Hybrid Recommendation Model using Markov Chains and Advanced Feature Integration

Anurag Rajput[1], Sonu Airen[2], K.K. Sharma[3]

[1]M.Tech. Research Scholar, [2]Assistant Professor, [3]Associate Professor
[1, 2, 3]Department of Information Technology
Shri G.S. Institute of Technology and Science, Indore, M.P., India

Abstract

The need for personalized product recommendations in online grocery shopping has been further emphasized by the evolution of e-commerce platforms in recent years. Traditional recommendation paradigms, including collaborative filtering and content-based filtering, encounter various challenges, such as, data sparsity, cold-start problems, and sequential dependencies, which are crucial to consider while modeling repetitive and temporal shopping activities. To tackle these limitations, in this study, we propose a hybrid recommendation model dedicated to next-basket prediction tasks.

The approach employs Markov Chains to capture sequential and historical dependencies while leveraging features like TF-IDF; product popularity, recency, and order-specific preferences are additional influences on prediction accuracy. The hybrid modelling incorporates probabilistic techniques with contextual information to provide more personalized and relevant recommendations.

We used Next-Basket Relevance Score, next-basket coverage, and Harmonic Basket Prediction Score as metrics to evaluate the performance. Our findings show that the Markov Chain hybrid model with additional features outperforms the baseline models. To be more specific Markov + Popularity + Recency got 0.165 Next-Basket Relevance Score, 1 next-basket coverage and 0.2833 Harmonic Basket Prediction Score, which indicates a model which can deal with sparse and sequential data effectively over Markov only (Next-Basket Relevance Score: 0.095 Harmonic Basket Prediction Score: 0.1735)

This work underlines the power of being able to combine sequential modelling with context features in order to build solid recommender systems. Through this, the findings have practical implications for e-commerce platforms looking to improve customer retention and satisfaction with accurate and scalable recommendations.

Keywords: Machine Learning, Markov Chains, Next-Basket Prediction, Recommendation Systems, TF-IDF Weighting

I. INTROUCTION

In the last decade, technology has advanced exponentially, and one of the industries that beheld dramatic change is the world of e-commerce and online grocery shopping. As services like Instacart, which enables consumers to order groceries on the Internet from various stores, emerged, the possibility of leveraging data to understand as much as possible about consumers — and their shopping excursion — expanded even further. This process of predicting customer preferences through machine learning analytics, helps them understand customer metrics, such that they make meaningful recommendations based on relevant browser

data. The primary goal of next basket prediction is to predict the next basket of items a customer is expected to purchase in his grocery shopping sessions, and such techniques can be incredibly useful in this related domain as well. This improves user experience dramatically, but such recommendations are directly connected to sales and retention in the very competitive world of online groceries.

For most e-commerce platforms, the success is feature-driven around creating a personalized, engaging, and seamless experience for its users. However, in the traditional recommendation systems, they were only based on collaborative filtering that predicted the interest of a new or existing user based solely on the behavior of other users by relying on the similarity with other users in a new or existing user. This research is particularly interested in the area of grocery shopping, because although collaborative filtering performs well in most 4 areas, it has serious limitations in e-commerce. In fact, grocery shopping consists of regular, repeat purchases where consumers tend to select a combination of both more staple and occasional items. Traditional collaborative filtering methods struggle to capture such complex and dynamic behavior. Additionally, it also suffers from cold-start problems (new users or new products have inadequate data to generate good recommendations) and sparsity (users do not have enough in common to make meaningful patterns).

Thus, the researchers have investigated different mechanisms, such as content-based filtering that considers the features of the user and the item. Because the method creates user profiles based on products they interacted with, it proposes products that are similar to the ones a user quotes. However, content-based filtering does not effectively consider the temporal/sequential aspects of consumer behavior, such as predicting a user's subsequent purchases based on their previous purchase history, or considering the time factor between purchased products. This is especially crucial for predicting the next basket specifically in grocery shopping, where a customer could buy items in a certain order, and the prediction becomes suddenly more precise once we learn the order of purchases.

Recently the interest in hybrid recommendation systems has grown to overcome these problems by combining different techniques to utilize the advantages provided by each method. Markov Chains is a common approach to this problem: it states that given certain probabilities, it is possible to determine if the users carried out a particular action sequence (e.g. buy a particular product). Markov Chains can model sequential dependencies in a way that traditional recommendation systems cannot, as they capture the transitions between products a user is likely to purchase. For instance, they can be used to predict what a user is likely to buy next, based on their previous behavior — this is particularly effective when there are many purchases of the same item, and users such as in-store purchases in grocery shopping. It is especially helpful for next basket prediction where the model needs to predict the precise assortment of products a user will add to their next basket based on previous purchases.

Markov Chains have gained significant popularity for a variety of recommendation tasks in the e-commerce field, primarily because they very well capture probabilistic dependencies between items. One can also use some other methods to model such sequences or Markov Chains. These models are based purely on historical data and do not take into account other aspects like preference of products, or recency of purchases, or are not able to be user-specific (i.e., these systems will provide recommendations to all users at the same time). The path that researchers must explore is the meetings of Markov chain models and the new functions, through which the ability of Markov chain models can be improved. For example, as a feature, if one uses the popularity of the objects then the model would understand which products were more relevant to customers and the recency feature would explain when a customer after an interval became interested to buy a product again which he has already purchased. Such improvements are essential for making next basket prediction systems more aligned with user preferences.

Another common aspect of recommendation systems is using some user-based features, for example: the number of unique products this user has purchased, or how often a specific purchase was made, resulting in more fine-grained predictions for this user. The key challenge in designing this kind of hybrid model is to balance these disparate elements so that the information content in the single elements is predictive enough of the outcome without introducing any noise or bias.

The evaluation of the performance of recommendation systems is another very important and challenging branches of research in modern recommendation systems. Evaluation metrics for recommendation models often consist of precision, recall and F1–score among others, which provide a more general perspective concerning the quality of the recommendations. In general, precision measures the number of relevant products among the recommended ones (i.e., how relevant the prediction results are), recall measures the ability of the model to retrieve all relevant products (i.e., how good the model is), and F1 combines both precision and recall to a single informative measure. Such metrics aid in interpreting the real-life usability of a recommendation algorithm, particularly in the ecommerce domain as the generation of product recommendations that expand the funnel of sales is the focus of business.

In brief, this dissertation research is positioned in the intersection of machine learning, data science, and e-commerce research fields. This addresses some of the constraints and advancements achieved via traditional recommendation systems and therefore aims to enhance the degree of accuracy and relevance of predictions by using a hybrid approach which integrates Markov Chain models along with other features such as recency, popularity, and user-based characteristics. This research mainly focuses on next basket prediction, which is one of the core problems, at hand in online grocery that seeks to predict the exact or accurate product that a user may buy next. In doing so, this study can contribute to the dynamic field of personalized recommendation systems in online grocery shopping, which faces a complex set of challenges due to the dynamic and multi-faceted nature of consumer behavior.

The demand for personalized experience in the Online Grocery Shopping sector today has revealed the limitations of traditional recommendation systems. Traditional models such as collaborative filtering are at odds with grocery shopping patterns that are strikingly both identifiable and random. Mechanisms of these classical systems have their buggers (e.g., cold-start problem, insufficient modeling of the sequential dependencies between products bought, and similarity of user's recommend/recommendation content etc.).

Grocery shopping is a complicated space — people buy some things over and over again, but also tend to buy new or seasonal items on impulse. This creates a very high demand for stronger systems, especially for a next basket prediction, essentially the process in which it needs to know what type of products a user is likely to buy next based on sequences. They need to capture not only normal purchasing dynamics, but also sudden, once-in-a-lifetime changes in buying behavior. In addition, these existing models do not take into account product popularity, recency of purchases, user preferences or any other external information that have a significant effect on buying behavior.

One drawback with prior work is that they came up with a hybrid recommendation system that considers sequential dependencies (using Markov Chains) in addition to a number of other features like product popularity, recency, and user specific behavior to make recommendations. However, such methods are insufficient to accurately forecast the next basket of items by the user. This research work aims to develop a new framework that will utilize different wide and deep learning architectures to improve the performance of our model. Next basket prediction is an incredibly important challenge in a traditional grocery store, and our proposed approach aims to be both scalable and effective for online grocery solutions, further emphasizes that recommendations should dynamically be made per user and user's context. By tackling this issue, the

research seeks to improve the overall user experience for end-users and drive successful business outcomes for e-commerce platforms.

With the Transmutation of E-commerce platforms, human shopping dynamics have radically shifted, and personalized product recommendations have become the prime determinant in having an upper hand. As user bases expand and catalogs become ever larger, e-commerce companies must strive to deliver customized experiences — a tall order — for every user. The motivation for this research stems from the challenges present within recommendation systems and the opportunities within online grocery shopping that address with next basket prediction.

