Assignment 2

SAMAN MANSOOR	**F2019332020**
DATAWAREHOUSE	
Submitted to	**Mam Alishba**

Report

Review Base Recommendation system of Amazon

Abstract:

Recommendation systems algorithms suggest the right things for users based on data. They make a lot of money in the modern e-commerce industry. 35% of Amazon web sales are made with their recommended products. This study aims to create an app to recommend Amazon users' clothing with user rating history, product images and product title text. Many in-depth learning models built on both easily accessible data and engineers lead to a multi-step recommendation system. Tableau and web application are used to display results, as well as test ratings. A product recommendation system based on product review information and metadata history was used in our project. The main goal of our recommendation system is to predict the amount of user feedback that a product will provide. We used an integrated filtering model in both user-based and object-based strategies, a matrix factorization model and a Network Inference graph model as our estimation models. We tested the effectiveness of these models in the Amazon Product co-purchasing Network metadata Dataset. We also discussed the pros and cons.

Overview

Introduction:

Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user. Recommender systems have become increasingly popular in recent years, and are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in genera etc. Recommender systems supply users with new suggested items, but they are sometimes considered as black boxes where no explanatory information about them is provided. Thus these recommendations could be accompanied by explanations that describe why a specific item is recommended. Explaining the recommendations usually make it easier for users to make decisions, increasing conversion rates and leading to more satisfaction and trust in the system. An explanation is a description that justifies the recommendations and makes users better realize if the recommended item is relevant to their needs or not. Now I am work on recommendation system on Amazon in which we discuss about product reviews, item etc. Recommender systems are used by E-commerce sites to suggest products to their customers and to provide consumers with information to help them decide which products to purchase. The products can be recommended based on the top overall sellers on a site, on the demographics of the consumer, or on an analysis of the past buying behavior of the consumer as a prediction for future buying behavior. Amazon currently uses item-item collaborative filtering, which scales to massive datasets and produces high quality recommendation systems in real time. This system is a kind of an information filtering system which seeks to predict the "rating" or preferences which user is interested.

Datasets:

Datasets The data was collected by crawling Amazon website including product metadata and review information. There are 548,550 different products. The dataset includes various information for each product and we extract the ASIN, title and review information for each product. There are 7,593,244 unique reviews extracted. From all the review information data, we obtained customer ID, rating score. By extracting user information from product review section, we have 2 1,555,170 unique users extracted, who gave rates and reviews to the 548K products. The grand average for user review rating is about 4.17.

Table 1 includes the general information for Amazon dataset.

Type	Number
Products	548,550
Reviews	7,593,244
User IDs	1,555,170

Table1: Amazon co-purchase Network Dataset Information

https://www.kaggle.com/arhamrumi/amazon-product-reviews?select=Reviews.csv **We use this dataset from kaggle**

Context

This dataset contains more than 568k consumer reviews on different amazon products. This dataset is also available on other dataset related sites, but I found it useful and shared it here

Content

This dataset contains the following attributes:

Total Records: 568454
Total Columns: 10
Domain Name: amazon.com File
Extension: CSV

Challenge

The challenge for this project is to create an apparel-specific recommender system that is personalized to an Amazon user which aims to enhance customer experience. Personalized recommendation based on user preference has a higher likelihood of conversion than general recommendations. In order to meet this challenge, we first have to figure out which machine learning algorithms to use in order to create an apparel recommender system for specific Amazon users. We would need to figure out which features to use for this task. Additionally, we would like our recommender system to recommend similar items relative to the item that a user is currently viewing. This task would be based on product features similarity. This would require a separate modeling task from the first task. Lastly, we would need to create a customer-facing product which will provide recommendations to a given user. We would need to assess dashboard tools for this task.

The algorithms which is used in this Recommendation system.

Naive Bayesian classifier has been widely used as a model-based approach for recommender systems. Let's use a video recommender system as an example, and a user's utility is measured by whether a recommended video is clicked by the user. More formally, this recommendation problem can be modeled as estimating the probability of click (or click-through-rate in some literature).

