CONTENT BASED RECOMMENDATION SYSTEM

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

by

Arpan Dixit
(15419MCA013)

Department of Computer Science
Institute of Science

Banaras Hindu University, Varanasi – 221005

May 2018

CANDIDATE'S DECLARATION

I hereby certify that the work, which is being presented in the report/thesis, entitled Content Based Recommendation System, in partial fulfillment of the requirement for the award of the Degree of Master of Computer Applications and submitted to the institution is an authentic record of my own work carried out during the period *Jan-2018* to *May-2018* under the supervision of Ms. Ankita Vaish. I also cited the reference about the text(s) /figure(s) /table(s) /equation(s) from where they have been taken.

The matter presented in this thesis as not been submitted elsewhere for the award of any other degree or diploma from any Institutions.

Date: Signature of the Student

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: Signature of the Research Supervisor

The Viva-Voce examination of *Arpan Dixit*, M.C.A. Student has been held on __________________.

Signature of External Examiner

Signature of Head of the Department

ABSTRACT

The main objective of this project is to implement a Content Based Recommendation System, similar to those find on e-commerce websites such as Flipkart, Amazon, etc. making use of machine learning techniques. Similar to the recommendation systems found on these popular sites, this project deals with the working and implementation of Content Based Recommendation. The similarity evaluated is based on text based similarity, text based semantical similarity and visual similarity of the images of these products. The models mainly used are tf-idf, word2vec (common bag of words) and visual similarity is evaluated using Convolutional Neural Networks

Keywords: recommendation system, content based, filtering,

TABLE OF CONTENTS

Title……………………………………………………………………………Page

ABSTRACT………………………………………………………………………ii

LIST OF TABLES………………………………………………………………v

LIST OF FIGURES……………………………………………………………...vi

CHAPTER 1 INTRODUCTION

1.1 General……….………………………………………………………...1

1.2 Objectives…..………………………….……………………………..1

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction…………………………………………………….……2

2.1.1 Content-Based……………………………………………………….2

2.1.2 Collaborative Filtering…………….………………………………2

2.2 State of the art of Content Based Recommendation Systems…………….….3

2.2.1 Keyword-based Vector Space Model……………………………..4

2.2.2 Semantic Analysis based on Ontologies………………………….6

2.2.3 Semantic Analysis by using Encyclopedic Knowledge Sources…………7

2.3 Methods to learn User Profiles……………………………………………8

2.3.1 Probabilistic Methods……………………………………………..8

2.3.2 Relevance Feedback………………………………………………9

2.3.3 Other Methods……………………………………………………9

CHAPTER 3 PROPOSED APPROACH AND IMPLEMENTATION

3.1 Preparing the Dataset……………………………………. ………..11

3.2 Data Cleaning…………………………………………………………12

3.2.1 Missing Data in Features…………………………………………12

3.2.2 Removing duplicate rows………………………………………...14

3.3 Text Preprocessing……………………………………………………………..15
3.3.1 Tokenization and Stop Word Removal……………………………………16
3.3.2 Stemming…………………………………………………………………….17
3.4 Text Based Product Similarity…………………………………………………….17
3.4.1 Bag of Words Model (BoW)………………………………………………17
3.4.2 TF-IDF Based Product Similarity…………………………………………19
3.5 Text Semantics Based Product Similarity…………………………………………20
3.5.1 Word2Vec Model…………………………………………………………..20
3.5.2 TF-IDF Weighted Word2Vec Model……………………………………....21
3.6 Deep Learning Based Visual Product Similarity…………………………………22
3.6.1 Neural Networks…………………………………………………………..23
3.6.2 Convolutional Neural Networks…………………………………………...23
3.6.3 Visual Based Product Similarity…………………………………………...23

CHAPTER 4 DISCUSSION

4.1 Introduction……………………………………………………………………….26
4.2 Scalability…………………………………………………………………………26
4.3 Implementation Details……………………………………………………………26

