Amazon Apparel Recommendation System

Akshay Kumar Gyara, Naga Abhilash Reddy Julakanti, Saikumar Reddy Sandannagari,
Sai Sandeep Jyothula, Sai Sarath Vattikuti
San Jose State University, CA

Abstract:

With the wide availability of the internet and the rise of e-commerce, many people have turned towards online shopping. And the likes of Amazon, Google shopping, Walmart, eBay have made online shopping a breeze. People are preferring to shop on these websites for almost all their needs especially for apparel shopping because of the wide range of options available and comparably cheaper prices. So as part of our project we have built a recommender system for apparel recommendation based on the user searches and also based on the images of the products which will help the user to make better purchases.

Introduction:

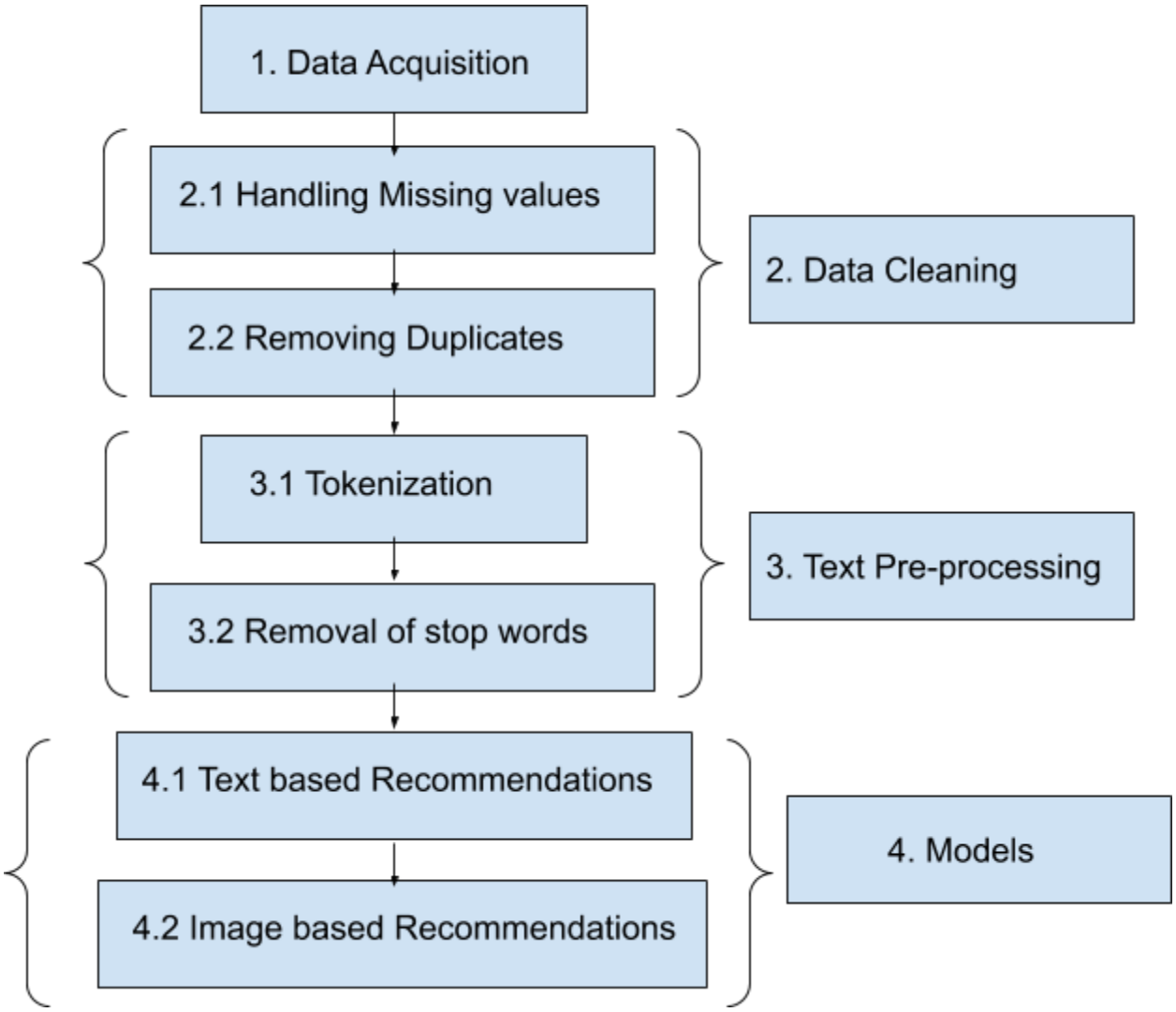
A recommender system, which is considered as a subclass of information filtering systems, are used to predict the preferences of a user would give to an item. Now-a-days recommender systems have become an integral part of almost all the e-commerce websites, entertainment streaming service websites, social media platforms and many other commercial applications. Number of models have been developed to implement recommender systems and the two major types of recommender systems are content based and collaborative systems. We have chosen to build 5 models for text-based product recommendation and merge them as a hybrid recommender system to provide more accurate predictions and also we have implemented Image-based product recommender system. This report will walk you through all the steps we have followed to build these models and also the results.

