Fashion Recommendation

Ankit Dhokariya
Computer Engineering Department
San Jose State University
San Jose, United States
ankitsachin.dhokariya@sjsu.edu

Parshv Patel
Computer Engineering Department
San Jose State University
San Jose, United States
parshvnileshkumar.patel@sjsu.edu

Varun Patwari
Computer Engineering Department
San Jose State University
San Jose, United States
varun.patwari@sjsu.

***Abstract*—Here we have implemented two types of recommendation engines, text based recommendation engine and visual based recommendation engine. We found that a visual based recommendation engine performed better than a text based recommendation engine.**

Keywords— Content based, Recommendation engine, Text processing, Bag of Words, TF_IDF, IDF, Word2Vec, Convolutional Neural Network.

I. Introduction

Recommendation engines are the core of the businesses that are customer focused. They are responsible for bringing in a major chunk of revenue to the company. Amazon's 35% revenue is from product recommendations. So we believe by implementing Fashion Recommendation, we can learn more about the in-demand use case of recommendation engines.

The recommendation engine we build in this project is content based recommendation as collaborative filtering needs item/user information which might be tough to find because of privacy concerns.

We plan to build this feature:

Product:

Fig. 1.1 Product

Recommendations:

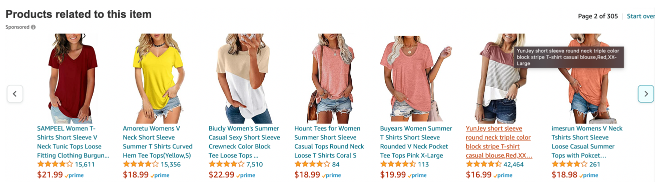

Fig. 1.2 Recommendations

II. System Design & implementation

A. Algorithms

We have built 2 types of recommendation engines:

1) Text based recommendation engines
   a) BoW
   a) TF-IDF
   b) IDF
   c) Average Word2Vec (Text Semantics)
   d) IDF weighted Word2Vec (Text Semantics)
   e) Weighted similarity using brand and color
2) Visual features based

We have built these 2 types of recommendation engines because in some cases text based engines might give better results (for books) and in some cases visual features based engines might give better results (for apparel).

We have considered the basic techniques of text pre-processing for Text based recommendation engines with the exception of Word2Vec, we did this because our main goal was to build a robust recommendation engine and not complicate things by applying more advanced techniques.

B. Technologies & Tools

We have used libraries like sklearn and gensim.models for text preprocessing and Word2Vec.
We have also used keras library to extract features from the product images.

C. System Architecture

The below architecture diagram shows the real world business use case of our recommendation engine.

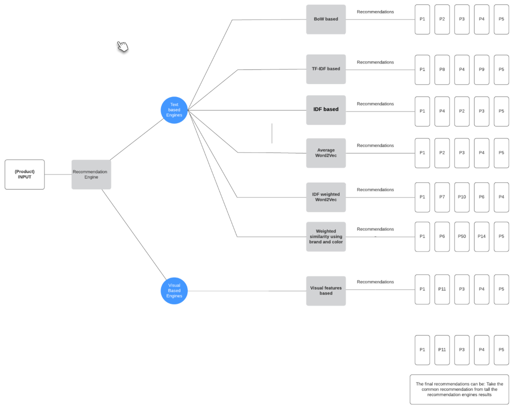

Fig. 2. System Architecture

After all the recommendation engines generate results, the business can customize the final results by applying their business logic on top of it.
Some of the business logic might include:

- Do not show same brand products in the recommendation thus avoiding product bias
- Take the common recommendations among all the results and display the final results (Kind of ensemble way).

III. EXPERIMENTS / PROOF OF CONCEPT EVALUATION

A. Dataset

The dataset consists of a total 183,138 number of records of different products. Each record is defined using 19 features, some of which are ASIN which stands for Amazon Standard Identification Number, the Brand of the product, the Type of the product, Availability of the product, Reviews given by users for the products, 3 different sizes of the product image and Manufacturer of the product and many more.