CHAPTER 5 CONCLUSION AND FUTURE WORKS

5.1 Introduction……………………………………………………………………….28
5.2 Conclusion………………………………………………………………………...28
5.3 Future Works……………………………………………………………………...29
5.3.1 Role of user-generated content in Recommendation Process……………...29
5.3.1.1 Social Tagging Recommender Systems…………………………30

APPENDIX…………………………………………………………………………..31

LIST OF TABLES

3.1 Details of the data sets corresponding to the feature product_type_name…………12

3.2 Details of the data sets corresponding to the feature brand………………………..13

LIST OF FIGURES

3.1 Details of the dataset…………………………………………………………...11

3.2 Head of the Dataset, showing values of features of interest……………………...12

3.3 Shape of the Data Set after Handling Missing Values……………………………13

3.4 Images of example of duplicate products, that differ only in size…………………14

3.5 Example titles of duplicate products that have different sizes……………………14

3.6 Images of Example of duplicate products that differ…………………………....15

3.7 List of Stop Words…………………………………………………………………16

3.8 Results of Text Based Similarity of Products using BoW…………………………18

3.9 Results of Text Based Similarity of Products using BoW(cont.)…………………...18

3.10 Results of Text Based Similarity of Products using TFIDF……………………….19

3.11 Results of Text Based Similarity of Products using TFIDF(cont)…………………20

3.12 Text Semantics Based Similarity using word2vec…………………………………21

3.13 Text Semantics Based Similarity using Brand and Color………………………….23

3.14 Text Semantics Based Similarity using Brand and Color (cont.)…………………..23

3.15 Artificial Neural Network: Representation and Activation Function………………23

3.16 Visual Based Similarity Results (Target Product)………………………………….24

3.17 Visual Based Similarity Results (Recommended Product)………………………..24

3.18 Visual Based Similarity Results (Recommended Product)………………………..25

CHAPTER 1

INTRODUCTION

1.1GENERAL

The current e-commerce focuses a lot of attention on the recommendation or suggestions offered to the users. The recommendation system is broadly divided into two categories: Content Based Recommendation and Collaborative Filtering. This project delves into Content Based Filtering. The filtering is performed using certain similarity measures. Popular similarity measures include Euclidean Distance, Manhattan Distance, Minkowski Distance etc. Euclidean Distance is evaluated as:

$$\text{dis(p,q)} = \sum_{i=0}^{n} \sqrt{(p_i - q_i)^2}$$

The recommendation system gives as a list of similar products based on the Euclidean distance between the target product and itself.

1.2 OBJECTIVES

The objective of this project is to filter the relevant results from the available inventory and output products that are to a certain extent similar to the product in question. The product is compared to the products available on the site and recommends them to the user browsing. The products recommended are evaluated for their similarity using the textual similarity of the titles of these products or the visual similarity of images of these products.

CHAPTER 2
LITERATURE REVIEW

2.1 INTRODUCTION

A common situation for recommendation systems is an internet application with which a user interacts. Typically, a system presents a outline list of things to a user, and the user selects among the things to receive a lot of details on associate item or to act with the item in some means. E-commerce sites usually gift a page with a listing of individual merchandise to enable the user to check the details of a couple of selected product and buy the product. Though the net server transmits hypertext markup language and also the user sees an internet page, the internet server generally has a info of things and dynamically constructs internet pages with a list of things. as a result of there are usually several a lot of things offered in a database than would simply work on an internet page, it's necessary to pick a set of things to show to the user or to see associate order during which to show the things. Content-based recommendation systems analyze item descriptions to spot things that are of specific interest to the user[1].

2.1.1.Content-based

The system learns to recommend those products similar to those previously liked by the user. The similarity of items is evaluated using the features of the compared items. For example, if a user has positively rated a movie that belongs to the action genre, then the system will try to recommend other movies from this genre. The state-of-the-art systems that have been adopted in several application domains[4].