Motivation:

a. Product recommendations will improve the product discovery as it will automatically list products related to the user preferences so the user need not waste his time looking for unnecessary things
b. Enhances the overall user experience and customer engagement
c. Stats like 35% revenue(which is about $40 Billion) for Amazon is from product recommendations

Factors like these have motivated us to work on making product recommendations more efficient and more accurate.

Workflow:



1. **Data Acquisition:**

Data Acquisition is an important step in any recommendation system. The output and accuracy for the project depends on volume,Variety and Velocity of Data Collection. We used Web Scraping to collect data. We considered ASIN(Amazon Standard Identification Number) as an important feature in web scraping. It is a unique id (Ex:B01HSIIFQ2) for all products on Amazon.

Initially, We collected ASIN values using octoparse tool. By using Asin in following way www.amazon.com/dp/'ASIN', It directs us to product page on Amazon. We build a scraper using python and HTML requests that extracts all the html contents of amazon page. We used XPATHS to extract the particular contents from that page. These

are the following attributes in our data:ASIN, Colour, Brand, Title, Price, Producttype, Medium_Image_URL. We collected data for around 183000 products with following attributes.

```
print ('Number of data points : ', data.shape[0], \
       'Number of features:', data.shape[1])
```

```
Number of data points :  183138 Number of features: 7
```

2. Data Cleaning:

Data cleaning is the process of removing unwanted records from dataset. As we collected raw data from website, There are some noisy records in the data. We followed following techniques to clean the data

I. Removing Null values:

We found that there are some missing values in the collected data for price and colour features. These are the features which cannot be filled with some random value. This forced us to remove those columns with missing data.

```
print('Number of data points After eliminating price=NULL :', data.shape[0])
```

```
Number of data points After eliminating price=NULL : 28395
```

```
print('Number of data points After eliminating color=NULL :', data.shape[0])
```

```
Number of data points After eliminating color=NULL : 28385
```

II. Removing titles with short description:

In text based recommendation, title is an important feature to be considered. We removed the titles of length less than 5 words for efficient recommendations. For example, Let's consider this title, 'tops for Women'. This title gives us very less information about the users interest. So we removed the rows with titles with length less than 5 words.

```
print("After removal of products with short description:", data_sorted.shape[0])
```

```
After removal of products with short description: 27949
```

III. Removing Duplicates with same title:

In this step, we focussed on removing data with similar tile but differ in

size or colour. Recommending the same product with different size or colour is not an efficient way of recommendation. Let's look at an example:

"Tokidoki The Queen of Diamonds Women's Shirt X-Large"

"Tokidoki The Queen of Diamonds Women's Shirt Small"

In this example, Both titles are the same but differ in size, So we removed such titles from our data.

```
print('Number of data points : ', data.shape[0])
```

```
Number of data points :  17593
```

IV. Removing Duplicates with same meaning:

In this step, we tried to remove the products with same titles but differ in meaning. Let's consider an example:

"UltraClub Women's Classic Wrinkle-Free Long Sleeve Oxford Shirt, Pink, XX-Large"

"UltraClub Ladies Classic Wrinkle-Free Long-Sleeve Oxford Pink Shirt XXL"

In this example, Both product titles differ with Women's & Ladies which are of the same meaning. We removed such titles to avoid duplicate recommendations.

```
print('Number of data points after stage two of dedupe: ',data.shape[0])
```

```
Number of data points after stage two of dedupe:  16042
```

3. Data Preprocessing:

Data Preprocessing is the process of transforming raw data into understandable data. This step helps us to transform collected data to data patterns which can be fed to models to predict recommendations. It helps us in extracting interesting information and knowledge from raw data. We used NLTK(Natural Language Tool Kit) library which is most commonly used to analyse, preprocess and understand the text. We used following Data Preprocessing techniques:

I. Tokenization: It is a process of breaking or splitting a sentence into set of words
Let's consider an example

Input: "Hello, How are you doing today?"

