Computing Machinery, 1999, p. 158–166. [Online]. Available: https://doi.org/10.1145/336992.337035

[6] E. O. a. Wilson, "Biophilia," *Cambridge (Mass.) ; London : Harvard University Press*, 1984.

[7] B. C. Wolverton, A. Johnson, and K. Bounds, "Interior landscape plants for indoor air pollution abatement," *NTRS- NASA Techincal Reports Server*, 1989.

[8] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Prior ratings: A new information source for recommender systems in e-commerce," in *Proceedings of the 7th ACM Conference on Recommender Systems*, ser. RecSys '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 383–386. [Online]. Available: https://doi.org/10.1145/2507157.2507200