2.1.2 Collaborative filtering

The simplest implementation of this approach [2] recommends to the active user those items that were preferred by other users of similar tastes. The similarity in taste of two users is calculated based on the similarity in the rating history of the users. That is why [3] refers to collaborative filtering as “people-to-people correlation.” Collaborative filtering is one of the most popular and widely used technique in recommender systems.

2.2 STATE OF THE ART OF RECOMMENDATION SYSTEM

Content-based filtering uses the content of items to predict its importance based on the user's taste. Research on content-based recommender systems intersects with a variety of computer science topics, especially Information Retrieval [5] and Artificial Intelligence.

From Information Retrieval (IR), research on recommendation systems facilitates the vision that users searching for recommendations are busy with an information seeking process. Items to be recommended vary a lot, largely because of the number and types of attributes used to describe them. Each item in a data set can be represented using the same attributes with their specific set of values, but this may not be very insightful for data, such as Web pages, news, emails or documents, described through unstructured text. In this case there are no features with straight forward values, and document modeling techniques are used in IR research.

From an Artificial Intelligence point of view, the recommendation problem is quite similar to a learning problem that depends heavily on the past history of the users. At their simplest, user profiles are arranged in a form of user-specified rules, and reflects to a certain extent the long-term interests of that person. More often than not, it is expected from the recommendation system to learn the finer details of the user profile rather than imposing upon the user to help with one. This delves into the applications of Machine Learning (ML) techniques, whose goal is learning to organize new items based on previously seen information that the user has labeled as interesting or not. Given these labeled items, ML methods are used to generate a model that, given any new information, will help to decide whether it can be of any interest for the target user.
Items that are recommended to the users are represented by using a set of features or attributes. For example, in a garments recommendation system, features adopted to describe a product, in general are: brand, color, gender targeted, image of the product, occasion, print, quality and type of cloth, etc. When each and every item is represented using the same set of attributes, and there is a small number of known values these attributes may take, the item is represented by means of structured data[6]. In this case, many ML algorithms can be used to learn a user profile [7]. In most content-based filtering systems, item details are text features that extracted from websites, emails, news

articles or product descriptions. As an effect, there are no features with well-defined values. Textual features pose a series of problems and complications when learning a user profile, due to the ambiguity of natural languages. The problem is that generally keyword-based profiles are a bit short, when it comes to capturing the semantics of user interests because they mostly use a string matching operation. If a string, or some morphological variant, is found in both the profile and the document, a match is made and the document is considered as relevant. String matching suffers from problems of:

•POLYSEMY: the presence of multiple meanings for one word;

•SYNONYMY: multiple words with the same meaning.

The result is that synonymy leads to neglecting relevant information if the profile does not contain the same exact keywords while, due to polysemy, irrelevant documents could be deemed relevant. Semantic analysis and its integration in these models is one of the most innovative and interesting approaches proposed to solve those problems. The major idea behind it is the adoption of knowledge bases, such as lexicons or ontologies, for annotating items and representing these profiles, thus obtaining a "semantic" equivalent of the user information.

2.2.1 KEYWORD BASED VECTOR BASED MODEL

Most content-based recommendation systems use simple retrieval models, such as keyword matching or the Vector Space Model with basic weighted TF-IDF. Vector Space Model is a spatial representation of text. In this model, documents are represented by a vector in a n-dimensional space, where each dimension corresponds to a term from the overall vocabulary of a given corpus.

Formally, every document is represented as a vector of term weights, where each weight represents the associability of the document and the term. Let D = {d1, d2,...,dN} denote a set of corpus, and T={t1,t2,...,tn} be the set of words in the corpus. T is obtained by applying standard natural language processing operations, such as tokenization, stop-words removal, and stemming [5]. Each document or corpus dj is represented as a vector in a n-dimensional vector space, so dj={w1j,w2j,...,wnj}, where wk j is the weight for term tk in the corpus dj.

