Final Project Report

E-Commerce Product Recommendation System

ANKIT SATI (AS14128)

KRINA BHIMANI (KB3687)

AKSHAT JAIN (AJ3186)

SHASHAANK GUPTA (SG6661)

NAMAS MANKAD(NKM9348)

1. Introduction

One of the most important places where data is used these days to inform decisions and save costs for customers is the E-Commerce web sector. The usage of recommendation algorithms in the e-commerce industry, where the primary goal is to enhance product sales based on past purchases and searches, is nothing new. Many applications only use the products that customers buy and openly rate to address their preferences, but they may also use additional characteristics like things viewed, segment information, subject interests, and favorite craftsmen. Utilizing the aforementioned data to train a model that improves with each prediction it makes is a prevalent trend among most recommendation algorithms. The formula adds up the items from these comparable clients, subtracts the items the client has actually purchased or valued, and then distributes the surplus items to the clients. Cooperative and content sifting models are two instances of these calculations that are well-known.

Our project's initial objectives were multifaceted: This project's main objective is to put these recommendation algorithms into practise, test them on a dataset of E-Commerce products, and compare the results to determine which algorithm performs better, why, and what qualities it needed to do so. The project's second objective was to develop an accurate price prediction model that would allow customers to buy goods when they would be most affordable. The main goal of this project was to work with a sizable real-world dataset using multiple models and techniques that were learned in class, and then compare the results using in-depth analysis.

2. Related Work

For this data set, we looked through numerous white papers and research studies. Later on, we will go into more depth about some of them that we have used. The identical dataset that we used in this experiment has been the subject of a Visually-Aware Recommender System developed by authors in [1]. The next section contains a thorough description of this dataset. This is because visual details are too important in determining whether a customer would like an item or not. When purchasing a shirt with a photo on it, people frequently read the information and other crucial details at the end. They want to build outwardly conscious recommender frameworks that are flexible, customizable, transiently progressing, and interpretable in order to handle these four problems. In order to achieve their goals, new models had to be developed; they have thus shown the two views, worldly and visual information.

For their Netflix movie posters, the authors in [2] have developed a Scoring-Based Recommendation System. The main goal of the Netflix personalised suggestion framework has been to put the appropriate movies in front of the right people at the right time. With this process, they have made tremendous progress in tailoring the definition of fine art for their proposals, which has significantly improved how our people discover new material.Finally we could not find any related work for our price prediction model so we had to explore different areas to predict the prices to get an estimate for our methods. For instance the stock market relies on many

features from mood to the real-time changes in the business scape. The features that we had to use were contrasting but the results we wanted to achieve were quite similar. We tried to make a prediction to influence a buy at a time where the cost of a commodity is lowest.

3. Dataset

We have developed a system for electronic products sold through online commerce. Product Metadata Dataset and Product Reviews Dataset are the 2 datasets we used. This dataset includes metadata and 150 million product reviews from Amazon that cover the period from May 1996 to July 2014. This dataset consists of links (also viewed/also bought graphs), product metadata (descriptions, Category information, price, brand, and image attributes), and reviews (ratings, text, helpfulness votes).

The first dataset includes information on user-submitted reviews, ratings, and overall star ratings for products that are listed on the website. ReviewerId, asin, reviewerName, helpful, reviewText, summary, unixReviewTime, and reviewTime are some of the fields that are present. The product id, which distinguishes a product sold on Amazon, is shown in this case by the field asin. The remaining fields are all self-explanatory. A collaborative recommendation model can be implemented using only the first dataset because this method finds user clusters without taking item attributes into account. To apply the content based models, however, we must combine the second dataset with the first one. The first dataset, which is in json format, is 2GB, and the second, which is in json format, is 12GB.

Figure 1: Product Reviews Dataset


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.
He is having a wonderful time playing these old hymns.  The music  is
at times hard to read because we think the book was published for
singing from more than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


The second dataset comprises information about the item features, such as the item's title, the items that are purchased together, the items that are seen together, the items that are endorsed, the item's category, the images that the vendor uses to display the product, etc. Keep in mind that the asin field, which is the product id to specifically identify the product, is also included in this second dataset.

4. CRISP-DM Approach

4.1) Business Understanding

Nowadays, a lot of research is done before purchasing a single item because there are so many data sources at the disposal of the customer. The main goal of this project is to minimise the amount of time and money spent by clients on research before making a purchase. It is crucial for online retailers like Amazon to help customers locate content and goods in increasingly complicated computerised shopping malls. Clarifying or defending recommendations to clients may help them become more frank and reliable. Based on the interests of the customer, these algorithms fundamentally alter, displaying programming books to a computer programmer and kid's toys to another mother. The navigation and change rates, two important components of email and Web-based advertising viability, utterly outperform those of untargeted information, such pennant alerts and top-merchant lists.