Of these 19 features, we used only 7 features:

1. ASIN (Amazon standard identification number)
2. Brand (brand to which the product belongs)
3. Color (Color information of apparel, it can contain many colors as a value ex: red and black stripes)
4. product_type_name (type of the apparel, ex: SHIRT/T-SHIRT)
5. medium_image_url (URL of the image)
6. title (title of the product.)
7. formatted_price (the price of the product)

We did this because features like 'sku', 'author', 'publisher', 'availability', 'large_image_url', 'availability_type', 'small_image_url', 'editorial_review', 'model', 'medium_image_url', 'manufacturer', 'editorial_reivew' are quite irrelevant while making recommendation. We got to know this by conducting a survey with our friends on what they think is the feature that they look at while scrolling through products.

Statistics for the above features:

- **Product_type_name**: Out of the 183K records we only have 72 unique types of product names. 167K products are shirts which make up approximately 92% of the total products. The top 10 common product types are:
  - Shirt
  - Apparel
  - Books from 1973 and later
  - Dress
  - Sporting goods
  - Sweater
  - Outerwear
  - Outdoor recreation product
  - Accessory
  - Underwear
- **Brand**: The dataset consists of products from 10,577 different brands. The top common brands are:
  - Zago
  - XQS
  - Yayun
  - YUNY
  - XiaoTianXin-women clothes
  - Generic
  - Boohoo
  - Alion
  - Abetteric
  - TheMogan

  There are around 150 products that did not have a brand mentioned.
- **Color**: We have 7,380 different colors mentioned for the products. Not all products have color mentioned. Only 35.4% (64,956) of the products have color mentioned to describe them. Out of which 7.2% of the products are black in color, which is the most frequent color. Other common colors of the products are:
  - White
  - Blue
  - Red
  - Pink
  - Grey
  - Green
  - Multicolored
- **Price**: From the complete dataset only 28,395 (which makes 15.5% of the whole dataset) products are provided with their price information.

- **Title**: All of the products in the dataset have a title. This feature about the product is short and informative and gives a fair description of what the product is. We used the title extensively in this project.

B. Data Preprocessing

We have 2325 products that have the same title but different colors. These shirts are exactly the same except that they are different in size.

Fig. 3.1 Shirts with same colour

Fig. 3.2 Shirts with different colour

These shirts are exactly the same except in color.
After looking at the data thoroughly, here are a few more examples of duplicate titles that differ only in the last few words.

1) Title 1
    a) woman's place is in the house and the senate shirts for Womens XXL White
    b) woman's place is in the house and the senate shirts for Womens M Grey
2) Title 2
    a) tokidoki The Queen of Diamonds Women's Shirt X-Large
    b) tokidoki The Queen of Diamonds Women's Shirt Small
    c) tokidoki The Queen of Diamonds Women's Shirt Large
3) Title 3
    a) psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for women Neon Wolf t-shirt
    b) psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for women Neon Wolf t-shirt
    c) psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for women Neon Wolf t-shirt
    d) psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for women Neon Wolf t-shirt

We first sorted the data in alphabetical order of the titles. Then we removed titles that are adjacent and similar.

But there are some products whose titles are not adjacent but very similar.

Examples:

1) Title 1
    a) UltraClub Women's Classic Wrinkle-Free Long Sleeve Oxford Shirt, Pink, XX-Large
    b) UltraClub Ladies Classic Wrinkle-Free Long-Sleeve Oxford Light Blue XXL
2) Title 2
    a) EVALY Women's Cool University Of UTAH 3/4 Sleeve Raglan Tee
    b) EVALY Women's Unique University Of UTAH 3/4 Sleeve Raglan Tees
    c) EVALY Women's New University Of UTAH 3/4-Sleeve Raglan Tshirt

C. *Algorithms*

There are mainly two types of of algorithms that we implemented:

1. Text based Recommendation Engines
2. Visual features based Recommendation Engines

1. Text based Recommendation Engines

a) Bag of words on product titles

We separated words from all the product titles and formed vectors of each title. We used these vectors to find similarity between other product title vectors and recommended the ones with the most similar titles.

