Data 294 Research Paper – A Study on Recommendation Systems

Divya Puraswani

013755391

MS in Data Analytics

San Jose State University

1 Washington Sq, San Jose, CA 95192

**Contents**

[Abstract 3](#_Toc8598446)

[Introduction 4](#_Toc8598447)

[Purpose 5](#_Toc8598448)

[Literature Review 6](#_Toc8598449)

[Building a Recommendation System 7](#_Toc8598450)

[Conclusion 17](#_Toc8598451)

[References 18](#_Toc8598452)

# Abstract

With the shift from traditional store shopping to online store(internet) shopping it has become important for the companies to decide their marketing strategies which leads to an overall improvement in sales conversions. A Recommender system is one such strongest weapon which helped Amazon to generate revenue. Studying Amazon’s approach to build personalized system for its customers and then using different approaches to build a recommendation systems model based on machine learning algorithms to find the most similar items by considering amazon dataset for toy products is the idea behind the research.

# Introduction

Data is just like a dense forest, the more you throw light on it, the better you can discover the path. This light of applying data science techniques to discover the path of insights has benefitted almost every industry today. One such industry is retail e-commerce which serves as a platform between vendors and customers. The industry is expected to grow to 4.135 trillion US dollars in 2020 and 4.88 trillion dollars in 2021 (Statista, 2019). Applying data science techniques of customer segmentation, market basket analysis, fraud detection, lifetime value prediction, recommendation engines, etc. would act as a catalyst to this growth allowing to answer 5 Ws about customers i.e. Who is our customer? What are the preferences of the customer?, When does the customer purchase most?, Why and Where does the customer purchase most from?.

The Dictionary defines the word ‘Recommendation’ as a suggestion or proposal as to the best course of action. Thus, a recommendation system is a computer program generated through machine learning algorithms. It helps retailers to connect meaningfully with customers by allowing them to focus on the products and content they desire on the basis of their past activities, purchases, and other customers ratings. Amazon credits recommender systems for an increase in 35% of their revenue. Best Buy has also gained huge returns after implementing recommender systems. (Recommendation Systems – How Companies are Making Money, n.d.). Changing the way of interaction with services, powerful recommendation systems play a major role in framing companies’ sales and marketing strategies resulting in a happy and satisfied customer who is there behind the success of every business or organization in the retail sector.

# Purpose

Given the growing role that data science techniques will play in the retail e-commerce industry this paper focus on how recommendation systems helped companies to earn revenue. By considering Amazon toy product dataset and building simple recommender system based on popularity, content-based using nearest neighbors machine learning algorithm and collaborative recommendation system is the idea behind the research.

**About Recommendation Systems and their types**

Recommendation engines help customers to discover products and services by predicting users’ ratings, the similarity in interests, purchases, etc. As a data filtering tool are in three different categories:

1. Collaborative Filtering
2. Content-Based Filtering
3. Hybrid Recommendation Systems

**Collaborative filtering** recommends new items which are unknown to customers based on historic crowdsourced information about other customer preferences. They are categorized as:

* User-User Collaborative filtering: Effective for a small number of users yet time taking algorithm which tries to look for similarity between users. Example: User A is Athlete and purchases protein shake, milk, eggs, and User B is also an athlete and purchases milk, Since, A and B are athlete protein shake and eggs can be recommended to User B.
* Item-Item Collaborative filtering: This looks for items which are similar to each other and is efficient and less time taking for a new customer. Example: User A purchases skirt, top, jeans, User B purchases skirt, top and User C purchases top So User C can be recommended to purchase skirt as well.

**Content-Based filtering** recommends items based on the similarity between features or characteristics of the product. Example: User A search’s for Indian attire suits they can also be recommended for Indian attire saree or legging kurta etc. This requires to have specific domain knowledge making scalability a challenge.

**Hybrid recommendation engines recommend** items combining features of both collaborative and content-based recommendations. This increases its performance in an effective manner. Netflix uses this approach. (Maruti tech labs , n.d.)

# Literature Review

Using customer’s interests based on several factors such as demographics, items viewed etc, as input several recommendation applications are built. A study about “Amazon.com Recommendations: Item-to-item Collaborative Filtering” by Greg Linden, Brent Smith, and Jeremy York compares approaches of traditional collaborative, cluster models and search based methods with their approach of item-to-item collaborative filtering which personalizes the online store for each customer changing radically based on his/her interests, showing baby items to a new mother.