Figure 2: Product Metadata Dataset


```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "price": 3.17,
  "imUrl": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "related":
  {
    "also_bought": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O",
"0000031909", "B00613WDTQ", "B00D0WDS9A", "B00D0GCI8S", "0000031895",
"B003AVKOP2", "B003AVEU6G", "B003IEDM9Q", "B002R0FA24", "B00D23MC6W",
"B00D2K0PA0", "B00538F5OK", "B00CEV86I6", "B002R0FABA", "B00D10CLVW",
"B003AVNY6I", "B002GZGI4E", "B001T9NUFS", "B002R0F7FE", "B00E1YRI4C",
"B008UBQZKU", "B00D103F8U", "B007R2RM8W"],
    "also_viewed": ["B002BZX8Z6", "B00JHONN1S", "B008F0SU0Y",
"B00D23MC6W", "B00AFDOPDA", "B00E1YRI4C", "B002GZGI4E", "B003AVKOP2",
"B00D9C1WBM", "B00CEV8366", "B00CEUX0D8", "B0079ME3KU", "B00CEUWY8K",
"B004FOEEHC", "0000031895", "B00BC4GY9Y", "B003XRKA7A", "B00K18LKX2",
"B00EM7KAG6", "B00AMQ17JA", "B00D9C32NI", "B002C3Y6WG", "B00JLL4L5Y",
"B003AVNY6I", "B008UBQZKU", "B00D0WDS9A", "B00613WDTQ", "B00538F5OK",
"B005C4Y4F6", "B004LHZ1NY", "B00CPHX76U", "B00CEUWUZC", "B00IJVASUE",
"B00GOR07RE", "B00J2GTM0W", "B00JHNSNSM", "B003IEDM9Q", "B00CYBU84G",
"B008VV8NSQ", "B00CYBULSO", "B00I2UHSZA", "B005F50FXC", "B007LCQI3S",
"B00DP68AVW", "B009RXWNSI", "B003AVEU6G", "B00HSOJB9M", "B00EHAGZNA",
"B0046W9T8C", "B00E79VW6Q", "B00D10CLVW", "B00B0AVO54", "B00E95LC8Q",
"B00GOR92SO", "B007ZN5Y56", "B00AL2569W", "B00B608000", "B008F0SMUC",
"B00BFXLZ8M"],
    "bought_together": ["B002BZX8Z6"]
  },
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports", "Dance"]]
}
```


Internet business recommendations algorithms frequently work in a difficult climate. For example:

- Because there have been thousands of purchases and appraisals, older clients may have an excess of data.
- A sizable shop might have a sizable customer base, a sizable amount of information, and a sizable number of specific list items.
- Many applications call for the results set to be returned continually in less than a second while yet producing excellent recommendations.
- Because they've just made a few purchases or seen a few product reviews, new customers typically have very

little information.

- Customer information can change at any time. Each collaboration favours crucial client information, and the calculation should respond quickly to fresh information.

4.2) Data Preparation

1st approach:

The Json object was missing from the list when we downloaded the dataset files for the two datasets. Instead, each new line included a new record since each record or row was separated by a new line character. A list of json objects in json file format is required by the majority of tools and algorithms, including MYSQL, Tableau, and K-NN in Rapidminer. You could be unsure of the definition of the list of json objects. To handle these dataset files, we needed each new entry to be present in a list, separated by commas (relate it to a Python list).

So, start looking to see whether the Rapidminor tool has any operators that could change a json file to suit our needs. We looked through a number of extensions, including the Web Automation Extension, Operator Toolbox, and Text Processing, to locate operators that could meet our needs. However, none of these extensions had an operator like that, which would have allowed us to make the JSON object comma separated rather than on a new line.

2nd approach:

We tried using big data tools like hadoop and spark after trying several other tools and failed. To process and clean the data, we utilised the Python plugin for Spark called PySpark. However, the 12GB dataset was still requiring a sizable amount of time.

3rd approach:

Now, thanks to the programming tasks from this course, we moved on to another tool where we gained practical experience. We attempted to preprocess the downloaded json file using Eclipse and Java as the backend language to comma separate each json object. Eclipse, however, was unable to handle a 2GB json input file. For a 2GB json input file, we kept getting the Java Heap out of memory space issue. Keep in mind that this is merely a preprocessing of the 2GB first dataset. As per customary, we attempted to raise Eclipse's heap memory allocation from 512 Mb to 1 GB. Even so, we were unable to preprocess the 2GB input json file. We tried increasing this value from 1GB to 2GB and then from 2GB to 4GB without any luck. The local machine's RAM memory, where we were attempting to preprocess this file, was 8GB. We gave Eclipse the full 8GB of memory to run our Java code. Finally, we had some achievement. Eclipse preprocessed our 2GB json input file this time, although it took close to 10-15 minutes. There was no doubt that the other 12GB json input file would not be preprocessed on a local system.