Corpus representation in the Vector Space Model raises two issues: weighting the terms and measuring the feature vector similarity. The most commonly used term weighting scheme, TF-IDF(Term Frequency-Inverse Document Frequency) weighting, is based on generic observations regarding text [8]:

• rare terms are more relevant than frequent terms (IDF assumption);

• multiple occurrences of a term in a document are not less relevant than single occurrences (TF assumption);

• long documents are not preferred to short documents (normalization assumption). In other words, terms that occur frequently in one document (TF=term-frequency), but rarely in the rest of the corpus (IDF= inverse-document-frequency), are more likely to be relevant to the topic of the document. These assumptions are well exemplified by the TF-IDF function:

$$TF\text{-}IDF(tk,dj)=TF(tk,dj).log(N/nk)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(2.1)$$

Where N denotes the number of documents in the corpus, and nk denotes the number of documents in the collection in which the term tk occurs at least once.

$$TF(tk,dj)=fk,j/maxzfz,j\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(2.2)$$

where the maximum is computed over the frequencies fz,j of all terms tz that occur in the corpus dj. In order for the weights to fall in the [0,1] interval and the documents to be represented by vectors of equal length, weights obtained by Equation (2.1) are usually normalized by cosine normalization:

$$wk,j=TF\text{-}IDF(tk,dj)\surd\sum TF\text{-}IDF(ts,dj)2\ldots\ldots(2.3)$$

which enforces the normalization assumption.

As stated earlier, a similarity measure is required to evaluate the degree of similarity between two documents. Many similarity measures have been derived to describe the proximity of two vectors. In content-based recommender systems relying on Vector

Space Model, both user profiles and items are represented as weighted term vectors. Predictions of a user's interest in a particular item can be dderived by evaluating the Euclidean distance.

2.2.2 SEMANTIC ANALYSIS BASED ON ONTOLOGIES

Semantic analysis enables learning user profiles more accurately, that references to the ideas present in knowledge bases. The main support for this approach comes from the fact that it is somewhat of a challenge of providing a recommendation system with the cultural and linguistic background knowledge which mainly emphasizes the ability of interpreting natural language documents and reasoning on their content.

The description of these strategies is carried out by taking into account several criteria:

•the type of knowledge source involved (e.g. lexicon, ontology, etc.);

•the techniques adopted for the annotation or representation of the items;

•the type of content included in the user profile;

•the item-profile matching strategy.

SiteIF[9] is a personal agent for a multilingual news Web site. It was the very first system to successfully adopt and execute a sense-based document representation in order to build a model of the user interests. The external knowledge base is MultiWordNet, a multilingual lexical database where English and Italian senses are arranged. The profile is built as a semantic networhose nodes represent synsets found in the documents read by the user.

ITR (Item Recommender) is a system deployed in order to provide recommendations for items in several domains (e.g., movies, music, books), if the details of items are available as text documents (e.g. plot summaries, reviews, short abstracts) [10, 11]. Similarly to SiteIF, ITR integrates linguistic knowledge in the process of learning user profiles. The linguistic knowledge comes exclusively from the WordNet lexical ontology. Items are represented according to a synset-based vector space model, called bag-of-synsets (BOS), that is a variation of the classical bag-of-words (BOW) [13, 12]. In the BOS model, a synset vector, rather than a word vector, corresponds to a document.

Quickstep[13] is a system for the recommendation of on-line academic research papers. This system utilizes the ontology based on the classifications made by the DMOZ open directory project (27 classes used) on computer science research papers. Semantic annotation of papers comprises in associating them with class names within the ontology, by using a k-Nearest Neighbor classifier. Profiles are computed by correlating previously browsed research papers with their classification. User profiles thus hold a set of topics and interest values in these topics. Foxtrot [13] extends the Quickstep system by implementing a paper search interface, a visualization interface and an email notification, in addition to the recommendation interface. Profile visualization is made possible because of their representation in ontological terms understandable to the users.