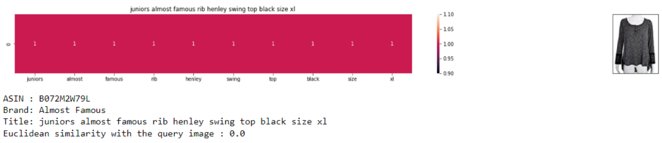

Fig. 3.3 Input Product for Bag of Words Model

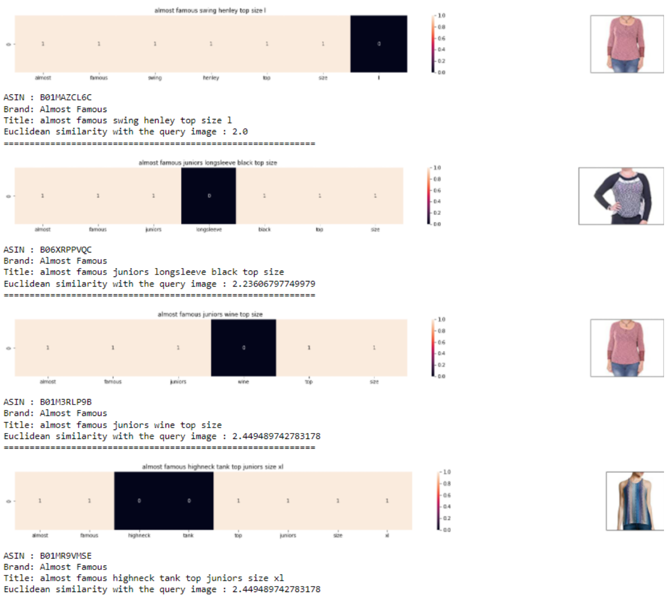

Fig. 3.4 Recommended Products

b) TF-IDF based product similarity

Term frequency times Inverse document frequency is used to scale down the impact of words that occur more frequently and do not contribute in distinguishing a title.

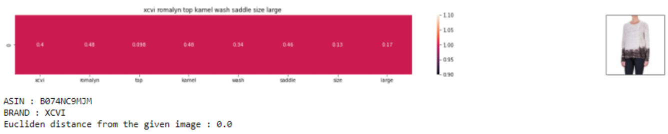

Fig. 3.5 Input for TF-IDF Model

Fig. 3.6 Recommended Products

c) IDF based product similarity

Inverse document frequency is used to find similarity between the vectors to recommend products.

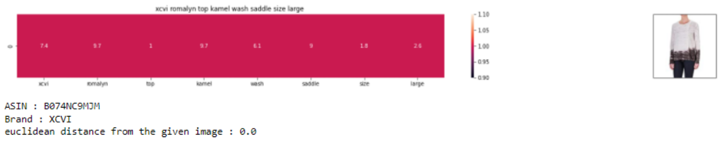

Fig. 3.7 Input for IDF Model

Fig. 3.8 Recommended Products

d) Average Word2Vec product similarity

We used Word2Vec to generate a vector for the product titles. We used these vectors to recommend similar products.