Now, this can come under the lens of recommendations on grocery items and thus a part of modern-day retail which increasingly happens online and thus platforms keep on improving the way they recommend items to their consumers. A great recommender system allows people to shop better by presenting products that they need or will likely enjoy, leading them to return as customers. The grocery industry, known for high-frequency, repeat purchases already, represents a challenge on its own. Consumers tend to buy groceries, such as staples, in a predictable manner but make impulsive purchases of less predictable items. In such domains, traditional recommendation algorithms, such as collaborative filtering, have frequently been found lacking—for example, owing to the cold-start problem or the inability to appropriately model the sequential variation of user actions, a requirement for predicting the next basket.

The ambitious goal of these research approaches is to enhance prediction performance of recommendation systems based on these next basket prediction techniques by incorporating various factors that influence sequential dependencies over product purchases as well as product popularity, recency and user features which can further amplify the predictions. Like grocery shopping, users often have consistent purchase decisions for core products, but they can also have varying tastes over time based on external features including marketing promotions, seasonality, or specific needs. Traditional recommendation systems often overlook these patterns, while ratings based recommendations can fail to capture users' actual preferences.

Lack of predictive accuracy in these models to identify the next item in a user's shopping list would result in poor recommendations and subsequently missed business revenue opportunities.

To address these gaps, this research proposes a hybrid approach that combines sequential models (e.g., Markov Chains) with recency, product popularity, and user-specific behaviors. Understanding that most of the processes happen in a probabilistic manner, this project leverages the fundamentals of Markov Chain to structure transitions across products, which will ultimately lead to the model predicting the next resulting product that a user would be interested to buy, based on their historical actions. The sequential recommendations implemented in predicting the next basket give a much-needed edge compared to conventional recommendation techniques, by retaining the temporal aspect of customer behavior: at what times the products are bought together, or how does the historical purchase pattern impact the future purchase.

This is important because one of the primary uses of such research is to learn more about the best products for the user, so to give them a more personalized shopping experience. The contextual product suggestions are thus gaining importance as e-commerce platforms are now realizing that just like whether it is a similar product or not, when and how a user is interacting with a product is equally important. So here, this methodology attempts to integrate unique purchasing tendencies amongst consumers in the presence of some contextual macro-level conditions, further that influence and mediate product choices beyond product metrics. This research answers this gap by proposing a holistic and comprehensive solution for product

recommendation that integrates different datasets and models which can enhance each other leveraging their unique strengths especially in next basket prediction.

Additionally, the practical implications of this study for firms in the online grocery sector is also a considerable motivating factor. Sales-driven recommendation systems that retain customers back to these, Big Basket, Grocers, and now Amazon Pantry they are launching, need to be focused towards sales, along with user-based experiences and recommendations. Not to forget that a well navigated platform will make customers return for more and more because they already know where to find their product(s). If using such a predictive tools, e-commerce platforms will be able to not only fetch better discoverability but also manage inventory and improve overall user experience.

This study was also motivated by, benefits of which is contributing to the general body of knowledge in machine learning and recommender systems. While the majority of literature currently available relates to general purpose recommendation systems, comparatively little has been written that specifically addresses the unique challenges present in the grocery sector, where purchase patterns are cyclic and influenced by user-initiated routines. Such research may enhance recommendation methodology by tailoring it to the complexities of grocery shopping, while producing insights of wider significance elsewhere.

Another key aspect which drives this research, is the evaluation of the model performance. Prediction accuracy is important not only for enhancing user experience, but also to measure how good the model actually is. The proposed hybrid model can provide meaningful results guaranteed by this study's data and methods. Using evaluation metrics such as Next-Basket Relevance Score, next-basket coverage and Harmonic Basket Prediction Score, this study will provide a thorough measuring of the prediction performance of the model in predicting future purchases, as well as the ability to capture the nuances of consumers at work in the e-commerce landscape.

The objective of this study is the absence of enough recommendation systems in online grocery shopping, especially with respect to next basket prediction. This study is targeted towards enhancing the recommendation systems literature by finding the downsides of conventional approaches within this domain and suggest an effective way to encompass all sequential dependencies with user-specific characteristics and contextual features. The research makes both theoretical advances and practical recommendations for e-commerce industries seeking to develop competitive advantages in a rapidly evolving environment.

II. LITERATURE REVIEW

In the last few decades, recommender systems (RS) have become significant components of several platforms, ranging from entertainment-based platforms, such as Netflix and Spotify, to e-commerce platforms, like Amazon and Instacart. Recommender systems have undergone a period of rapid development, transitioning from simple content-based filtering to more complex collaborative filtering techniques and deep learning methods that can provide users with personalized experiences. Recommender systems were initially established with collaborative filtering (CF) where the main approach was providing preference prediction based on user-item interactions， new users and items. Since then, the field has advanced to include many more techniques, such as content-based filtering (CB) and hybrid techniques, which have become essential components of today’s recommender systems.

The most widely used early approach was collaborative filtering based on the principles of “word of mouth” or social influences and the notion that users with similar tastes and behaviors will prefer similar items. This approach is based on the premise that we can use previous actions to anticipate new choices and that there are commonalities between users (or items) we can use for recommendation [1] [2]. But it does have its limitations. Cold-start problems plague collaborative filtering, as new users or new products in the

system do not possess message interaction histories, rendering it ineffective. Another problem is the sparsity problem: users generally only interact with a small subset of the available items, making it hard for the system to correctly identify similar items or users [3] [4]. This weakness was the reason they had to come up with different approaches that could last this trend.

Conversely, content-based filtering (CB) analyzes the attributes of items themselves without relying on user interaction history. CB systems utilize information regarding product attributes to recommend items with similar characteristics to the items the user has expressed interest in [5]. Although this bypasses a few of the flaws of collaborative filtering, this method is still quite limited, especially when it comes to exposing and recommending exciting new items to the user that they may not have interacted with before. Additionally, when users put together a sequence of purchases, content-based systems do not reveal the larger contextual information between items or any temporal dependencies between them.

Hybrid methods combining several recommendation strategies have arisen as a strong answer to the disadvantages of collaborative filtering and content-based filtering. Hybrid systems combine different data sources and models, allowing them to provide more robust recommendations with greater accuracy, diversity, and personalization [6]. This research work seeks to investigate hybrid systems by integrating Markov Chain models with additional features to enhance next product prediction in online grocery shopping under product popularity, recency and user preferences.

Further, key extensions to traditional recommender systems have arisen in recent years. Among them, there are knowledge-based recommender systems (KBSR), context-aware recommender systems (CASR), and demographic-based recommender systems (DBSR), each tailored to improve personalization and relevance within the recommendation process [7].

Their effectiveness has made them useful in domains where users need more niche or personalized recommendations, for instance real estate or automobile purchases [7] [8]. These systems aim to elicit clear user preferences or needs by means of direct questions or guided submissions. Knowledge-based systems also actively involve the user in the recommendation process by integrating information (including user and product information) methods for integrating in user feedback (i.e. Feedback based knowledge-based systems).

Context-aware recommender systems [9] use contextual information (location, time, device, etc.) as part of the user interaction. These approaches try to increase the relevancy of recommendation by adjusting to the user's environment which is changing in real-time. This is especially beneficial in online grocery as purchase patterns tend to vary on the basis of the hour of the day or seasonality.

Based on demographics, demographic-based recommender systems [10] cluster users with common demographic traits, including age, gender, and geographical area. By utilizing these demographic attributes, DBSR systems can address cold-start problems under the premise that users with similar demographic characteristics are prone to have similar preferences. However, this method can be more imprecise when working with broader user bases with unique preferences.

Let us now recall some recent works addressing the issue of next basket recommendation.

The authors of [11] introduced an efficient model, called Dynamic REcurrent basket Model (DREAM), based on recurrent neural networks (RNNs). One of the main advantages of DREAM is that it is not only able to learn a dynamic representation of a user’s behavior but also to take into account the global sequential characteristics between baskets [11]. However, the original DREAM model by Yu et al. was designed to perform only binary classification. For each available product, the model generates a probability score representing the probability that this product will be included in the next basket purchased by a given

customer. However, DREAM cannot provide predictions in a multi-store (i.e., multi-class) context, consisting of predicting the store where the recommended product should be purchased. Moreover, in their work, Yu et al. did not consider some important features such as product prices, availability, and weekly specials offered in local stores. This motivated us to generalize the original DREAM model to a multi-class classification task to predict both whether a given product should be included in the customer's next shopping cart and also in which store the purchase should be made.

The authors of [12] described a novel prediction method using attention-based recurrent neural networks to detect and model inter- and intra-cart relationships. The authors proposed to consider all available shopping carts of the user concerned in order to model his/her long-term preferences (inter-shopping cart relationships) [12]. At the same time, the intra-shopping cart attention model is intended to act at the level of the items in the user's most recent shopping carts to predict his/her short-term behavior and preferences. Thanks to their adaptive attention model, the authors of [12] were able to outperform state-of-the-art methods for next shopping cart recommendation. But again, their method is only applicable in a binary classification context.