4th approach (Used in the project)

Finally, by requesting resources with higher processing power, we executed our 2GB file on the HPC Greene computers at NYU. The HPC Greene 2-Core GPU with 32GB internal memory machine we used was able to preprocess the 2GB json file and simply parse it so that it matched our requirements. Finally, using Python on the HPC Greene system, we preprocessed both the 2GB and 12GB json input files. The next phase, data integration and data reduction, is presently in progress.

4.3) Data Integration & Reduction

The aim was to merge both datasets after they had been obtained. As a result, we integrated the datasets into a dataframe format using inner join in Python. Then we made the decision to remove the strongly connected columns, such as total sales in relation to the quantity of a certain product sold, etc. Then we made the decision to move forward with items that had at least been purchased five times and with customers who have done so at least five times.
This was very close the deadline and luckily worked for us in the first attempt.

4.4) Data Modeling

As was noted in the presentation, we experimented with a variety of models that were slightly altered to fit the characteristics being used. We looked into new graph-based algorithms in addition to using the existing techniques for recommendations. We chose the program's flow after reviewing all the algorithms and testing them out. Two fields, customer ID and product ID, were provided to the model. Our model's output will depend on these fields to provide results.

If you are a new customer, the model will make recommendations based on demand. Now, the application will suggest collaborative filtering-based outputs if you enter an existing customer ID. However, adding product IDs also means that you will receive outputs based on association mining and content-based filtering if you purchase specific things. We made an effort to create the model in a way that can include all of the algorithms at once.

The interactive nature of this input-output based approach was designed with the intention of making it more user-friendly.

4.5) Data Evaluation

We tried to cross-verify the results by entering the input ID and printing the images from the dataset's URL to see if the product that has been added is indeed providing recommendations of the products that are related to the original input product because it was difficult to actually confirm the answers provided by the algorithm.

As you can see below, we simply added a product to the cart to observe which products were returned and whether the projected numbers had come true or not.

```
    print ("Based on product reviews, for ", df3["asin"][lentrain + i] ," average rating is ",df3["overall"][lentrain + i])
    print ("The first similar product is ", df3["asin"][first_related_product] ," average rating is ",df3["overall"][first_rel
ated_product])
    print ("The second similar product is ", df3["asin"][second_related_product] ," average rating is ",df3["overall"][second_
related_product])
    print ("-------------------------------------------------------------")
('Based on product reviews, for ', 'B00AE0790U', ' average rating is ', 4.1457286432160805)
('The first similar product is ', 'B007RTR9DS', ' average rating is ', 2.9884393063583814)
('The second similar product is ', 'B002WTC37A', ' average rating is ', 4.3097345132743365)
-------------------------------------------------------------
('Based on product reviews, for ', 'B00AE07BDU', ' average rating is ', 4.217821782178218)
('The first similar product is ', 'B005XIDZHO', ' average rating is ', 3.4554455445544554)
('The second similar product is ', 'B0056GDG90', ' average rating is ', 4.0175438596491224)
-------------------------------------------------------------
('Based on product reviews, for ', 'B00AE07FQI', ' average rating is ', 3.8316831683168315)
('The first similar product is ', 'B005XIDZHO', ' average rating is ', 3.4554455445544554)
('The second similar product is ', 'B001LF4I8I', ' average rating is ', 4.2407407407407405)
-------------------------------------------------------------
('Based on product reviews, for ', 'B00AE07H0M', ' average rating is ', 3.875)
('The first similar product is ', 'B002WTC37A', ' average rating is ', 4.3097345132743365)
('The second similar product is ', 'B004INUWX0', ' average rating is ', 3.6000000000000001)
-------------------------------------------------------------
('Based on product reviews, for ', 'B00AE07IEM', ' average rating is ', 4.2752293577981648)
('The first similar product is ', 'B002TSA90M', ' average rating is ', 4.4521739130434783)
('The second similar product is ', 'B0089JVDPK', ' average rating is ', 4.3931623931623935)
-------------------------------------------------------------
('Based on product reviews, for ', 'B00ANF1GK6', ' average rating is ', 4.1812865497076022)
```

5. Algorithms Used

Content Based Recommendation: In this method, we created a content-based engine whose main concept was to use a product's features to locate clusters of related products. We used a variety of attributes from the second Product Metadata Dataset, such as things similar to, products bought with, items seen before, items seen after, items zoomed in, etc., to describe the features of each product. Each row will be a product of the same data matrix, and these variables became our characteristics for each product, acting as columns in a data matrix. Now, depending on the characteristics of the product, we used the KNN method to discover the unknown clusters. In the Collaborative Based method, whenever a user purchases a product, all the other items that are part of the cluster in which the purchased product is located are also recommended to that user. The main distinction, however, is that the data matrix was filled up using product attributes rather than taking into account sentiment scores, ratings, or reviews.