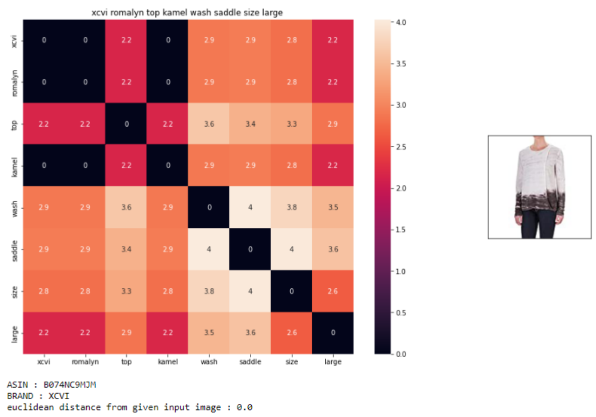

Fig. 3.9 Input for Average Word2Vec Model

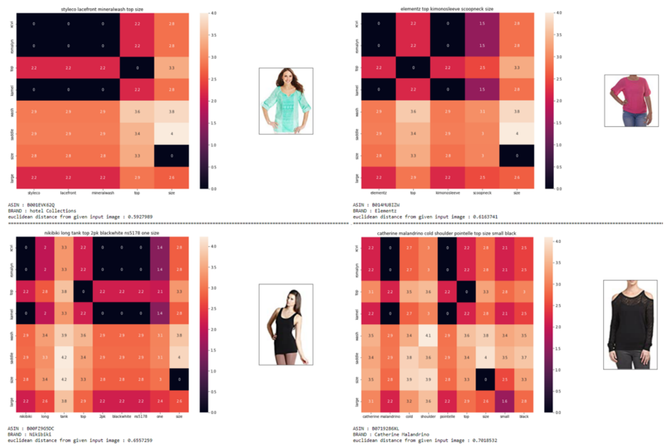

Fig. 3.10 Recommended Products

e) IDF Weighted Word2Vec product similarity

We used Word2Vec to generate a vector for the product titles. While generating vectors from Word2Vec we multiplied the vector provided by Word2Vec with the inverse document frequency of the title. We used these vectors to recommend similar products.

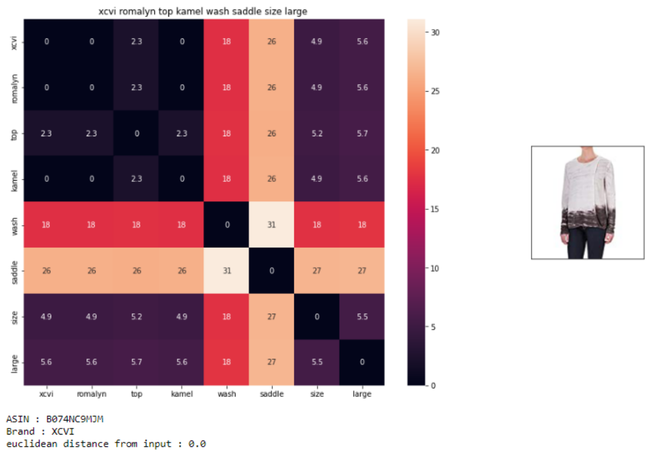

Fig. 3.11 Input for IDF Weighted Word2Vec Model

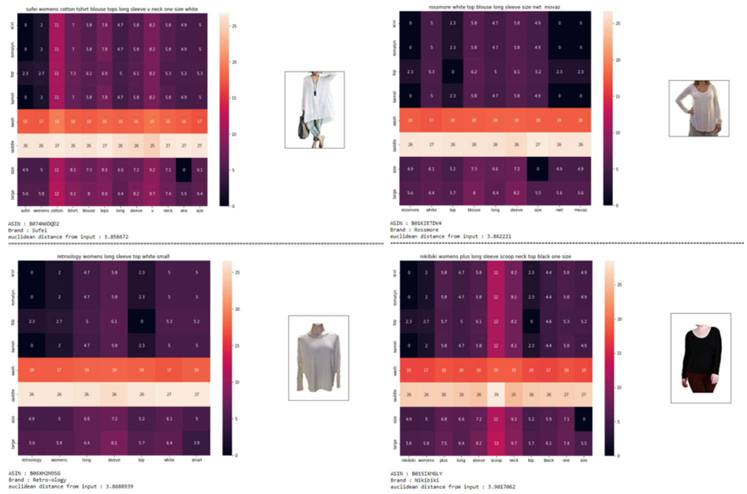

Fig. 3.12 Recommended Products

f) Weighted similarity using brand and colour

We made use of brand and color features to recommend products. We used weighted importance of brand and color while calculating similarity of two products based on the title.