The authors of [13] used the recency factor to predict a consumer's next grocery cart by applying a prediction method based on collaborative filtering in a general top-n products recommendation framework. To demonstrate the effectiveness of their approach, the authors compared it to some state-of-the-art collaborative filtering models. The method developed during the study is based on two aspects considered essential for the recommendation of a user's next shopping cart: the popularity of the products considered and the relative recency window for their purchase [13]. These two factors are associated and integrated into a collaborative filtering approach that has proven capable of competing in terms of performance with the recognized methods used for comparison.

Content-based recommendations have also been shown to be effective in the area of next-shopping cart and grocery coupon recommendation. In this context, the authors of [14] proposed a content-based filtering model based on the use of a tree for coupon recommendations. These authors conformed the coupon selection process to personalize the recommendation and thus increase the click-through rate. Using Random Forest and XGBoost classifiers, the authors of [14] were able to improve the estimated coupon click-through rate from 1.20% to 7.80% [14].

Furthermore, the authors of [15] described and tested a new statistical method based on Yandex's CatBoost model to predict whether a given customer is likely to purchase certain selected products. This study presented some major algorithmic techniques including ordered boosting which is an alternative based on permutation as well as a new algorithm dedicated to category processing [15]. The combination of these techniques allowed CatBoost to outperform other boosting implementations in terms of performance on several datasets.

The authors of [16] considered real and especially unbalanced purchase data from an e-commerce platform and then used the CatBoost model to predict whether customers will purchase certain available products or not [16]. The method proposed by [16] obtained an accuracy of 88.51% during the prediction. The proposed model was able to effectively reduce common overfitting issues with imbalanced data through symmetric trees and adopts a more scientific and interpretable algorithmic approach for categorical variables, which resulted in less information loss during model training and improved model robustness [16].

The authors of [17] proposed to use recurrent neural networks instead of collaborative filtering techniques to create a multi-period product recommendation system that is linked to an online food marketplace. The system introduced by [17] is capable of recommending products according to multiple periods over a defined time sequence. The authors of [17] showed that the proposed recommendation system performed better in

terms of accuracy and diversity in a multi-period perspective compared to systems based on collaborative filtering [17]. In addition, the proposed system was also found to be robust to users' repetitive purchasing patterns.

The authors of [18] proposed a personalized and immersive recommendation system based on an Immersive Graph Neural Network (IGNN), which aims to increase the marketability of various products, improve users' shopping experience, promote sales, and boost market growth. "Immersion" here refers to the user's total involvement in their shopping activity, through a user experience that allows them to focus on their current needs [18]. This aspect combines qualitative and quantitative factors to interpret users' psychophysiological needs in a digital space.

The authors of [18] therefore set up an immersive marketing environment using deep learning models and graph neural networks (GNNs). The results of their work suggest that such an immersive marketing approach is particularly successful in reflecting the essential attributes and characteristics of products. The proposed model and the methods it was compared to were tested on public datasets. The model developed by the authors outperformed other methods in terms of precision and recall. However, as suggested by the authors, the proposed recommendation system has not been verified in practical applications. Thus, the impact of the presented model on real users has not been evaluated.

The authors of [19] recently proposed to use both a recurrent neural network and a feed forward neural network (FFNN) that they combined with non-negative matrix factorization (NNMF) and gradient boosting trees (GBT) to build smart grocery carts for users of the CircuitPromo platform. The authors of [19] considered different variables and a reduced number of customers (compared to our study) to describe the behavior of the platform users. Their best result for the F-score was 0.37 [19]. This result was obtained when their prediction model was applied to an augmented dataset. However, in their work, the authors did not perform any partitioning analysis and did not consider different user profiles. This type of analysis turns out to be of great importance when it comes to improving the prediction performance of a user's next shopping cart. Also, the authors of [19] did not compare the results generated by their deep learning model with those provided by traditional machine learning algorithms. Such a comparison is crucial when the available dataset is relatively small. Finally, the deep learning model introduced by the authors is not personalized because the same model architecture was used for all the customers considered.

In their paper, the authors of [20] presented a collaborative filtering-based model designed to overcome the cold start problem. To achieve this, the authors proposed to calculate the weighted sum of four different variables [20]. The first of these represents the product rating obtained using Weighted Non-Negative Matrix Factorization (Weighted NNMF) following which an affinity propagation technique was applied. The other three variables considered are graph-related similarity measures based on users' metadata and purchasing habits. The authors of [20] reported that their model outperformed existing approaches based on Hit Ratio (HT) and Normalized Discounted Cumulative Gain (nDCG) as metrics.

The authors of [21] suggested several new metrics to measure the repetition/exploration ratio in customers' shopping habits to evaluate the performance of next-basket recommendation systems. They compared and analyzed the results of state-of-the-art next-basket recommendation models on three public datasets. Their study was conducted with a focus on their new metrics to help illustrate the scope and current state of research focused on this particular area and then explain the advances made by existing approaches [21]. Their work also aimed to highlight the reasons behind the advances claimed by the studied methods. The authors of [21] indicated that future research on next-basket recommendation should consider analyzing repetition and exploration behavior (i.e., novelty discovery) to gain useful insights and help design unbiased models.

The authors of [22] proposed a framework to model a user's shopping cart sequences. Their hierarchical network model, called Beacon and based on a LSTM (Long short-term memory) architecture, consists of three main components, taking as input a shopping cart sequence and a correlation matrix [22]. The first component, the shopping cart encoder, produces correlation-aware shopping cart representations after capturing intra-shopping relationships and reciprocities between products in the cart. The sequence of shopping cart representations is then used as input to a sequence encoder, the second component, to extract inter-shopping sequential associations. The output of this component is combined with the correlation matrix, and both are used by the third component, the predictor, to generate the user's next shopping cart taking into account product correlation. Thus, [22] considered the correlative dependencies between products in order to improve the representation of individual baskets as well as the more global sequence of the user's baskets.

In a recent paper, the authors of [23] conducted an in-depth study on the recurring consumption habits of users in the field of online grocery shopping. Through the analysis of transactional data from both public and proprietary sources, they analyzed the purchasing behaviors and came to the conclusion that a significant part of the performance of the recommendation systems for the next basket can be attributed to the products that users have already purchased in the past. In this context, the authors introduced a new neural network model, ReCANet, specifically designed to take into account the recurring consumption habits of users. To do this, the model uses information about the products previously purchased to predict more accurately which items will be selected by the user in his next shopping basket. The results obtained by the authors highlighted that ReCANet outperforms current next-basket recommendation models in terms of recall and nDCG. They also performed an ablation study to explain the impact of each ReCANet component on its overall performance, showing that each of its components contributes significantly to the obtained performance. Finally, they showed that a user's repeat ratio, i.e. the frequency with which a user purchases the same item again, has a direct influence on the effectiveness of their new model [23].

In another paper, the authors of [24] also investigated the real-time personalized recommendation of the user's current shopping cart in the context of online shopping, particularly in the grocery domain. They put forward another model, PerNIR, based on the neighborhood and which takes into account both the user's personal history and his current shopping cart. This approach considers the user's short-term interests, represented by the shopping cart being built, and his long-term interests, reflected by his purchasing history. Neighboring user profiles are also considered in order to capture "collaborative" shopping behavior. The results obtained by the authors of [24] show that PerNIR outperforms other approaches by a significant margin, offering gains of more than 12% in terms of success rate compared to the second best approach used which corresponds to their previous model, ReCANet [23]. The authors of [23] also focused on optimizing their new method, which is able to quickly provide recommendations in real-world situations.

Association rules analysis (ARA) is one of the data mining techniques and aims to discover interesting associations (relationships, dependencies) of data points in large data sets. The transaction data that provides input to the analysis consists of each purchase record to be obtained from a retail store (a purchase, in other words, a basket may contain a single product or thousands of products). It is encountered not only in retail applications, but also in different applications such as text mining and web click analysis, as emphasized before [25].

Market basket analysis is the most well-known application of ARA and analyzes the transactions (sales receipts) of customers from a retail store and detects products purchased together. The concept of basket here refers to the baskets or shopping carts used in shopping. Market Basket Analysis provides support in many important decision-making processes such as store layout planning, product diversification, cross-selling, shelf design, catalog design, customer segmentation, and purchasing processes. [26].

The authors of [27] used a data set consisting of a total of 962 thousand sales receipts from a supermarket in Konya for one year. With this data set, the purchasing behaviors of customers based on product categories were examined with the Apriori algorithm, and the results were examined in detail to support the marketing activities of the company. For example, it was determined that the product group related to meat products and daily foods was fresh vegetables, while the ready-to-wear group was the biscuit group.