Traditional Collaborative filtering generates recommendations based on a few customers similar to the user. It reduces the quality in several ways: if the sample of customers is small, selected customers will be less similar to the user, item space partitioning restricts recommendations to a specific product or subject area etc, The goal of Cluster Models algorithm is to assign the user to the segment containing the most similar customers. It has recommendations which are too general or too narrow thus popular items by the same author or in the same subject category fail to achieve this goal. Amazon’s item-to-item collaborative filtering scales to massive data sets and produces a high-quality recommendation in real time which offer customers product suggestions based on the items in their shopping cart by building a similar items table by finding items that customers tend to purchase together. An iterative algorithm provides a better approach and key to its scalability and performance is that it creates the expensive similar- items table offline being better than other algorithms for targeted marketing (Greg Linden, 2003).

# Building a Recommendation System

**About Dataset**

This is a pre-crawled dataset providing information about toy products, taken as a subset of a bigger dataset (more than 115k products) that was created by extracting data from Amazon.com downloaded through Kaggle (Kaggle, n.d.) to know more about the customer and getting closer to them by knowing their preferences or choice for product which can help in building a recommendation engine.

Description and Structure of the data

Dataset is in CSV file format having 10000 rows and 17 columns:

* uniq\_id- gives a unique id of product
* product\_name- the name of toy products like Hornby Santa’s Express train Set, Funko POP! Harry Potter Sword etc.
* manufacturer- This column tells the names of manufacturers, as reported on Amazon. Few like Disney and others are found to outsource their assembly line.
* price-gives price of the product.
* number\_available\_in\_stock- tells the quantity available in stock and its status new etc,
* number\_of\_reviews- number of reviews given by the customer on the product.
* number\_of\_answered\_questions- Amazon includes a Question and Answer service on all or most of its products. This field is a count of how many questions that were asked actually got answered.
* average\_review\_rating- gives an average rating out of 5
* amazon\_category\_and\_sub\_category- A tree-based, >>-delimited categorization for the products which are present in.
* customers\_who\_bought\_this\_item\_also\_bought- This column references to other items that similar users bought. It can be one of a recommendation engine component that can play a big role in making Amazon popular.
* Description- gives a description of the product.
* product\_information- gives technical details like weight, recommended age , etc. about the toy product
* product\_description- gives a description of the product.
* items\_customers\_buy\_after\_viewing\_this\_item- it is similar to column customers\_who\_bought\_this\_item\_also\_bought
* customer\_questions\_and\_answers- This column contains a string entry with all of the product's JSON question and answer pairs.
* customer\_reviews- This column contains a string entry with all of the product's JSON reviews.
* sellers- This column contains a string entry with all of the product's JSON seller information (many products on Amazon are sold by third parties).

**Data Exploration/Exploratory Analysis**

Exploring datasets forms an important pillar of data science process which helps in the detailed understanding of data and allows to stay close with data by cleaning, wrangling/munging (transforming data) and snooping (poking around data). There are several tools available for the same. I used extensive Python libraries like Pandas, NumPy, Matplotlib and Seaborn for the same. There are different formats in which datasets are available (XLS, TXT, CSV, JSON). Python has several libraries which make easy to load data from any source. Amazon toy products data is in CSV format which is loaded using the read\_csv function. After having read the dataset using functions- head() and tail() following inferences were drawn-

1. Using .shape function I found that there are 10000 rows and 17 columns in the dataset.

2. On gaining more information about dtypes, non-null values using .info() function following inferences were drawn-

* Dataset has 1 numerical column and 16 columns are stored as object
* uniq\_id and product\_name does not have any null/missing values and rest all other columns have null/missing values
* price column should be float but is stored as object
* uniq\_id can be renamed to unique\_id
* Memory storage is 1.3+ MB

3. Checking missing values in the dataset using .isnull().any() and also count of null values present in particular column using .isnull().sum() inferred that only uniq\_id and product\_name does not contain null values. Maximum null values about 90.86% are present in the customer\_questions\_and\_answers column so it can be dropped as it does not support much significance in the analysis.

4. Since description and product\_description have same values so description after merging with product\_description a new column description\_of\_product is formed and duplicates can be removed.

5. I replaced null values with “ ” in an amazon\_category\_and\_sub\_category column as it will help in knowing the popular category and subcategories under which products are purchased by the customer.

6. I split “number\_available\_in\_stock” into “quantity\_available” and “quantity\_available\_status” than concatenating this new dataframe with old dataframe and dropping the “number\_available\_in\_stock” column.