Output: {'Hello', ',','How', 'are', 'you', 'doing', 'today', '?' }

The above example shows how tokenization splits sentences into set of words

II. Removal of Stopwords: Stopwords are words with no specific semantics/meaning Examples of stopwords are is, am, are, a, as etc. We used set of stopwords from NLTK library to remove stopwords from the titles.
For example:

Input: {'Hello' ',' 'How', 'are', 'you', 'doing', 'today', '?' }

Output: {‘Hello’, ’How’, ‘are’, ‘you’, ‘doing’, ‘today’}

In the following example, Stopwords such as ‘,’ and ‘?’ are removed from the sentence.

4. Text Based Recommendation Systems:

For text based recommendation, we consider the feature “Title” of each product to recommend apparels. We consider different products p1, p2, p3 and their corresponding titles t1, t2, t3 to find the similarity point using euclidean distance between them. The main concept of Text based recommendation is to convert title into vector of words which will make the similarity measure easy to calculate.

For this system, we are using four different models including the hybrid model that we will discuss later in this paper. Firstly, we will start with ‘Bag of Words’, ‘TF-IDF’, ‘Word2Vec’ and later discuss about hybrid model. We will discuss about each model in depth in the next section.

1. Bag of Words Model:

Bag of words(BOW) is a technique that describes the occurrence of a word within a document. Initially we create a corpus of document which is a collection of all the unique words from all the titles. Then, we arrange each document/title as a vector of corpus of document that represents the value of occurrence of each word in the title. We are using feature extractor from Scikit Learn package to convert each document into a vector.

Of the 16,000 titles that are left, we have extracted around 12,600 unique words to train our model. The vector that we get from this model is sparse and has integers which describes the occurrence of word in the particular title. Once we get vectors for all the titles, we provide input to that model for which we would like to find recommendations and we use cosine similarity between the required title and all other titles. We then print the titles, images, ASIN’s, Brand and Euclidean similarity for top 20 recommendations for the given title.

We also print the heat map for all recommendations which shows the basic difference in words between the titles. Although, this technique groups all the unique words that are available, sometimes it fails to differentiate context between texts and provides wrong recommendations. For example, “Pasta is tasty” and “Pasta is not tasty” have the same words as “Pasta tasty” after stop word removal and this model assumes that both are the same. Hence, we tend to move to other models such as Word2Vec model, Hybrid and Image based Recommendation model, TF-IDF for better recommendations.

2. TF-IDF Model:

Term Frequency(TF) is the number of times a word appears in a given document divided by the total number of words in the document. Inverse Document Frequency(IDF) is a measure of how much information the word provides. TF focuses more on the common words where as IDF concentrates on rare words. In BOW model, all words are given the same weightage and prediction will go wrong sometimes. Hence, TF-IDF which is product of TF and IDF provides the correct combination and gives similar products.

$$\mathbf{TF = f_{t,d} \div (number\ of\ words\ in\ d)}$$

$$\mathbf{IDF = \log\left(\frac{total\ number\ of\ documents}{number\ of\ documents\ with\ term\ t\ in\ it} \right)}$$

Similar to bag of words model, we create a vector of unique words and then we create vector for each and every title and fill it with TF*IDF values. Then we apply cosine similarity to required title with all other titles. After finding the euclidean similarities, we will print the top 20 recommendations for the given title. Also, this model doesn't resolve the problem that we faced for previous model. Hence, we look for Word2Vec, Hybrid and Image based Recommendation models.