The authors of [28] emphasized that the most appropriate facility layout planning is very important for effective product/service production and implemented an application for facility layout using ARA for a hospital. According to the result given by the Apriori algorithm, the “ambulance entrance” and “green area” rules have the highest confidence value.

To measure the effectiveness of different methods, they are compared to baseline methods. Many of these baselines are described by the authors of [29]. One naive approach, called the same as last trip approach, assumes that a customer will buy the same items as purchased during the previous impulse-shopping session. TOP: The other method predicts that the customer purchased the most frequently purchased items over all of their past purchases. While this approach is simple to implement, it is not very precise. An intermediate baseline method uses association rules. This method determines the probability of a product being bought based on the previous purchases of the customer. The probability is based on support — how often products are bought together — and confidence — how reliable the prediction is. However, this method has limitations as far as the quality of the prediction.

Market basket analysis has gained much attention in the sequential manner of product purchases in recent years. A common approach to this is through the use of Markov Chains (MC), which use a customer's past purchases to estimate the probability of future purchases [30]. The approach is to estimate the next-state of a sequence using decision-tree approximations. One important method in this domain is a collaborative filtering method that is widely used in recommender systems with product ratings from users. Non-matrix factorization (NMF), a form of collaborative filtering, is used in market basket analysis [31].

A leap is achieved, showing that if we take into account the sequential nature of the purchases and the individual preferences of each client using FPMC based on non-matrix factorization [32]. This results in an individual transition matrix for each user and thereby overcomes the problem of generating identical purchase histories where the purchase scenario of a user is identical. A recent representative approach is the Hierarchical Representation Model (HRM), which is an attempt to combine individual purchase sequences with general customer preference [33]. HRM recommends modeling both users and products as vectors, and describes a two-layer model that predicts the next purchase given past behavior. Nonetheless, HRM fails to represent information between separate purchase baskets and thus cannot predict consumer behavior. To tackle this issue, a novel method was proposed based on recurrent neural networks (RNNs) [11]. RNNs approach the human brain since with each observation multiple categories are processed. This makes them particularly well-suited for applications such as shopping lists, where products frequently occur in multiple baskets.

Another recent process in this field is the Temporal Annotated Recurring Sequences Basket Predictor (TBP) [34], which concentrates on customer-specific data. TBP utilizes multiple facets of a customer’s purchasing patterns, including product repeatability, to optimize decision-making.

The work by the authors of [35] is a strong extension of the sequential purchase analysis. Their methodology starts by calculating the similarity between market baskets of different customers followed by using similarity between customer histories. Their solution models all these components and predicts future purchase baskets, providing state-of-the-art results in next-basket prediction.

This research captures a variety of approaches for enhancing the predictive validity in market basket analysis, from basic heuristics to multi-parameter machine learning algorithms like recurrent neural networks and temporal models.

A wide variety of methods have been used to develop effective, tailored recommender systems, as seen in the literature review. Even though the basic ones like collaborative filtering, content-based filter and similarities yield effective results, novel methods such as Hybrid and deep learning and context-aware systems are being developed to make it more specific and minimize missing insights. Various advanced models, including recurrent neural networks and attention mechanisms, have been proposed to address the next-basket recommendation problem, attempting to capture short-term and long-term user preferences in online grocery shopping context. Despite promising results, many of these models also face challenges related to scalability, cold-start issues, as well as the need for incorporating external influences such as product pricing or availability. Thus, the proposed work will exploit these existing methodologies to leverage the integration of sequential input along with additional contextual parameters to propose an integrated recommendation system able to predict the groceries being bought in future. This reading material reflects the increasing sophistication of recommender systems and encourages the utilization of heterogeneous data sources and advanced algorithms to reinforce their effectiveness in dynamic and complex environments, such as online grocery shopping applications.

III. PROPOSED METHODOLOGY

A. Methodoogy and Flow Chart

In the context of online grocery shopping, the proposed methodology refers to the development of hybrid recommendation systems that can accurately predict the next product that a user is likely to purchase. As shown in the Figure 3.1, the initial step is to load and merge relevant datasets from the Instacart platform to form a complete record of user interactions with products.

The next step after that is performing data preprocessing, this is where the data is cleaned, sorted, and sequences are generated, etc., in order to get the data ready for modeling. At the heart of the methodology is where they form a Markov Chain model which will aid the model in estimating the probabilities of an item to be used after the previous one. Feature extraction, to be used for user-specific features, product popularity and recency of buying, is also employed for making the model predictive. Combining these three components makes the hybrid predictive model to provide more accurate and personalized recommendations. This leads through the analysis of the results and discussion on the model's applications on e-commerce sites, as well as the impacts it could have in fulfilling a better user experience and increasing e-commerce sales optimization. Rest of the methodology is given in the subsequent sub-headings.

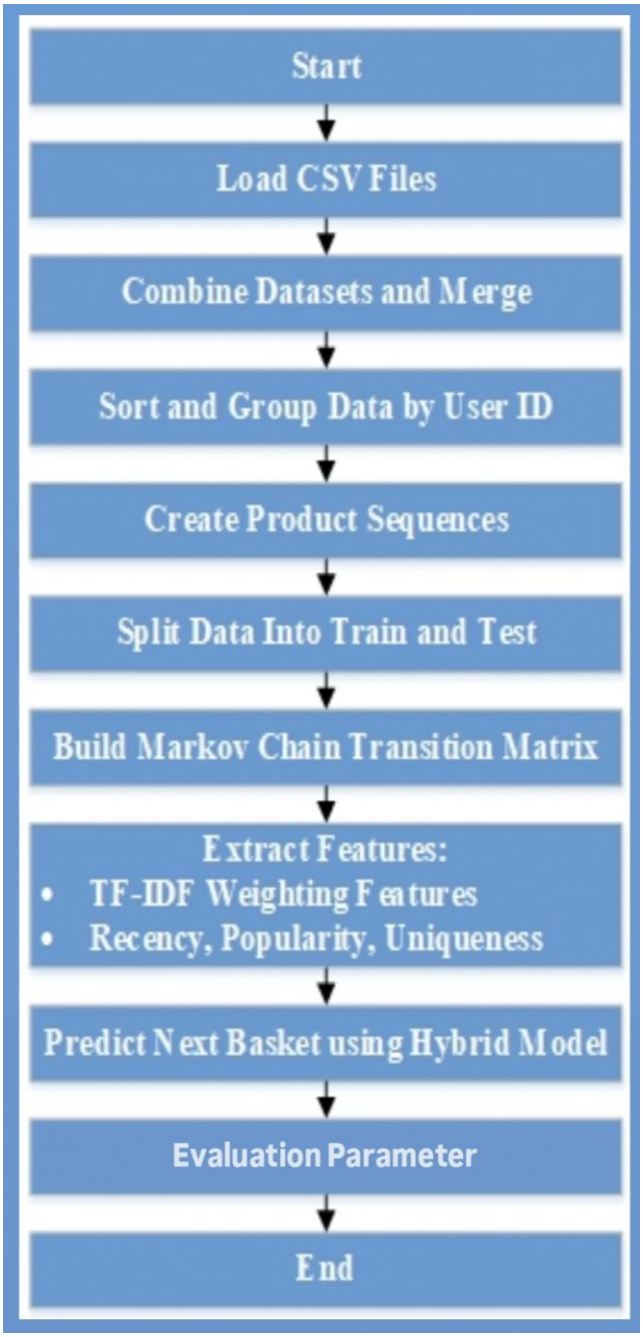


Figure 3.1: Flow diagram for proposed approach for next basket prediction

B. Data Collection

The first step of the recommendation process is to collect data, which must be relevant, and then shape it so that it can be effectively used in different analytical models. Privacy Research This paper is based primarily on data from Instacart. The dataset consists of detailed interactions users have with products on the platform, which is crucial for developing personalized recommendation models. The aim is to collect various datasets that offer insights relating to user behavior, product interaction, purchase history, etc. These insights are crucial for constructing a predictive model for future user purchase behavior.

Loading Data

The process of data collection starts with loading the relevant datasets from the instacart collection. These datasets hold vital data about the user orders, product information, etc. These generally include the following key datasets:

- Orders Data: This dataset represents information of each order by users like order id, user id, number of orders, and time since the last order made by the user. This helps to keep track of how often users purchase, recognize repeat behavior, and note when those purchases take place.

- Order Product Data: This dataset contains the product purchase information for each order. It also tells you which products were purchased for each order, providing product/order ID mappings. Order_products_prior: A dataset containing information about products purchased in prior orders prior, it includes products purchased prior to the training data set. Train orders: The orders data devoted to the training set.

- Products Data: This dataset contains details about the products themselves, like the product ID, name, aisle, and department. This is important for segmenting products, understanding user preferences in a more granular way and in doing feature engineering in the recommendation model.