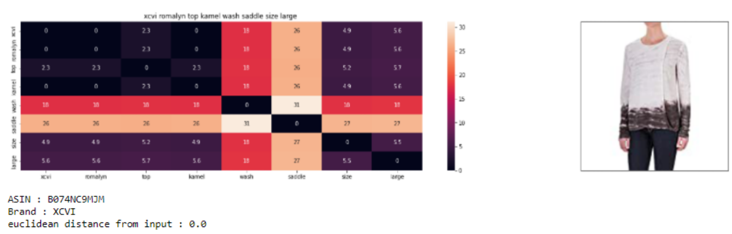

Fig. 3.13 Input for Weighted Similarity using brand and colour

Fig. 3.14 Recommended Products

2. Visual features based Recommendation Engines

Used images to recommend similar products.

Fig. 3.15 Recommended Products using visual features

D. Analysis of results

We used A/B testing to find out which recommendation engine gave the best results. We took 10 test subjects (consisting of friends and family) and we divided them into 2 groups, namely, Group A and Group B. Group A was shown the results from the "Text based recommendation engines" and Group B was shown the results from the "Visual based recommendation engines". Since we can measure the "goodness" through user (test subjects feedback), we took the feedback from 2 groups. (Results shown below):

Note: we are anonymizing test subjects' information for privacy concerns.

- Group A (shown the results from Text based recommendation engines)

Person	**Feedback**
Person 1	"Not so relevant results"
Person 2	"Decent, not so great"
Person 4	"I searched for shirt, recommended tank top"
Person 7	"I liked the recommendations"
Person 8	"Somewhat relevant"

Table 1. Results from Text based Recommendation Engines

- Group B (shown the results from Visual based recommendation engines)

Person	**Feedback**
Person 3	"brilliant"
Person 5	"Relevant to my product"
Person 6	"This is quite relevant and helped me to understand my options"
Person 9	"Its okay"
Person 10	"I like the fact that my recommendations were relevant"

Table 2. Results from Visual based Recommendation Engines

IV. DISCUSSION AND CONCLUSIONS

1) Decisions made / Things that worked

We decided to make multiple recommendation engines so that our outputs (recommendations) are more robust and this helped us a lot.

2) Difficulties faced / Things that didn't work well

We tried stemming and it gave us really bad recommendations, our test subjects did not find the recommendations relevant.

The data for the recommendation was taken from an archived page of amazon since the Amazon API was giving technical issues.

3) Conclusions

The Visual based Recommendation Engines performed better than the Text based Recommendation Engines and it makes sense because in case of apparel the "Visual" part is the core of recommendations as it's the same way a human would recommend an apparel.

In the case of books, we feel Text based Recommendation Engines will perform better because there is no "Visual" part required to recommend books, book's titles, genres would be the core of book recommendation as it's the same way a human would recommend a book.

Our engines are built on simple yet very effective techniques which give very good recommendations.

V. Project Plan / Task Distribution

We divided the tasks in three parts for each of the team members.

1. **Varun**

- Varun was assigned into preprocessing data and understanding the data.
- He was responsible for finding out the statistics of the data.
  - This included finding the missing values
  - Finding the duplicates
  - Also cleaning the data.
- He was also responsible for creating helper functions to display_img, plot_heatmap and plot_heatmap_image which made the text based recommendation engine's code more modular
- He was also responsible for building the Visual based recommendation engine.

2. **Ankit**

- Ankit was assigned the task of implementing text-based recommendations.
- He was responsible for the implementation of
  - Average Word2Vec based product similarity,
  - IDF weighted Word2Vec based product similarity, and
  - Weighted similarity using brand and color.

3. **Parshv**

- Parshv was assigned the task of implementing text-based recommendations.
- He was responsible for the implementation of
  - Bag of Words (BoW) based product similarity,
  - TF-IDF based product similarity, and
  - IDF based product similarity.

Link to Dataset and Code

1. **Dataset**

Dataset Link

2. **Code**

Code Link / GitHub

References

[1] *https://gist.github.com/fchollet/f35fbc80e066a49d65f1688a7e99f069*

[2] *https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html*