3. Word to vector Model:

Unlike TF-IDF and Bag of Words models word to vector is a semantic similarity based model which is used to convert each word of the document into a vector of dimension d. This model takes a large corpus of textual data as its input and then it produces a vector space.Where every unique word in the corpus is assigned to corresponding vector in the space.These vectors are positioned in the vector space such that words which share the common meaning in the contextual text will be located in close proximity to one another in the space.

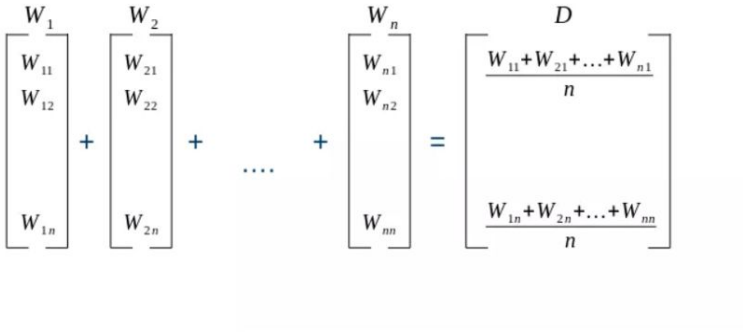
In our model a word to vector model is built using the library gensim where each word in the document D_i in the corpus C_i are converted to a 300 dimension vector.The data used in the word to vector is similar to the frequency based models(16000 documents).The vector formed by using word to vector model is dense unlike sparse vectors.

In our project we have implemented word to vector model as average word to vector model and TF-IDF word to vector model. By using word to vector model we obtained distinct results and discovered new patterns when compared with the frequency based models.

3.1 Average Word to vector model:

In a Document D_i each vector of the corresponding word is considered to form an average vector which is represented as the vector to subsequent

document.By Using average word to vector model we can leverage the vector to a better value where a different output cannot be obtained with the outliers in the vector(highest and lowest values of vector).



average word vectors

3.2 TF-IDF word to vector model:

In a Document D_i each vector of the corresponding words are multiplied with the TF-IDF value and are leveraged by dividing the sum of all the word vectors to the sum of the TF-IDF Values of each vector.This model obtained better results and discovered many new patterns during the recommendation because both semantic Meaning and word importances are considered as TF-IDF value of each word is used as the weight in the model.

$$Tf - Idf\ w2vec = \frac{1}{N}\sum_{D=1}^{n}\sum_{W_{i=1}}^{D_i} tf - idf_{D_i} * [wi]$$

Where N= $\sum Tf - Idf_{D_i}$

4. Hybrid Recommendation Model:

This is an important model which is also being used in real world business . As every model has some pros and cons ie BOW ,TF-IDF models only focuses on frequency of words but not on words meaning whereas Word2Vec models emphasis on semantic based similarity. It would be wise decision to combine all the models to get the best possible results.

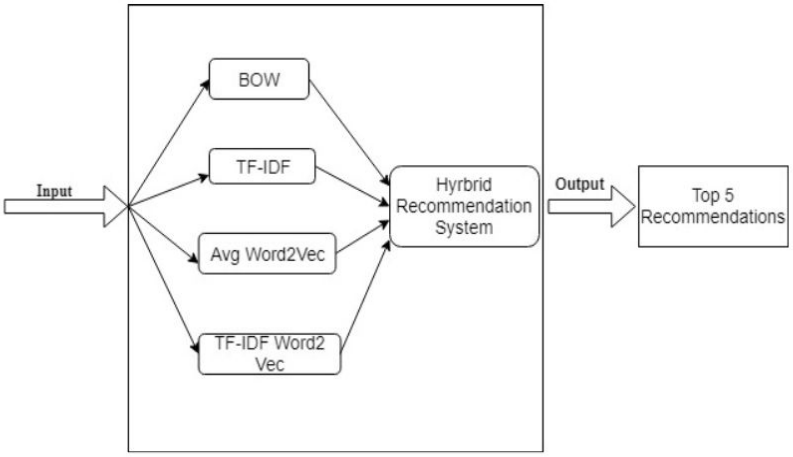


Fig. Hybrid Recommendation System

The advantage of this Hybrid model is it combines the output of all the above models and recommends the top most similar items. We have constructed parallel hybrid recommendation model with 4 recommendation system. The outputs of all the four models will be sent to hybrid recommendation system which outputs top n similar products.