7. I also changed datatypes of columns:

* The datatype of quantity\_available from object to float
* The datatype of number\_of\_reviews from object to numerical
* The datatype of average\_review\_rating from object to numerical
* The datatype of price from string/object to numerical and then I filled null values using the mean price of data.

Using helper function made the program easy to compute and read.

8. I renamed column “uniq\_id” to “unique\_id” for easy interpretation.

9. Using. describe () gives descriptive data of all the numerical columns in the dataset which helps in analyses.

For analyzing more on the dataset I looked for Who is the leading manufacturer for toy products in Amazon? And What are major categories and subcategories under which toys are brought?

A screenshot of a social media post

Description automatically generated

Figure 1

**Analysis:** Lego is the leading manufacturer of toy products in Amazon.

A screenshot of a cell phone

Description automatically generated

Figure 2

**Analysis:** Hobbies being Major Category under which toys are bought.

A screenshot of a cell phone

Description automatically generated

Figure 3

**Analysis:** Toy, Vehicles & Accessories being top among subcategory for purchasing toys.

I started with one of the simplest approaches to **Recommender System i.e. building Popularity- Based Recommenders** which is simply telling overall the products that are popular. Example, In the shopping store popular dresses can be suggested by purchase count. (M, n.d.)

In toy product dataset to know the top five most popular products, I counted which product has a maximum number of average review rating using group by and count functions. The result for which is as shown in the data frame as shown in Figure 4-

A screenshot of a cell phone

Description automatically generated

Figure 4

Analysis: Zoo Animal Hand Sock Glove Finger Puppets Sack Plush Toy Cow, Polyhedral Dice, Meng “Model 1:35 Toyota Hilux Pick Up truck w/ZU23-2” Kit(Multi-Colour), Playmobil 6678 Large Floating Pirate Raiders’ Ship with 3 Pirates and Tofern Syma are popular ones which Amazon can keep always in stock.

After knowing popularity, a function named recommend\_me was created which takes an item (product\_name) as input which is rated over 3 stars and suggests the highest-rated matching item. Here product recommended for Pony is Dolls House Miniature 1:12th Scale Rocking Horse Pony, Fire truck is Siku 2110 Model Fire Engine Crane Truck Assorted Colors as shown in Figure 5.

A screenshot of a cell phone

Description automatically generated

Figure 5

**Machine Learning Based Recommendation System: Content-Based Recommender system implemented using a Nearest Neighbors Algorithm**

The k-nearest neighbors (KNN) is one the basic, simple, easy to implement algorithms in machine learning which assumes that similar things exist in close proximity. In simple words, it can be understood as things which are alike are near to each other. (Harrison, 2018) It is a versatile algorithm and can be used for both classification and regression problems. In classification problems to predict the label of an instance first find k closest instances to the given one based on the distance metric and based on the majority voting scheme or weighted majority voting (neighbors which are closer are weighted higher) predict the labels. In an unsupervised algorithm such as in this reference simply the neighbors can be found and used to recommend items which are similar to each other. (Mayeesha, 2018) Hence, useful in solving problems that have solutions that depend on identifying similar objects.

I used price as 16.0$ and average review rating as 4.0 to recommend items similar to that keeping value of the number of neighbors to find to 3. As the value of K increases, predictions are more likely to be stable and accurate due to averaging or majority voting. The result is shown in Figure 6.

A screenshot of a cell phone

Description automatically generated

Figure 6

**Analysis:** Using .loc as it is primarily label based for indexing and identifying items Doctor Who Wave 2 Action Figure – Zygon, Display Box - Pirates & Corsairs – Papo, Disney World of Cars lizzie' are product names similar or nearest to the values specified. Depending on the preferences of the customer the value of features can be changed, and similar items can be found.

Though being easy to implement, this algorithm becomes significantly slow with the growing size of the data.

**Machine Learning Based Recommendation System: Model-based Collaborative Filtering**

**system implemented using SVD Matrix Factorization**

From Scikit learn’s (sklearn) decomposition, TruncatedSVD is imported to perform SVD in the context of recommendation systems used as a collaborative filtering algorithm which offers a speed and scalability Most of these collaborative filtering algorithms are based on user-item rating matrix where each row represents a user (here unique\_id is used for the same), each column an item (here product\_name represents the same). The entries of this matrix are ratings (here average\_review\_rating column represents the same) given by users to items (Malaeb, 2016). These matrices are known as Utility matrix or user-item matrix. They are usually sparse as every user does not review every item so null values can also be present in these matrices.