Demand Based Recommendation: The fundamental tenet of this strategy is that every user who has registered with the organisation should be given recommendations for products based on what has been most frequently sold or purchased. The problem with this strategy, as you may have already guessed, is that newly released products don't get recommended at all, which results in a cold start for those new products. However, from a learning viewpoint, we still used this because it can provide us with some insightful information about which electrical item has been purchased more frequently than other comparable things.

So, based on the product metadata, we choose to recommend the top 5 highly rated products. The methodology is rather straightforward; after integrating the two datasets, we perform sentiment analysis on the evaluations to determine if a product is scored favourably or unfavourably. This is done for each review that is accessible for that product, and then we repeat the process for all the other products. As we go along, we keep note of how many favourable reviews each product has received, and the top five that have the highest total will be the ones that all users are advised to buy. As can be seen in the graphic below, the top 5 items with the identifier "B0002L5R78" received the most favourable ratings (9487), while the products with the identifier "B0007E7JU" received the least favourable reviews (3528).

```
productId
B0002L5R78    9487
B0001FTVEK    5345
B000I68BD4    4903
B000BQ7GW8    4275
B00007E7JU    3523
Name: Rating, dtype: int64
```

Association Based Recommendation: In this method, we initially used the apriori algorithm to determine the validity of each product before using the association rule mining library to further determine the outcomes for the input products or list of products. We used the affiliation rules of the structure A => B for our motivations. This suggests that all single-thing linkages were observed. In other words, given that the dynamic client has evaluated item A, what is the likelihood that the dynamic client would rate item B? All of these one-item correlations or associations, along with the corresponding confidence scores between

all n goods in the dataset, were compiled into a square data matrix. The user or customer is then treated as a vector in n-dimensional space at that point. The items that the user is most likely to rank based on the ones they have previously rated are obtained by multiplying the matrix by the vector. Its suggestion vector can surely be used to organize a client's preferences. This approach has the benefit of being incredibly rapid. Building the data matrix just takes a very brief amount of time, and any recommendations that follow are then given right away. Additionally, it yields reliable results, such as your judgement regarding recommending someone based on what you have seen.

Collaborative Based Recommendation: By detecting the grouping of related products based on a similarity metric, we implemented an item-item collaborative recommendation engine in this method, where the main concept is to look at item to item relationships instead of user to user relationships. We first transposed the data matrix on product ids, or the "asin" column/field, as indicated in the dataset part of this report, in order to apply this approach. After transposing the data matrix, we preserved the columns representing customers who had purchased, reviewed, and rated a certain product. A cell of this data matrix thus indicates the total number of times the same user has purchased the product, the emotion score of the review from MIT's Vader collection, and the total number of star ratings that user has gotten. The normalised range for this cumulative score is [0, 1]. After that, we applied the KNN algorithm to this data matrix in order to identify any unidentified clusters in the dataset. Now, whenever a user purchases a product, all the other products in the cluster where the purchased product is located will be recommended to that user. On this merged dataset, we applied the Collaborative Recommendation engine.

6.Results and Conclusions:

We tested the results on a real-world model, and they were encouraging. A suggestion system is a means to comprehend clients in today's industry and meet their top needs. Making wise decisions and expanding your business are two more benefits.

Here, we used well-known methods and algorithms to learn about the inner workings of a recommendation system. A notion for a graph-based recommendation system also caught our attention, and we believe it to be the direction recommendation systems are headed.

Although we could not deliver on the price prediction model as we thought but that was just down to the time as it was too little to even cross the data gathering stages. This is something that we will look for in the near future once our crawlers give us at least some usable data to land a few results in a given domain range.

7.References:

1. https://towardsdatascience.com/sentiment-analysis-on-amazon-reviews-45cd169447ac
2. https://www.kaggle.com/ssismasterchief/knnbaseline-recommendation-systems-surprise
3. https://towardsdatascience.com/product-recommender-using-amazon-review-dataset-e69d479d81dd
4. http://snap.stanford.edu/class/cs224w-2013/projects2013/cs224w-072-final.pdf