Clustering

We used a K-Means algorithm to determine the number of possible clusters in our data set. We analyzed the inertia of the model up to 50 clusters. The results are shown in Figure below. Although it is challenging to determine the location where the elbow occurs, we settled on 15 clusters. The products from each cluster are highlighted in the two dimensional projection plot on the bottom.

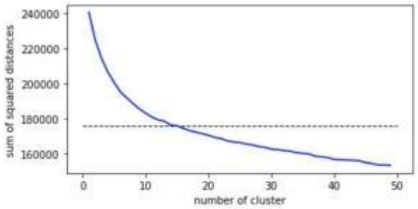


Fig 3: Inertia of the k-means algorithm up to 50 clusters.

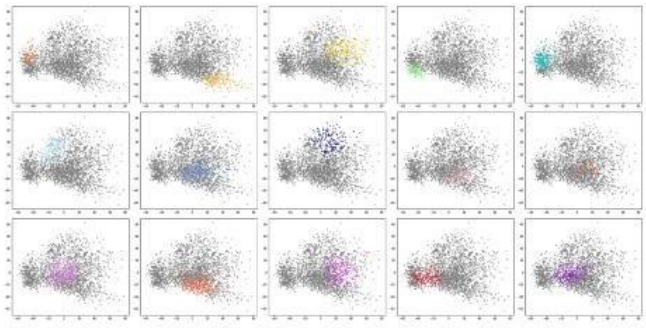


Fig two-dimensional projection highlighting the products that belong to each of the 15 clusters.

Problem

The main problem we don't know the proper acknowledgement of product in which product is good and also the product rating user confuse which product we will buy also the information is overload, and we can call it an information filter system. It greatly influences what we interact with the world: shopping (Amazon, Best Buy), music (Spotify), video (YouTube, Netflix), etc. To build a recommendation system providing recommendations to millions of users with millions of items.

Gathering Known Ratings for Matrix It has been observed that most of the users do not give any ratings. So a research problem arises that how to know whether they are satisfied with the product and how much. There are two ways of taking ratings. When is explicit like asking them after they have purchased or gone through any item. Another way which is predicting their ratings for a particular item based on their preferences on some other item. This method is known as an implicit method of collecting ratings. According to our research, explicit and implicit methods have sufficient gap which gives an opportunity for researchers to start research in this domain.

Cold Start Problem A cold start problem is a problem that arises when no information is found about the user or item in the system. Collaborative filtering recommender system which needs mandatory information.

Lack of Data

Perhaps the biggest issue facing recommender systems is that they need a lot of data to effectively make recommendations. In the world of the recommender system, it is common practice to use a publicly available dataset from a different environment in order to evaluate the efficiency of recommendation algorithms. [35] These data sets are very important and are used as a benchmark to develop new recommendation algorithms. Most of the top companies like Google, Amazon, make good recommendation because they are having a lot of consumer user data so recommender system firstly needs consumer/item user data (from different sources), then it must perform some statistical analysis based on some procedure (User behavior observation or events), and then the Recommendation algorithm does its work. If we will be having more consumer/item data we will be in a situation with the help of the recommender algorithm to have better recommendations

Explanation interface

An explanation interface is a representation of the explanation provided for a recommended item suggested by a recommender system. There are various explanation interfaces used in the literature, some of them are traditional and can be applied and adapted to all domains, and others are specific for each domain. Some of the explanation interfaces presented contains data in the form of ratings. When creating an educational dashboard for students, it is hard to ask students, especially under the age of 18, to frequently rate items as it would be a boring task for them and there would be a risk that they will stop using our system. As a result, these kinds of interfaces don't fit in our case as we can't have data in the form of ratings. Some examples of the most commonly used explanation interfaces for recommender systems are described below.