Singular value decomposition or SVD is a linear algebra method that can be used to decompose a utility matrix into three compressed matrices. It is useful for building a model-based recommender system because these compressed matrices can make recommendations without having to refer back to the complete and entire dataset.

The first performed step is the creation of a list of column names unique\_id, product\_name, price, average\_review\_rating and saving it in an amazon\_data dataframe. This dataframe contains records for each of the users, each product names which are reviewed and the rating they gave to each of the products. Group by product\_name and then count up the number of ratings that were given to each of the products using the .count() function, then sort the data frame in descending values using .sort\_values (ascending=False). A matrix called rating\_crosstab is created which contains a value for each unique\_id and each product\_name cases where ratings are not provided null values are returned. Transposing this utility matrix and then using SVD to decompose it down to synthetic representations. A list of product names is created and numeric index for the product of our interest is taken which is passed for calculating Pearson r correlation coefficient for every product pair in the resultant matrix such that the goal of recommending the product that has the highest correlation with our product of interest, based on generalized user preferences (ratings given by them). (Lillian Pierson, n.d.)

A screenshot of a social media post

Description automatically generated

Figure 7

**Analysis:** Batman Transform And Attack Batmobile is the product of our interest and the products which are highly correlated to that are Heavy Duty roulette set with bakelite and metal wheel, Nerf Super Soaker Flash Blast and Oxford Diecast Trolleybus Cardiff (Streamline).

# Conclusion

Artificial Intelligence and Machine learning have made a great come back in all senses be it sight, touch, smell, etc. The rapid advancement and increasing use of both can enable retailers to enhance efficiency and productivity by applying recommendation algorithms for targeted marketing both online and offline. Used in social media websites like Facebook, Instagram, Music streaming companies like Spotify, movies/tv shows streaming companies like Netflix, different approaches to build recommendation systems have advantages and disadvantages. For big retailers like Amazon, a good recommendation algorithm is one which can cover large customer bases and product catalogs with least processing time to generate the result, able to respond promptly to changes in data of user’s and generating appropriate recommendations. Item-to-item collaborative filtering is one such algorithm. More the data is feed into the engine, more personalized the recommendations become helping retail giants in making smarter, faster decisions, driving business efficiently thus winning customer loyalty and uncovering trends.

Recommendation systems help in listening to customers by telling out some story every time they shop at the store in either online or offline mode. With the increase in the technology one thing is sure that data science will have a lot to offer in the world of retail!! (Bennani, 2018)

# References

(n.d.). Retrieved from Maruti tech labs : https://www.marutitech.com/recommendation-engine-benefits/

(n.d.). Retrieved from Kaggle: https://www.kaggle.com/PromptCloudHQ/toy-products-on-amazon

(2019, May). Retrieved from Statista: https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

Bennani, Y. (2018, May 31). Retrieved from Data Science applied to the retail industry : https://www.margo-group.com/en/news/data-science-applied-to-the-retail-industry-10-essential-use-cases/

Greg Linden, B. S. (2003). *Amazon.com Recommendations:Item-to-Item Collaborative filtering.* IEEE Computer Society.

Harrison, O. (2018, September 10). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Retrieved from Towards Data Sceince: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Lillian Pierson, P. (n.d.). *Building a Recommendation System with Python Machine Learning & AI*. Retrieved from Linkedin Learning: https://www.linkedin.com/learning/building-a-recommendation-system-with-python-machine-learning-ai/model-based-collaborative-filtering-systems?u=2071660

M, M. (n.d.). *Medium*. Retrieved from https://medium.com/@madasamy/introduction-to-recommendation-systems-and-how-to-design-recommendation-system-that-resembling-the-9ac167e30e95

Malaeb, M. (2016, December 24). *Singular Value decomposition (SVD) in recommender systems for Non-Math-statistics-programming wizards*. Retrieved from A Medium Corporation [US]: https://medium.com/@m\_n\_malaeb/singular-value-decomposition-svd-in-recommender-systems-for-non-math-statistics-programming-4a622de653e9

Mayeesha, T. (2018, January 29). *Recommending Animes Using Nearest Neighbors*. Retrieved from A Medium Corporation[US]: https://medium.com/learning-machine-learning/recommending-animes-using-nearest-neighbors-61320a1a5934

*Recommendation Systems – How Companies are Making Money*. (n.d.). Retrieved from https://sigmoidal.io/recommender-systems-recommendation-engine/