5. Image Based Recommendation System:

In this recommendation system, the user inputs an image to the model and it will recommend the top similar items that match with the input image. To implement this model, we first have to convert the given image to a vector format so that it can be applied to a recommendation system. For converting image to vector we have used VGG 16 CNN model. It is the latest model proposed by Oxford University which works well with deep convolutional Network for large scale image recognition.

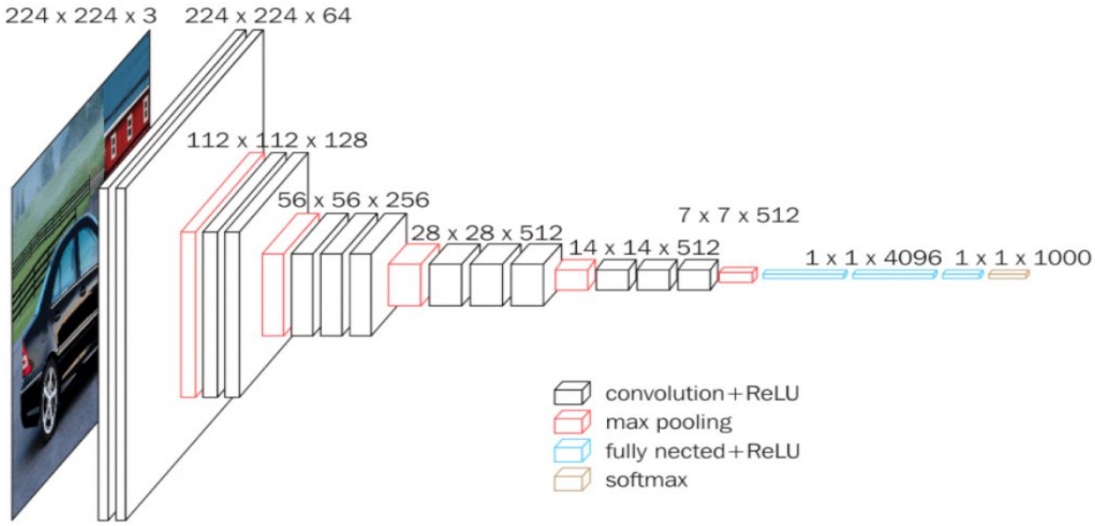


Fig. VGG16 architecture

We have chosen this model as it has 16 layers deep network which is efficient to extract even the precise features from an image, as a result, the network learns rich representation for a wide range of images. This model accepts an input size of 224 x 224 RGB image.

We have collected nearly 16,000 images which will be converted to vector using VGG16. We ran the model with 50 epochs and batch size =1 so that all the important features of the image can be extracted. Since the image dataset is huge it took nearly 3 hrs to complete the process.

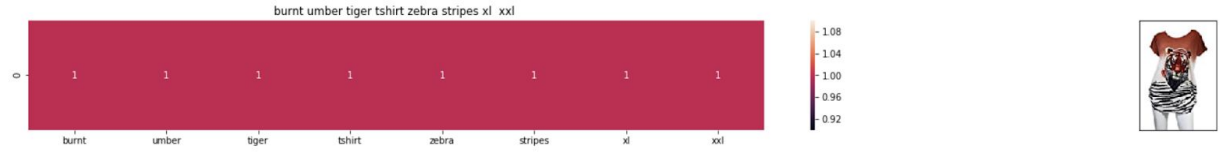
The output of this VGG16 model is a dense vector of length 300 which has all the information of images such as colors, patterns, edges, and shapes.
Now we have 16,000 vectors for which we will calculate the euclidean distance between every image. Whichever image has small euclidean distance with the user input will be recommended to the user.

Results:

Now we have looked at all different types of models and how we are using them to get recommendations for the product. Now let us look at the results of each model given the same input. This will help us to evaluate all the models and choose the best one for our recommendation system.