Data cleaning

To bring the data into a consistent format, steps taken are:

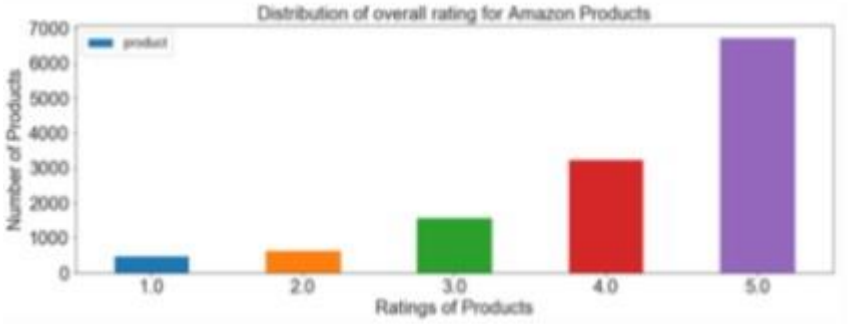
- Drop unnecessary columns
- Drop duplicate records
- Check for invalid data
- Check ranges for applicable columns (such as ratings between 1 and 5) Deal with missing values and outliers

Exploratory Data Analysis

It's good practice to know the features and their data types and to take a look at the data distribution. Plotting the data can provide insights into the patterns that the data follows. 'Patio, Lawn & Garden' product category dataset is used for plotting graphs 1 to 6.

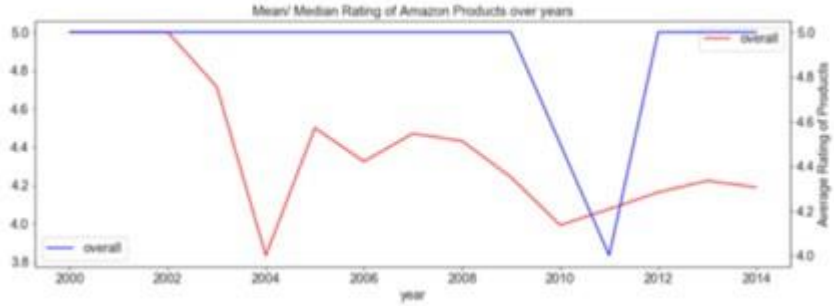
1) Distribution of overall product ratings

Many users have given a rating of 5 to products followed by 4 and 3 whereas very few users have given a low rating of 1 or 2.



Mean and median of ratings over years:

Looking at below plot, we can infer that over the years 2000 to 2014, the mean rating of the products has reduced. Median of ratings given to products remains at 5 from 2000 to 2014 except for years 2010 and 2011

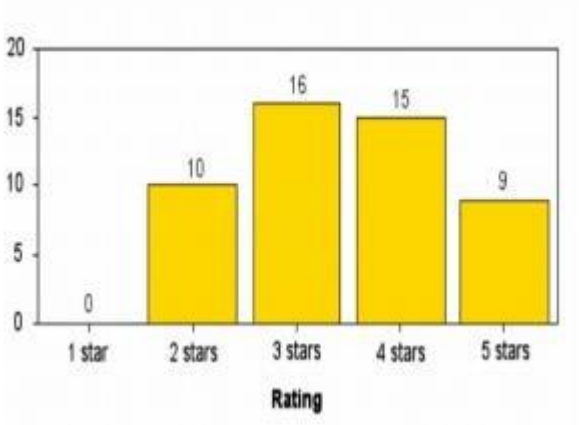


Distribution of ratings per user

Distribution of ratings per user shows a long-tail normal distribution. Total number of users in Garden and Patio dataset is 1685. Maximum number of ratings given by single user is 63 and minimum number of ratings by single user is 1. On an average a user gives 7.55 ratings on Amazon as per the data present.