2.2.3 SEMANTIC ANALYSIS BY USING ENCYCLOPEDIC KNOWLEDGE SOURCES

Common-sense and domain-specific knowledge may prove to be very useful in improving the strength of natural language processing techniques by generating more informative features than the bag of words. The process of learning user profiles could see a measurable benefit from the addition of knowledge which is being externally provided, with respect to the use of knowledge which is extracted from the corpus. Examples of general purpose knowledge bases include the Open Directory Project (ODP), Wikipedia etc. Explicit Semantic Analysis (ESA) [15, 16] is a technique used to provide a fine grained semantic representation of natural language texts in a high-dimensional space of natural concepts from Wikipedia. Concepts are defined by Wikipedia articles, e.g., INDIA, COMPUTER SCIENCE, or TAJ MAHAL. This approach is inspired by the keen desire of being able to augment text representation with colossal amount of knowledge. In the case of Wikipedia as knowledge source, there are several advantages, such as its constant addition of details by the highly motivated and the learned community, multilingual articles, and the high accuracy [17]. Empirical evaluations showed that ESA brought about impressive improvements in computing word, and in the text categorization task. It has also been shown that ESA supplemented the traditional BOW-based retrieval models [30]. Another approach of adding semantics to text is the Wikify! system [18, 19], which has been seen to be able to identify important concepts in a text , and then link these concepts to the corresponding Wikipedia pages. The annotations produced by the Wikify! system can be used to automatically enrich documents with references to semantically related information.

2.3 METHODS TO LEARN USER PROFILES

Machine learning techniques, generally used for text categorization are similar to those used in the task of inducing content-based profiles [20]. For a machine learning technique to be useful in text categorization, a text classifier is built automatically by a inductive process [6]. The classifier is trained from a set of training documents (labeled documents).

The problem of learning user profiles can be visualized as a binary task: each document has to be classified as interesting or not with respect to the user's taste. Therefore, the set of categories is C = {+ve, -ve}, where +ve is the positive class (user-likes) and -ve the negative one (user-dislikes)

2.3.1 PROBABILISTIC METHODS

Naïve Bayes is a probabilistic approach to inductive learning, and belongs to the general class of Bayesian classifiers. These approaches generate a probabilistic model based on previously observed data. The model estimates the a posteriori probability, P(c|d), of document d belonging to class c. This estimation is based on the a priori probability, P(c), the probability of observing a document in class c, P(d|c), the probability of observing the document d given c, and P(d), the probability of observing the instance d. Using these probabilities, the Bayes theorem is applied to calculate P(c|d):

$$P(c|d) = \frac{P(c).P(d|c)}{P(d)}$$

To classify the document d, the class with the highest probability is chosen

The Naïve Bayes classifier makes an independence assumption to simplify the model: all the words or tokens in the observed document d are conditionally independent of each other given the class. The conditional independence assumption is quite easily violated in real-world data, however, despite these violations; the Naïve Bayes classifier does a good job in classifying text documents [21, 22].

2.3.2 Relevance Feedback

Relevance feedback is a technique used in Information Retrieval that helps users to refine search queries incrementally according to previous search results. It consists of the users feeding back into the system the relevance of retrieved documents with respect to the information need.

Significance criticism and its adjustment to content classification, the notable Rocchio's recipe [24], are usually utilized by content-based recommender frameworks. The general rule is to enable clients to rate reports proposed by the recommender framework as for their data require. This type of criticism can along these lines be utilized to incrementally refine the client profile or to prepare the learning calculation that surmises the client profile as a classifier.

Some direct classifiers comprise of an express profile (or prototypical report) of the classification [26]. The Rocchio's strategy is utilized for prompting direct, profile-style classifiers. This calculation speaks to reports as vectors so archives with comparable substance have comparable vectors. Every segment of such a vector compares to a term in the record, commonly a word. The heaviness of every segment is registered utilizing the TF-IDF term weighting plan.