- Aisles and Departments Data: Those datasets give you the hierarchical context of the products. The aisle or category in which each product is found in the store departments – A higher-level classification of products (i.e. dairy, produce, etc.)

Merging Data

After loading the datasets into the system, the next step is to merge the datasets. Merging takes place to create a comprehensive dataset that assembles all information about user-product interactions, drawn from all available sources. The merging of this data is important for defining purchasing tendencies, as well as predicting future behaviors. The steps undertaken in the merging process are as follows:

- Merge 1: Combining the orders data with the order products data. This step joins the product details to each order, based on the order_id. This way, it allows to correlate between each product purchased with a specific user and order info. It helps to identify the users' order history and thus the users' purchasing behavior can be analyzed.

- Merging Product Information: Once the order data is joined with the information of the products that were purchased, the product description will be mapped to data frame. By joining product dataset with order-product data on.column product_id that were able to include the product features for each transaction, such as product name, aisle and department. By adding product context to the data, this enables us to categorize our products and understand our users better.

- Grouping Data by User and Order: The next important step is to sort the data chronologically on user_id and order_id to maintain the purchases sequence. This is also incredibly valuable for sequential modeling, in which the order of purchases are predictive, and aids in understanding the timing and pattern of user transactions, which is critical for accurate predictions.

- Creating user-product sequences: The last step of this phase after the data has been merged and processed is constructing sequences of products for users. These sequences indicate the history of products a user bought over time. This proximal segmentation being done, it can also directly go for custom sequencing in such a way that can have in sequence of certain data elements coming to the computational structure in such a way that they can act as a user_id's product purchases submission in the same order. These sequences are used to provide the basis for further analysis; for example, to predict the next product a user will buy.

This will provide a truly merged dataset with all relevant details to be aware of user purchasing pattern keyword. It will contain columns such as user_id, order_id, order_number, days_since_prior_order, product_id, product_name, aisle_id, and department_id, as well as other applicable features monitoring the past purchase behavior of the user and various product details.

C. Sequence Generation

After cleaning and sorting the data, the next step is to create meaningful product sequences for each user. These sequences are critical for creating a predictive model to predict the next product the user is going to purchase according to their shopping behavior.

User-Product Sequences: In this step, the main goal is to create product sequences for each user, with each sequence corresponding to the history of products purchased by that user. It does this by grouping by user_id, then listing all products purchased by that user in order of purchase.

So for a user who has multiple orders, one can have a sequence like:

$$User\ 1\ Sequence = \{Product\ A, Product\ B, Product\ C, Product\ D, \dots\}$$

These sequences are significant since they aid in capturing the time and sequence dependencies between products. The model could then leverage that information to determine which products were most likely to be purchased next by the user.

Data Splitting: After the text is tokenized, the next step is to generate the sequences from the data in the previous step. Normally 80-20 split of data is used, in which 80% data is used for training model, and 20% data to test and validate that model.

Training Set: It consists of user-product pairs that will be used to train the model that will provide recommendations to users. This dataset is intended to teach the model the trends it can find in the purchasing behavior of users.

Test Set: This is used to evaluate the performance of the model. To assess how well the model is doing, we measure the accuracy of next product burn, by comparing the model predictions with actual purchases in the test set.

This training vs testing allows for an unbiased evaluation of the model against unseen data, more closely representative of potential real-world performance. This step is important to evaluate the model's ability to generalize and avoid overfitting to the training data.

Mathematical Formulation for Sequence Generation: Formally for sequence generation, let S_u denote the sequence of product purchased by user u:

$$S_u = \{p_1, p_2, p_3, \dots, p_n\}$$

Where:

- p_i is the product bought at the i^{th} position in the order by user u,
- n equals total products purchased by user u.

For a customer u, let S_u be the history of products u has purchased, the task is to predict what will be the next product in the array p_{n+1}.

The model attempts to predict p_{n+1} (which means the next position) with a function f:

$$f(S_u) = p_{n+1}$$

The above function f gives the most likely subsequent item in the sequence, given user purchase history S_u. The model then uses sequential learning techniques to learn the relationships between products and captures the temporal patterns in the user's purchase behavior.

D. Markv chain Model

One such method that has proven to be useful for modeling sequences of events or states, dependent on previous states is the Markov Chain model with the explosion of e-commerce, especially in online grocery shopping, the Markov Chain model can be used to find sequential dependencies for users' product selections. This model is called as n-gram model, the concept in

this model is that the probability of purchasing a product next depends only on one of the previous product(s) purchased and not on the entire purchasing history. It is particularly well suited for modeling purchasing patterns of users, as each purchase in a sequence can be considered a state in the Markov process.

The Markov Chain model assumes that the last purchased item has the most influence on predicting the next item, and essentially models the transitions between products as a series of probabilistic events. Here, the objective is to predict the probability that a user will buy a specific product based on their last purchase(s). Building the Markov Chain involves building the transition matrix and normalizing the transition probabilities to output these transitions based on probability.

Construction of Transition Matrix

So, the first step in developing of Markov Chain model, is creating the transition matrix that will encode the probabilistic relationships between products based on the historical purchase sequences of users. In the transition matrix, each row and column is a product, and each entry is the probability of transitioning from one product to another. For each transaction in the dataset going back far enough, the pairs of items are extracted with their bins, and this bin counting is placed in the next matrix.

Assume we have a set of products as $P = \{p_1, p_2, p_3, \dots, p_k\}$, where p_i a unique product and total count of unique products in the i^{th} dataset=k. T is a $k \times k$ square matrix representing the transition frequency of products in user historical purchase patterns, where $T(i,j)$ denotes how many times product p_j follows p_i. The matrix can be written as:

$$T(i,j) = \text{Count}(p_i \to p_j)$$

Where $\text{Count}(p_i \to p_j)$ represents the number of occasions product p_j is bought immediately after product p_i across all observed purchase sequences.

For instance consider a series of purchases:

$$S = \{p_1, p_2, p_3, p_1, p_2, p_3, p_1\}$$

Here, p_1 leads into p_2, which leads into p_3. The transition matrix will track these transitions, for product pairings that occur regularly, there will be more entries in the transition matrix with higher values.

This transition matrix encapsulates the sequential nature of the product purchasing process. But this matrix still presents counts and has not yet normalized for differencing frequency of transitions between products.

Normalization of Transitions

The important next step after building the raw transition matrix is to convert the transition counts to probabilities. This is achieved by normalising each entry of the transition matrix by the number of transitions from the current product. This is important because it guarantees that each row of the transition matrix now corresponds to a probability distribution, i.e., the sum of the probabilities of all possible next products is equal to one.

The normalization of the transition matrix, which we name P, is obtained by plucking each over $T(i,j)$ entry by the number of transitions from product p_i, which is as follows:

$$P(i,j) = \frac{T(i,j)}{\sum_{i=1}^{k} T(i,j)}$$

Where:

- $P(i,j)$ is the probability that product p_j follows product p_i,
- $T(i,j)$=The raw transition count from product p_i to product p_j,
- $\sum_{j=1}^{k} T(i,j)$ has been denoted as the total number of transitions from product p_i, in other words all transitions allowed from any product p_i.

This normalization guarantees that the sum of the probabilities of all possible products to follow any product p_i equals 1:

$$\sum_{j=1}^{k} P(i,j) = 1 \quad \forall i$$

Normalizing the transition matrix allows us to convert the raw counts into probabilities, which we can use to predict whether a user will buy a certain product according to their last purchases. For instance, if product p_1 was just purchased, the transition matrix will provide the probability of every product p_i being purchased afterwards.

Mathematical Formulation for Markov Chain Model: The Markov Chain model predicts the probability of a user buying the next product after their recent purchase. The probability of purchase for product p_i if user just buys product p_i, is given as:

$$P\left(next\ purchase = p_j \middle| previous\ purchase = p_i\right) = P(i,j)$$

Where $P(i,j)$ is the normalized probability from the transition matrix. This probability is then used for ranking products being recommended, which causes the most likely products for a user, based on his past buying behavior to be suggested.

In building the Markov Chain (transition matrix) and making it normal, etc. The transition matrix captures the probability that a user will buy a particular product given what they have bought in the past, thus providing a probabilistic interpretation of future purchases. This process allows us to normalize the counts of possible transitions into probabilities that reflects our expectation, ultimately allowing the model to draw a meaningful ranking of possible products to deliver a meaningful personalization. This approach is the foundation for constructing a powerful recommendation model that can forecast users' next purchase in their shopping experience.

E. Feature Extraction

The general process of obtaining the most relevant features for your use case in your recommendation system. Feature extraction through TF-IDF Weighting Features and product- and user-based features sets is used in this research. The first0 applies for the history of a user in the importance of a product using the TF-IDF weighting technique, and the second includes user features such as the recency of purchases, popular products, and general product preference. This dual set of features allows for a more personalized experience in terms of product recommendations, enhancing the accuracy and relevance of predictions made by the system.