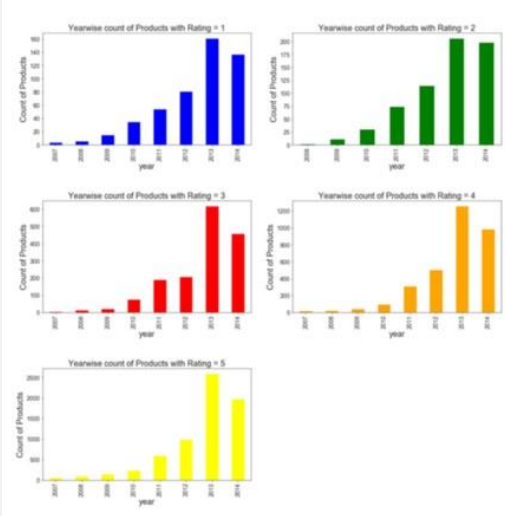
Histogram of ratings

One of the traditional and most used explanation interface is the histogram of ratings that displays the ratings of similar users to the target user for the recommended item. It is usually used for explaining the recommendations of suggested movies or items to buy. It could be a histogram representing the ratings from 1 to 5 separately as in Figure 1 or a histogram with grouping representing a group of good ratings and another of bad ratings



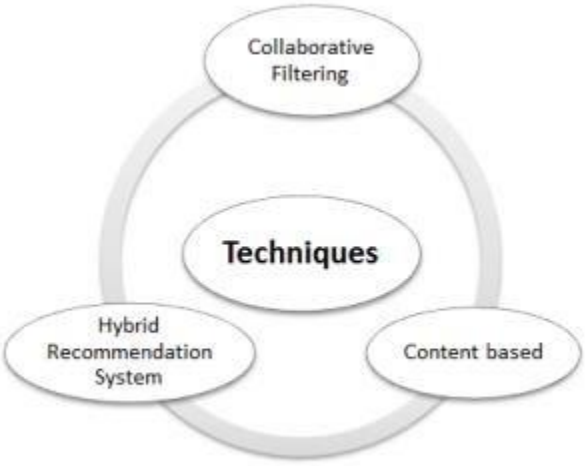
Year wise count of Amazon products with particular rating

Below plot shows that number of products with ratings ‘1’, ‘2’ or ‘3’ is significantly less over the years as compared to those with ratings ‘4’ and ‘5’.



Methods USED IN RECOMMENDATION SYSTEM

The Recommender systems are categorized on the basis of rating techniques used by the system. The way a systems takes ratings or predicts user preferences defines the Recommender systems. Figure describes the basic techniques used in recommender systems. In the coming sections, we are going to describe them. In this paper, we have given a brief description of the basic techniques used in recommender systems. Other techniques are also there which are combined to generate hybrid recommender systems. Figure describes them.



Collaborative Filtering

The collaborative filtering system is a system where we take input from the user and use those inputs to create relations with other users and items. For example, any user in a collaborative filtering system is asked to rate a particular item. Similarly, other users also rate different items. In this way, we get a useritem rating matrix. The rating done by the user gives historical data and choices of the user. Through choices of users, we can create a profile of the user. Then we can calculate what new items we can purchase on the basis of his history and his profile.

Content based

Content-based system requires the item to the user based on the description of the item rather than the history of the user. These types of recommender systems are used in newspapers and article recommendation systems. These recommender systems may have little information about the user. Figure 3 explains the process of recommendation. These recommender systems do not have a history of the user. The most common technique used by content-based recommender systems is item profiling. This popularity of the item is recorded. User profiling is also done sometimes with little basic information about the user which system has. Item profiles are based on the properties of items. The recommender system uses the properties to recommend items to different users. Unlike collaborative filtering recommender systems, they do not have rating information. User profiles can also be created from the

likes and dislikes of users to a particular item. [9, 17] When we want to recommend any item to the user, user interests are compared against item properties. This is also known as content features of items which includes the following:

- The system contains a big database of the item to be recommended. This database consists of features of items. This database is known as an item profile database.

- Users provide little information about their preference likes and dislikes to the system and with this little information the system builds a user profile.

- The recommendation is done on the basis of a comparison of item profile with user interest. One can make better-personalized recommendations by means of utilizing the elements of gadgets and users. An object profile is defined by way of its essential features. For example, a book can be described using its title, genre, language, publisher, cost etc. Using the weighting procedure, similarity can be calculated between items. In some domains, we can represent elements by means of Boolean values while in others we can represent the values using a set of restrained values. Consider the example of the newspaper where we analyze the newspaper articles on the basis of the exceptional form of topics. Boolean cost is indicative of whether a phrase is present in the article or not. Integer cost may want to define the categorical way the range of time of word appears in an Article. This method gives a successful recommendation in content-based recommender systems without using explicit ratings.