Input:

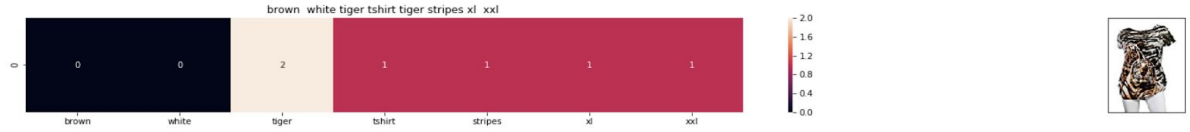
This is the product we will be taking as input for all the models and we will try to find similarities between titles using euclidean distance with different models.




```
ASIN : B00JXQB5FQ
Brand: Si Row
Title: burnt umber tiger tshirt zebra stripes xl  xxl
Euclidean similarity with the query image : 0.0
```

A. Output for Bag Of Words:


```
ASIN : B00JXQASS6
Brand: Si Row
Title: pink tiger tshirt zebra stripes xl  xxl
Euclidean similarity with the query image : 1.73205080757
============================================================
```

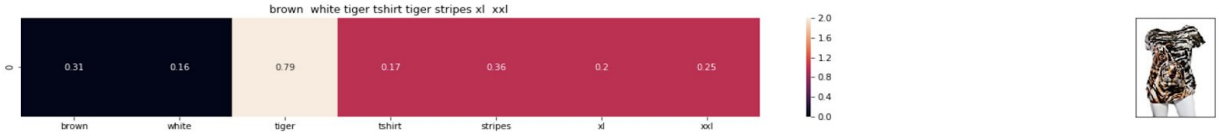



```
ASIN : B00JXQCWTO
Brand: Si Row
Title: brown  white tiger tshirt tiger stripes xl  xxl
Euclidean similarity with the query image : 2.44948974278
============================================================
```

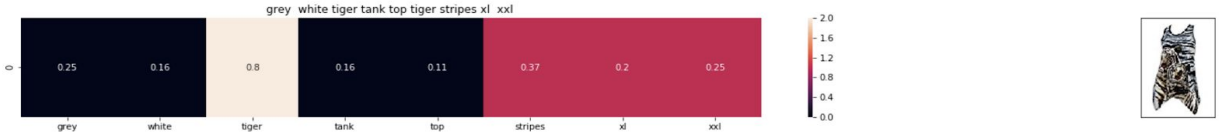

```
ASIN : B00JXQCUIC
Brand: Si Row
Title: yellow tiger tshirt tiger stripes  l
Euclidean similarity with the query image : 2.64575131106
```

B.Output for TF-IDF:


```
ASIN : B00JXQASS6
BRAND : Si Row
Eucliden distance from the given image : 0.753633191245
============================================================================================================================
========
```

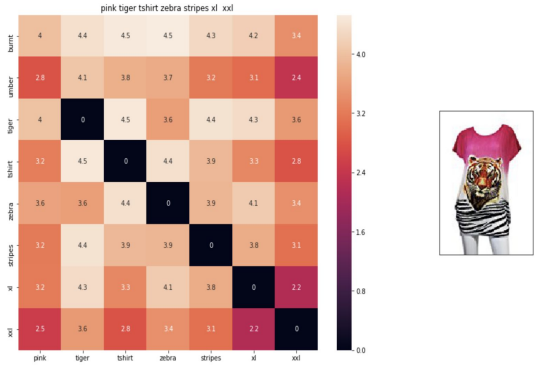



```
ASIN : B00JXQCWTO
BRAND : Si Row
Eucliden distance from the given image : 0.935764394377
============================================================================================================================
========
```

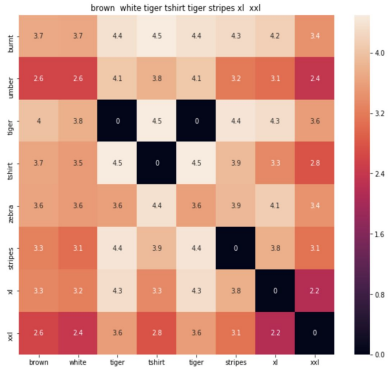



```
ASIN : B00JXQAFZ2
BRAND : Si Row
Eucliden distance from the given image : 0.95861535242
```

C. Output for Avg Word2 Vec:

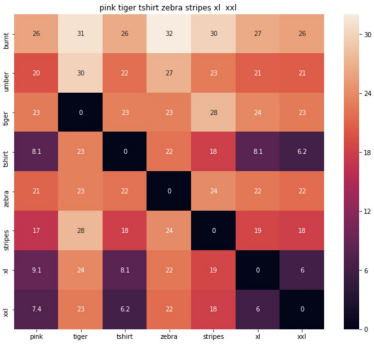


ASIN : B00JXQASS6
BRAND : Si Row
euclidean distance from given input image : 0.589193