First Feature Set: TF-IDF Weighting Features

Utilizing TF-IDF weighting is a common method for understanding how important a product is based on the frequency of each product purchased by a user versus that product's frequency across all users. In essence, products with lower product-wise frequency usually indicate the product has higher significance to the particular user and should be given higher importance as it better binds the user profile.
To calculate the TF-IDF score of each product, the following equation is applied:

$$TF\text{-}IDF(p) = TF(p) \times IDF(p)$$

Where:

- $TF(p)$ is the term frequency, which is the number of times product p was in a user purchasing sequence.
- $IDF(p)$: The inverse document frequency which reflects the diversity of product p among all user purchase histories.

Step-by-Step Calculation for TF-IDF Features:

- ***Term Frequency (TF):*** Within each user, we first compute the frequency of product p in the purchase sequence. The term frequency is normalized by taking the number of purchases they made.

$$TF(p) = \frac{Count\ of\ product\ p\ in\ user's\ sequence}{Total\ number\ of\ products\ in\ user's\ sequence}$$

(3.10)

- ***Inverse Document Frequency (IDF):*** The IDF for each product p is calculated based on the frequency or rarity rate of the products across all users. Simply put, a product that a lot of users bought has lower IDF value and a product only a few users bought will have a higher IDF, indicating its uniqueness.

$$IDF(p) = \log\left(\frac{N}{1 + DF(p)}\right)$$

  Where:
  - N = Total number of users in the dataset.
  - $DF(p)$ is the document frequency denoting the quantity of users purchasing the product p.
- ***TF-IDF Score:*** The last part in TF-IDF is the multiplication of two terms from above is the TF-IDF score that we gets for the one product, that is calculate from this Equation (3.9). This score indicates how strongly the user is interested in the product but also what is the mastery in that user's cohort.

These features serve to promote the products that are rare but distinctly influential in a particular user purchasing activity, weighted up by the TF-IDF hash. These feature sets allow the recommendation system to find products that may be unique to their own experience, but are still relevant to the user's preferences.

Second Set of Features

The second set is features that incorporate variables which contribute to how the next purchase will be predicted such as recency, product popularity and user-specific products. These features are useful in letting the recommendation model adjust for factors that affect what items you end up buying — like trends in product popularity and a user's individual purchase history.

Order-Based Features: Recency of Purchases

Recency-based features quantify how recent the user's purchases are and this denotes how fresh the user's purchase history is. This makes sense because more recent purchases are likely to have the greatest impact on a user's interests or needs. For instance, once a consumer has purchased a product, they are more likely to buy complementary things in a particular timeframe.

So for the recency feature, we have the cumulate of how many days since last order for each user. In particular this counts the number of days since the last purchase for each product p in the user's history. This helps capture how time-decay works, meaning that purchases in the past have less significance given more time has passed.

$$Recency\ (p) = \sum_{i=1}^{n} D_{p_i}$$

Where:

- n is the number of all purchases by the user..
- D_{p_i} is the difference in time of current purchase and last purchase for same product p_i.

This recency score helps also to capture the temporal dynamics of user behavior since users often re-purchase an item more often for a period of time.

Product Popularity Features

Another important feature is product popularity, which indicates how frequently a product is purchased by all users in the dataset. Popular products refer to the products that users buy more often and less popular products for those products that are not purchased often. It is important because it enables the recommendation system to prioritize products that are popular among a wider audience.

To quantify how popular a product is, we count how many times the product has been purchased by all users and normalize this count over all products. The popularity score of a product p can be expressed as:

$$Popularity\ (p) = \frac{Number\ of\ purchases\ of\ product\ p}{Total\ number\ of\ users}$$

The product popularity feature captures these trends by weighting products more — giving more influence to products that are more commonly purchased. These are also products that tend to be recommended to users based on their more general purchasing habits.

User-Specific Features: Unique Products Ordered by User

User-specific features are an approach for capturing preferences for a user by analyzing their shopping behavior. One of the specific features related to the user, is the number of unique products purchased by a user. This feature reveals the variety of products a user has purchased and may provide details about their shopping habits.

To calculate the unique product count purchased by the user, we return the length of the distinct product columns in the purchase history. The unique products ordered by user u can be calculated by the following formula:

$$Unique\ Products(u) = |p_1, p_2, \dots, p_k|$$

Where:

- $p_1, p_2, \dots, p_k$ are the unique products purchased by user u.
- $|\cdot|$ denotes the cardinality of the set, which gives the number of unique products.

That allows this feature to personalize bill recommendation process through identifying certain user with diversified interests and recommending the bill products which address their specific preferences. A user who buys dairy, fruits, and electronics differs from a user that only buys groceries, in the user's preferred purchase behavior.

In developing a powerful recommendation system, the feature extraction discussed in this section is a key element. By adding TF-IDF weighting features, the model enhances its focus on rare but significant products unique to individual users. Also, the second group of features—recency, product popularity, and user-specific preferences—refines the recommendations further while considering the temporal and

marketplace dynamics of product purchases. Such profiles aggregate user information through behavioral tracking over unique time frames and are a key component for a more accurate recommendation system that enhances personalizing shopping experiences. Combination of these feature sets provides a holistic way to learn from user behavior to predict which product will be the next to purchase as these layers ensure the recommendations are not only based on historical behavior but also considering holistic perspectives of trends outside of user linear behavior.

F. Hybrid Prediction Model

Combining multiple data sources and techniques — at the heart of this system is Hybrid Prediction Model which improves the accuracy and relevance of product recommendations. The model itself is a combination of Markov Chain transitions, TF-IDF weighting as well as other features like popularity of product purchased, recentness of purchase and user features. We develop a model that strives to produce personalized probabilistic outputs by combining the sequential dependencies as learnt via the Markov Chain model, as well as more general characteristics of products, including their popular and unpopular nature. By using a combination of unique techniques a more accurate and context aware recommendations can be achieved.

Predicting Next Basket

The primary challenge when predicting the future basket is to calculate how likely it is that a user will add a given set of products to their cart based on their current basket (which products users have already purchased). We do this by adjusting the output of the Markov Chain model with features that we derive from, TF-IDF weighting coupled with other contextual features like product popularity, recency and user-specific preferences.

This model works well with the data as we have trained on sequential data and Markov Chain is probabilistic model designed to predict the next based on previous data training, meanwhile achieving a possibility for rare, but significant products using TF-IDF weight for sample of each training. The popularity of products allows incorporation of wider market trends, recency leads to more novel products being considered more relevant and user-based features ensure that the idiosyncrasies of each user are being captured.

Given a current basket, defined as $B = \{p_1, p_2, \dots, p_m\}$ (where $p_1, p_2, \dots, p_m$ are products that the user recently purchased), the problem is to predict the next basket — the set of products that will most likely be purchased next. Here the recommendation is based on calculating the probability of a product p_i being added to the basket after taking into consideration the current products B. Mathematically this will be as follows:

$$P(next\ basket|B) = P(p_1, p_2, \dots, p_m|B)$$

This probability is calculated by considering the sequential data (across the Markov Chain transitions) and the extra features.

Sequential Component (Markov Chain): The transition probabilities in the Markov Chain model indicate the probability that a user buys product p_i following product p_i given the users past purchase sequences. Given that the user has just purchased product p_i, the probability of them subsequently purchasing product p_i can be defined as.

$$P(p_j|p_i) = \text{Transition Probability from } p_i \text{ to } p_j$$

These probabilities are extracted from the transition matrix constructed using the Markov Chain model, which reflects the relationships between products over time.

Feature-Based Component (TF-IDF, Popularity, Recency, User-Specific Features): To further refine our prediction, we introduce additional features, including TF-IDF weights, product popularity, recency, and user-specific preferences. These features enable the model to take into account not just the series of purchases but also overarching trends and specific user activity.

- ***TF-IDF Weighting:*** The TF-IDF score, as discussed above, measures the significance of products within the user's sequence. Using a higher weight for items that are rare but important to the user will prevent bias from occurring for unique preferences.
- ***Product Popularity:*** It is a view of how popular the product is to all users. The more a product has been purchased, the more likely it is to be recommended. The popularity score is normalized for each product to make it comparable.
- ***Recency:*** It is an additional possible measure, which shows how recent the purchase of a product by a user is, and thus recommends more recent purchases. It is the sum of the time since previous orders, iterated cumulatively.
- ***User-Specific Features:*** These features capture specific characteristics of a user including the number of unique numbers of products bought or the diversity of items in a user's shopping cart. It helps compensate for shopping behavior at the individual level, which is not always reflected in broader trends.