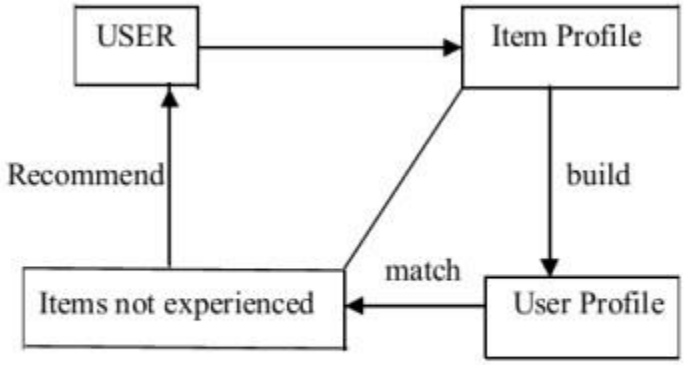


Figure 3. Content based Process

Hybrid Recommendation System

This system recommends on the basis of the mixture of some techniques. This is one of the recent trends in the recommendation system. The table given shows different types of recommendation techniques. The Hybrid makes use of a combination of these recommendation techniques. The recommendation system by Facebook seems to make use of all these techniques. The hybrid recommendation takes advantage of all the techniques but once should be very careful as hybrid may involve a lot of computation and may give conflicting results.

Conclusion

From creating this system we have discovered that when it comes to optimization, data engineering on selective data is more effective than tuning parameters on various models, which often takes a lot more computing power and resources because of the scale of this dataset. We also discovered that with deep learning models, appropriate drop-outs implemented had the best performance in terms of accuracy. Due to the size of the dataset, the use of AWS was essential to the project. From this we learned that the usage of cloud-computing tools such as Amazon Sage Maker and Data bricks enable quick workflow to provide effective model building and testing. Another goal of the project was to build a customer-facing data product which will provide recommendations to a given user. For this, Tableau provided us with an easily buildable and deployable dashboard that is intuitive for customer use. We would like to note that the product is not ready for deployment as some recommendations were not ideal after reviewing the output in the dashboard. For example, some items recommended based on image similarity were from different product categories. Also, to deploy this product, we need to create a better design for scalability as much as possible to enable iteration and proficiency.

Implementation:

```
print("Total data ")
print("-"*40)
print("\nTotal no of ratings :",dataset.shape[0])
print("Total No of Users   :", len(np.unique(dataset.UserId)))
print("Total No of products  :", len(np.unique(dataset.ProductId)))
```

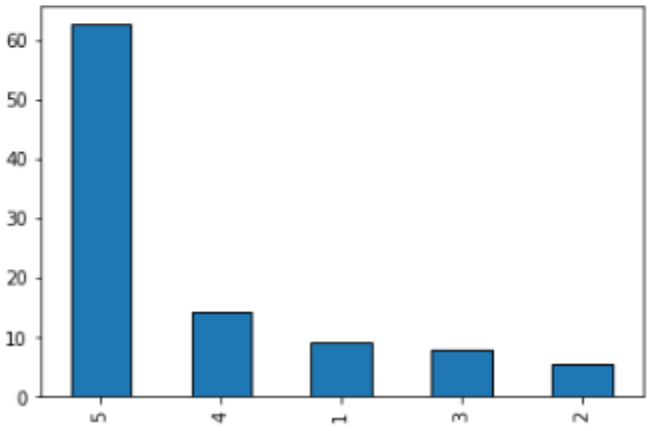
```
Total data
----------------------------------------

Total no of ratings : 35617
Total No of Users   : 31563
Total No of products  : 4494
```

Distribution Review:

```
print(dataset['Score'].value_counts())
rating_pct = dataset['Score'].value_counts()/len(dataset) * 100
rating_pct
rating_pct.plot.bar()
plt.show()
```