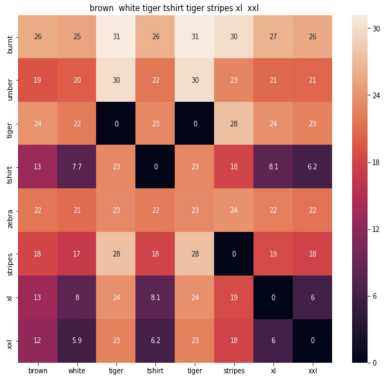


ASIN : B00JXQCWTO
BRAND : Si Row
euclidean distance from given input image : 0.700344

D. Output for Tf-IDF Word2Vec:

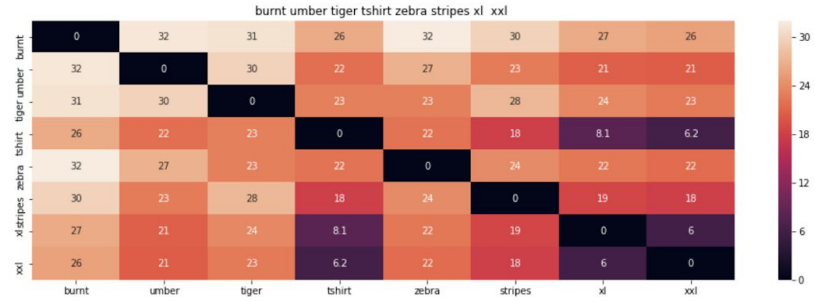


ASIN : B00JXQASS6
Brand : Si Row
euclidean distance from input : 4.06389

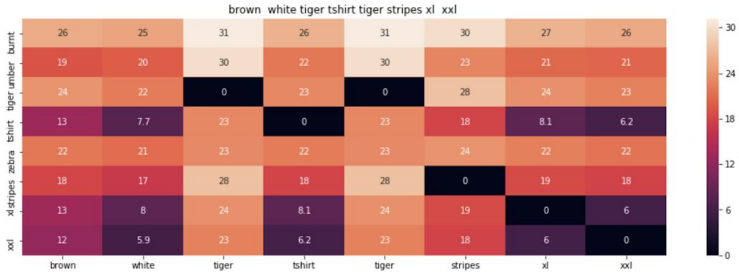


ASIN : B00JXQCWTO
Brand : Si Row
euclidean distance from input : 4.77094

E. Output for Hybrid Model:



ASIN : B00JXQB5FQ
Brand : Si Row
euclidean distance from input : 0.000355113636364



ASIN : B00JXQCWTO
Brand : Si Row
euclidean distance from input : 0.433722027865

F. Image Based Recommendation system :

Input:

Product Title: fashion crop tops women casual summer emoji sexy lady girl shirt hipster tank top
Euclidean Distance from input image: 3.8146973e-06
Amazon Url: www.amzon.com/dp/B010V3B44G

Output:

Product Title: women quotes boys print white sleeveless crop top
Euclidean Distance from input image: 22.04546
Amazon Url: www.amzon.com/dp/B0748CKWF3

Product Title: women three wise monkeys emoji print sleeveless crop top
Euclidean Distance from input image: 24.452633
Amazon Url: www.amzon.com/dp/B074VPC98H

Conclusion:

We have built two kinds of recommender systems using text-based model and image based model to recommend the apparel to the users.

Text-based models:

- There are 4 text based models which are built as part of our project which are Bag of words, Tf-Idf, Avg word2vec, Tf-idf-Word2vec. Using these models a parallel Hybrid recommendation system is built, where the most occuring recommendations of the 4 models are considered as the final recommendations to the user.

Image-based model-

- Vg-16 CNN model is used in our project to build a recommendation system for the users.Each image is converted to a dense vector of 25,088 dimensions size .

In all models Euclidean distance is computed between the vectors of each product.The products having less euclidean distance are considered to be most similar.

References:

- http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
- https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
- https://www.kaggle.com/c/word2vec-nlp-tutorial#part-2-word-vectors
- https://docs.aws.amazon.com/AWSECommerceService/latest/DG/Welcome.html