Hybrid Score Calculation

With probabilities derived, and feature scores calculated, combining them to a single hybrid score per potential next product follows. This score combines the sequential data (through Markov Chain probabilities) with the other features (using TF-IDF, popularity, recency and user characteristics). The hybrid score for product p_i is computed as a weighted sum of these factors:

$$Score(p_j) = \alpha \cdot P(p_j|B) + \beta \cdot Popularity(p_j) + \gamma \cdot Recency(p_j) + \delta \cdot User\text{-}Specific\ Features(p_i) + \lambda \cdot TF\text{-}IDF(p_i)$$

Where:

- $P(p_i|B)$ a transition probability from the Markov Chain model
- $Popularity(p_i)$ is a product popularity score.
- $Recency(p_i)$ is the recency score for the product.
- $User\text{-}Specific\ Features(p_i)$ represent behaviors specific to the users, like the number of distinct products they have purchased.
- $TF\text{-}IDF(p_i)$ denotes the TF-IDF weighting of product p_i, highlighting underrepresented but substantial products.
- The parameters $\alpha, \beta, \gamma, \delta, \lambda$ are weights that give importance to features in making predictions.

The weights $\alpha, \beta, \gamma, \delta, \lambda$ are determined using the training data and sensitive to trade-off between each aspect. That is, the hybrid score weights the probabilistic transitions between products (using the Markov Chain) against the product-specific (TF-IDF) features as well as popularity, recency, and personal preferences.

Mathematical Formulation for Hybrid Model: Based on these components of the model, the probability of product p_i being the next product in user's shopping journey is calculated as:

$$P(p_j|B) = \alpha \cdot \frac{Count(B \to p_j)}{\sum_{j=1}^{k} Count(B \to p_j)} + \beta \cdot \frac{Popularity(p_j)}{\sum_{j=1}^{k} Popularity(p_j)} + \gamma \cdot \frac{Recency(p_j)}{\sum_{j=1}^{k} Recency(p_j)} + \delta \cdot \frac{User\text{-}Specific\ Features(p_j)}{\sum_{j=1}^{k} User\text{-}Specific\ Features(p_j)} + \lambda \cdot \frac{TF\text{-}IDF(p_j)}{\sum_{j=1}^{k} TF\text{-}IDF(p_j)}$$

Where:

- The $Count$ is the number of valid transitions from basket B to product p_i.
- All of the previously mentioned features, those of popularity, recency, user-specific features, and TF-IDF are normalized so that they are more comparable through scale.
- The denominator contains a summation that normalizes the contributing components (e.g., popularity and recency) across products.

The final prediction is done by sorting the products based on their hybrid scores and recommending top N products. The above products are the ones that are most likely being the next purchased by the user according to both its previous purchase history and wider contextual data.

IV. Results and Analysis

A. *Evaluation Parameters*

It is vital to assess the effectiveness of the hybrid recommendation model in predicting the next products that a user is willing to buy. To quantify this we use standard metrics from classic classification and recommendation tasks: Next-Basket Relevance Score, Next-basket coverage, and Harmonic Basket Prediction Score. These metrics can be examined to understand how well the model performs in achieving a good prediction of relevant products. These metrics are intuitively helpful in understanding the model's skills at recommending products based on user's historical purchases.

The idea behind these metrics is that we take the predicted next products and compare them to the actual items that users bought in the test set. Compared to this reference, we compute the following metrics:

Next-Basket Relevance Score (NBRS)

The Next-Basket Relevance Score (NBRS) measures the fraction of relevant recommendations among all products suggested by the model. It is mathematically defined as:

$$NBRS = \frac{\text{Relevant Captures (RC)}}{\text{Relevant Captures (RC)} + \text{Irrelevant Suggestions (IS)}}$$

Where:

- Relevant Captures (RC): Products recommended by the model that were actually bought.
- Irrelevant Suggestions (IS): Products recommended by the model but never purchased.

A high NBRS means the model recommends mostly relevant items, minimizing unnecessary suggestions.

Next-basket coverage

Next-basket coverage (or Sensitivity) defines how many of the relevant products (the products that were actually bought by the user) were correctly recommended. It is mathematically defined as:

$$NBC = \frac{\text{Relevant Captures (RC)}}{\text{Relevant Captures (RC)} + \text{Missed Purchases (MP)}}$$

Where

Missed Purchases (MP): Products that the user purchased but were not recommended by the model.

Harmonic Basket Prediction Score

The Harmonic Basket Prediction Score is the harmonic mean of Next-Basket Relevance Score and Next-basket coverage: it provides one number that takes into account both.The Harmonic Basket Prediction Score is defined as:

$$HBPS = 2 \times \frac{NBRS \times NBC}{NBRS + NBC}$$

Thus, Harmonic Basket Prediction Score is used wherever both Next-Basket Relevance Score and next-basket coverage are equally important. A high value of Harmonic Basket Prediction Score means the model can recommend relevant products well (high Next-Basket Relevance Score) and can also identify all relevant products (high next-basket coverage).

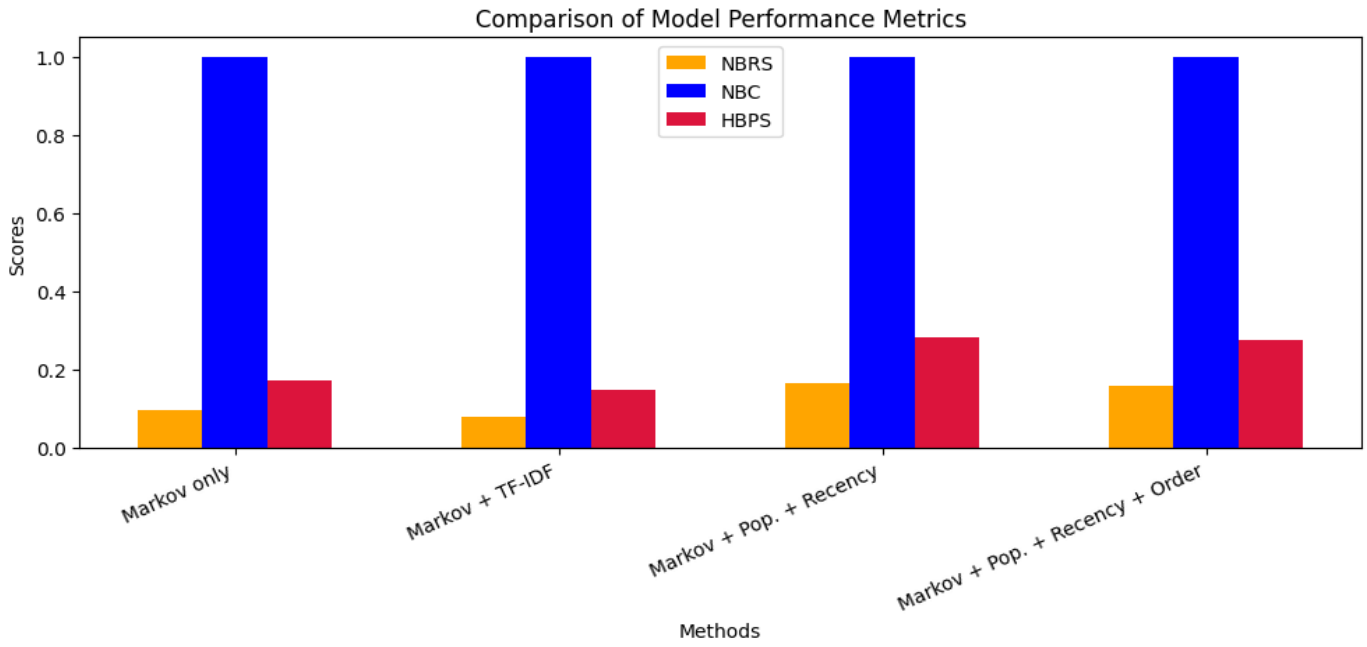
Assessing these metrics tells us about the accuracy (Next-Basket Relevance Score) vs. completeness (next-basket coverage) trade-off of the hybrid model. Moreover, by analyzing the performance of the hybrid model with respect to conventional recommendation systems like collaborative filtering, this research can showcase the improved prediction accuracy and enhanced personalization in recommendations that the hybrid approach achieves through integrating information from Markov Chains with TF-IDF weighting and other contextual aspects.

B. Results

In this results section, an in-depth analysis is presented of the performance of the model through various evaluation metrics. It shows how the implementation of features like TF-IDF, Popularity, Recency, and Order, affects the prediction accuracy of the recommender system. Then, it's followed by some visualizations (bar charts, prediction evaluation matrixes, etc.) that validate the findings. The proposed models are compared using Next-Basket Relevance Score, Next-basket coverage, and Harmonic Basket Prediction Score as the evaluation metrics. Results also demonstrate the effectiveness of hybrid methods compared to traditional Markov-based methods.