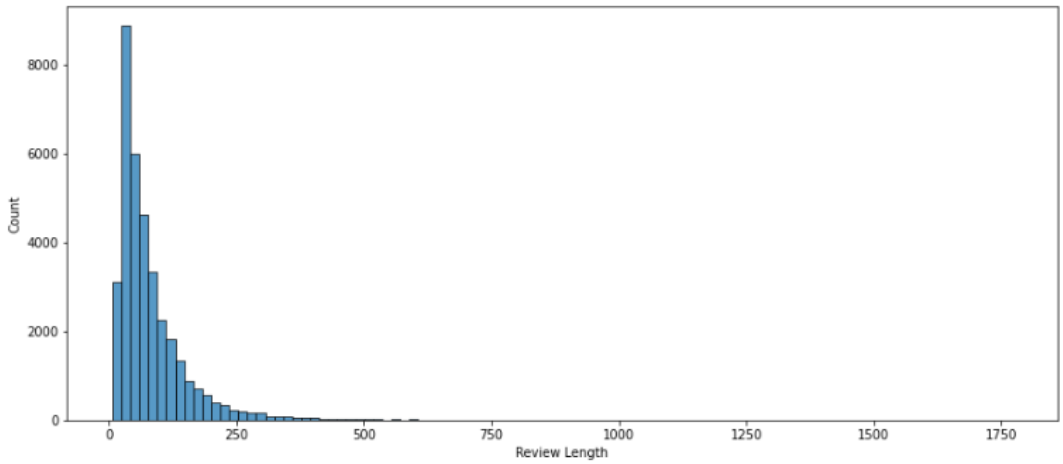
```
5    22282
4     5128
1     3319
3     2870
2     2018
Name: Score, dtype: int64
```



Word per Review:

```
sns.histplot(WordsPerReview,bins = 100)

plt.xlabel('Review Length')
plt.show()
```




```
[62] from wordcloud import WordCloud
     wordcloud = WordCloud(
                 background_color = 'white',
                 max_font_size = 100,
                 max_words = 100,
                 width = 800,
                 height = 500
                 ).generate(txt)


     plt.imshow(wordcloud,interpolation = 'bilinear')
     plt.axis('off')
     plt.show()
```

Clean Text:

```
[67]      Review = str(Review).lower() # convert to lowercase
          Review = re.sub('\[.*?\]', '', Review)
          Review = re.sub('https?://\S+|www\.\S+', '', Review) # Remove URls
          Review = re.sub('<.*?>+', '', Review)
          Review = re.sub(r'[^a-z0-9\s]', '', Review) # Remove punctuation
          Review = re.sub('\n', '', Review)
          Review = re.sub('\w*\d\w*', '', Review)
          return Review
```

```
[68] dataset['Review']
```

```
24750    Our dogs just love them.  I saw them in a pet ...
24749    My dogs loves this chicken but its a product f...
2774     We have used the Victor fly bait for 3 seasons...
2773     Why is this $[...] when the same product is av...
1243     I just received my shipment and could hardly w...
                               ...
32548    I was a little nervous about giving this as a ...
32547    This is a resubmittal of a previous review.  O...
32546    Cheesecake arrived within a week of ordering. ...
1477     This coffee supposedly is premium, it tastes w...
5702     Purchased this product at a local store in NY ...
Name: Review, Length: 35617, dtype: object
```

Collaborative Filtering Result:

```
[84] algo = KNNWithMeans(k=5, sim_options={'name': 'pearson_baseline', 'user_based': False})
     algo.fit(trainset)
```

```
Estimating biases using als...
Computing the pearson_baseline similarity matrix...
Done computing similarity matrix.
<surprise.prediction_algorithms.knns.KNNWithMeans at 0x7fa26d6e3b90>
```

```
[85] # run the trained model against the testset
     test_pred = algo.test(testset)
```

```
[38] test_pred
```

RMSE(ROOT MEAN SQUARE ERROR)

```
[86] print("Item-based Model : Test Set")
     accuracy.rmse(test_pred, verbose=True)
```

```
Item-based Model : Test Set
RMSE: 0.9634
0.9633619489399123
```