Method	**NBRS**	**NBC**	**HBPS**
Markov only	0.095	1	0.1735
Markov + TF-IDF	0.08	1	0.1481
Markov + Pop. + Rec.	0.165	1	0.2833
Markov + Pop. + Rec. + Or.	0.160	1	0.2759

Table and figure summarizes the performance of different models based on their Next-Basket Relevance Score, next-basket coverage, and Harmonic Basket Prediction Score. It shows the gradual enhancements made in the base Markov model in using features like tf-idf, popularity, recency and order. This shows that using both contextual features and sequential features lead higher values for Next-Basket Relevance Score and Harmonic Basket Prediction Scores (and perfect next-basket coverage), thus proving that context-aware recommendation systems can benefit from the hybridization.



CONCLUSION AND FURURE SCOPE

The study successfully develops and validates a hybrid recommendation system to overcome the limitations of traditional systems applied to next-basket prediction tasks. The proposed model combines sequential dependencies using Markov Chains with contextual relationships reflected in advanced features such as TF-IDF weighting, popularity, recency, and order-specific data.

The experimental evaluations show substantial improvements in performance. The best performance was achieved by the Markov + Popularity + Recency model, with Next-Basket Relevance Score 0.165, next-basket coverage = 1 and Harmonic Basket Prediction Score = 0.2833. In contrast, the Markov only model gained Next-Basket Relevance Score of 0.095 and Harmonic Basket Prediction Score of 0.1735. These findings demonstrate that the inclusion of contextual and sequential elements leads to an improvement in the accuracy of recommendations.

This research also highlights that in addition to using Markov Chains, feature-based approaches are necessary to alleviate sparsity and cold-start problems. This model we propose is to be used in reality by e-commerce platforms with the advantage of being scalable and personalized. This work shows how hybrid models can greatly help to increase predictive quality and establishes a clear path for future improvement of recommendation systems.

- Utilization of Deep Learning: Use more advanced architectures [such as RNNs and Transformers] to model orders correctly or capture complex sequential dependencies and long-term dependencies.
- Multi-modal data can be used such as using text, pictures, and user reviews to enhance the feature extraction helping in achieving the diversity of recommendations.
- Real-Time Recommendations: In this contact less city era with everyone using smart device real-time predictions to cater per users requirements becomes
  necessity and even the capabilities of real time predictions can be utilized on a truly massive scale where there is a huge data stream.
- Domain Expansion: Extend the framework to other domains including healthcare, education, and retail, capitalizing on its adaptability to varied contexts.

REFERENCES

[1] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J., 1994, October. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186).

[2] Zadbood, A. and Hoffenson, S., 2022. Social network word-of-mouth integrated into agent-based design for market systems modeling. *Journal of Mechanical Design*, *144*(7), p.071701.

[3] Tahmasebi, F., Meghdadi, M., Ahmadian, S. and Valiallahi, K., 2021. A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. *Multimedia Tools and Applications*, *80*, pp.2339-2354.

[4] G. Silva, M., C. Madeira, S. and Henriques, R., 2024. A Comprehensive Survey on Biclustering-based Collaborative Filtering. *ACM Computing Surveys*, *56*(12), pp.1-32.

[5] Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., Wen, Z., Wang, F., Zhao, X., Tang, J. and Li, Q., 2023. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*.

[6] Li, Y., Liu, K., Satapathy, R., Wang, S. and Cambria, E., 2024. Recent developments in recommender systems: A survey. *IEEE Computational Intelligence Magazine*, *19*(2), pp.78-95.

[7] Shao, Z., Wang, S., Zhang, Q., Lu, W., Li, Z. and Peng, X., 2022, October. A systematical evaluation for next-basket recommendation algorithms. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 1-10). IEEE.

[8] Xie, Z., Wu, J., Jeon, H., He, Z., Steck, H., Jha, R., Liang, D., Kallus, N. and McAuley, J., 2024, October. Neighborhood-Based Collaborative Filtering for Conversational Recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems* (pp. 1045-1050).

[9] Ko, H., Lee, S., Park, Y. and Choi, A., 2022. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, *11*(1), p.141.

[10] Roy, D. and Dutta, M., 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1), p.59.

[11] Yu, F., Liu, Q., Wu, S., Wang, L. and Tan, T., 2016, July. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 729-732).

[12] Che, B., Zhao, P., Fang, J., Zhao, L., Sheng, V.S. and Cui, Z., 2019. Inter-basket and intra-basket adaptive attention network for next basket recommendation. *IEEE Access*, *7*, pp.80644-80650.

[13] Faggioli, G., Polato, M. and Aiolli, F., 2020, July. Recency aware collaborative filtering for next basket recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization* (pp. 80-87).

[14] Xia, Y., Di Fabbrizio, G., Vaibhav, S. and Datta, A., 2022. A Content-based Recommender System for E-commerce Offers and Coupons. In *Proc. SIGIR Workshop eCommerce*.

[15] Airen, S. and Gupta, P., 2021. AIIoT-Enabled Soil Irrigation System. African Diaspora Journal of Mathematics, 24(1).

[16] Dou, X., 2020, April. Online purchase behavior prediction and analysis using ensemble learning. In *2020 IEEE 5th International conference on cloud computing and big data analytics (ICCCBDA)* (pp. 532-536). IEEE.

[17] Lee, H.I., Choi, I.Y., Moon, H.S. and Kim, J.K., 2020. A multi-period product recommender system in online food market based on recurrent neural networks. *Sustainability*, *12*(3), p.969.

[18] Zheng, Q. and Ding, Q., 2022. Exploration of consumer preference based on deep learning neural network model in the immersive marketing environment. *Plos one*, *17*(5), p.e0268007.

[19] Tahiri, N., Mazoure, B. and Makarenkov, V., 2019. An intelligent shopping list based on the application of partitioning and machine learning algorithms. In *SciPy* (pp. 85-92).

[20] Gupta, A. and Shrinath, P., 2023. A novel recommendation system comprising WNMF with graph-based static and temporal similarity estimators. *International Journal of Data Science and Analytics*, *16*(1), pp.27-41.

[21] Li, M., Jullien, S., Ariannezhad, M. and de Rijke, M., 2023. A next basket recommendation reality check. *ACM Transactions on Information Systems*, *41*(4), pp.1-29.

[22] Le, D.T., Lauw, H.W. and Fang, Y., 2019. Correlation-sensitive next-basket recommendation.(2019). In *28th International Joint Conference on Artificial Intelligence, Macao, China* (pp. 10-16).

[23] Ariannezhad, M., Jullien, S., Li, M., Fang, M., Schelter, S. and de Rijke, M., 2022, July. ReCANet: A repeat consumption-aware neural network for next basket recommendation in grocery shopping. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1240-1250).

[24] Ariannezhad, M., Li, M., Schelter, S. and De Rijke, M., 2023, February. A personalized neighborhood-based model for within-basket recommendation in grocery shopping. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (pp. 87-95).

[25] Han, J., Pei, J. and Tong, H., 2022. *Data mining: concepts and techniques*. Morgan kaufmann.

[26] Alawadh, M.M. and Barnawi, A.M., 2022. A survey on methods and applications of intelligent market basket analysis based on association rule. *Journal on Big Data*, *4*(1).

[27] Airen, S. and Gupta, P., 2020. A Customer Preference-Based Intelligent Song Recommendations System. African Diaspora Journal of Mathematics, 23(6).

[28] Kokoç, M., Ersöz, S., Aktepe, A. and Türker, A.K., 2016. Improvement of facility layout by using data mining algorithms and an application. *International Journal of Intelligent Systems and Applications in Engineering*, *4*(Special Issue-1), pp.92-100.

[29] Ünvan, Y.A., 2021. Market basket analysis with association rules. *Communications in Statistics-Theory and Methods*, *50*(7), pp.1615-1628.

[30] Zimdars, A., Chickering, D.M. and Meek, C., 2023. Using temporal data for making recommendations. *arXiv preprint arXiv:1301.2320*.

[31] Gupta, P., Sharma, V. and Varma, S., 2022. A novel algorithm for mask detection and recognizing actions of human. Expert Systems with Applications, 198, p.116823..

[32] Mao, H., Mao, M. and Mao, F., 2024. Ranking on user–item heterogeneous graph for Ecommerce next basket recommendations. *Knowledge-Based Systems*, *296*, p.111863.

[33] Gupta, P., Sharma, V. and Varma, S., 2022, September. An Algorithm for Counting People using Dense Nets and Feature Fusion. In 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1248-1253). IEEE..

[34] Guidotti, R., Rossetti, G., Pappalardo, L., Giannotti, F. and Pedreschi, D., 2018. Personalized market basket prediction with temporal annotated recurring sequences. *IEEE Transactions on Knowledge and Data Engineering*, *31*(11), pp.2151-2163.

[35] Kraus, M. and Feuerriegel, S., 2019, July. Personalized purchase prediction of market baskets with wasserstein-based sequence matching. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2643-2652).

[36] Airen, S. and Gupta, P., 2020. A Customer Preference-Based Intelligent Song Recommendations System. African Diaspora Journal of Mathematics, 23(6).