2.3.4 OTHER METHODS

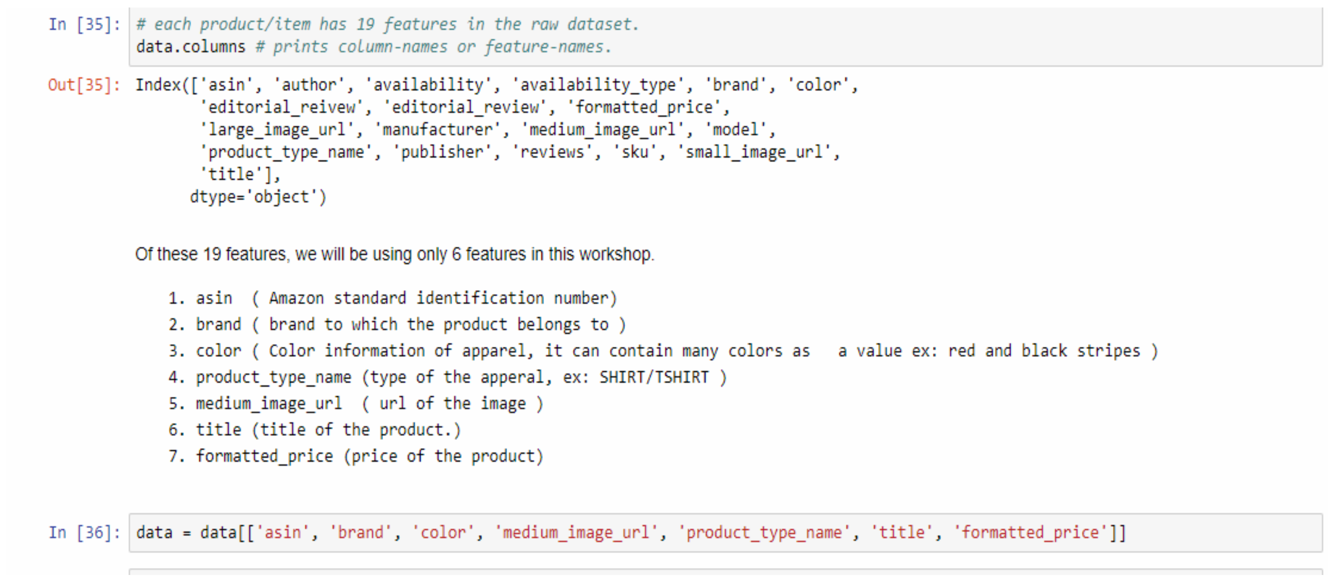
Other learning algorithms have been utilized as a part of substance based suggestion frameworks. An extremely short depiction of the most critical calculations takes after. A careful survey is displayed in [27, 28, 26].

Decision trees will be trees in which inside hubs are marked by terms, branches withdrawing from them are marked by tests on the weight that the term has in the test archive, and leaves are marked by classifications. Decision trees are found out by recursively partitioning training data, that is content reports, into subgroups, until those subgroups contain just occasions of a solitary class. The test for parceling information is keep running on the weights that the terms naming the inward hubs have in the record. The decision of the term on which to work the segment is by and large made by a data pick up or entropy model [101]. Choice trees are utilized as a part of the Syskill and Webert [70, 68] recommender framework.

CHAPTER 3
PROPOSED APPROACH AND IMPLEMENTATION

3.1 PREPARING THE DATASET

The dataset was downloaded from the Amazon API, in the form of a JSON file. The data contains roughly 183138 unique points. There are 19 features associated with each of these data points.



```
In [35]: # each product/item has 19 features in the raw dataset.
         data.columns # prints column-names or feature-names.

Out[35]: Index(['asin', 'author', 'availability', 'availability_type', 'brand', 'color',
                'editorial_reivew', 'editorial_review', 'formatted_price',
                'large_image_url', 'manufacturer', 'medium_image_url', 'model',
                'product_type_name', 'publisher', 'reviews', 'sku', 'small_image_url',
                'title'],
               dtype='object')
```

Of these 19 features, we will be using only 6 features in this workshop.

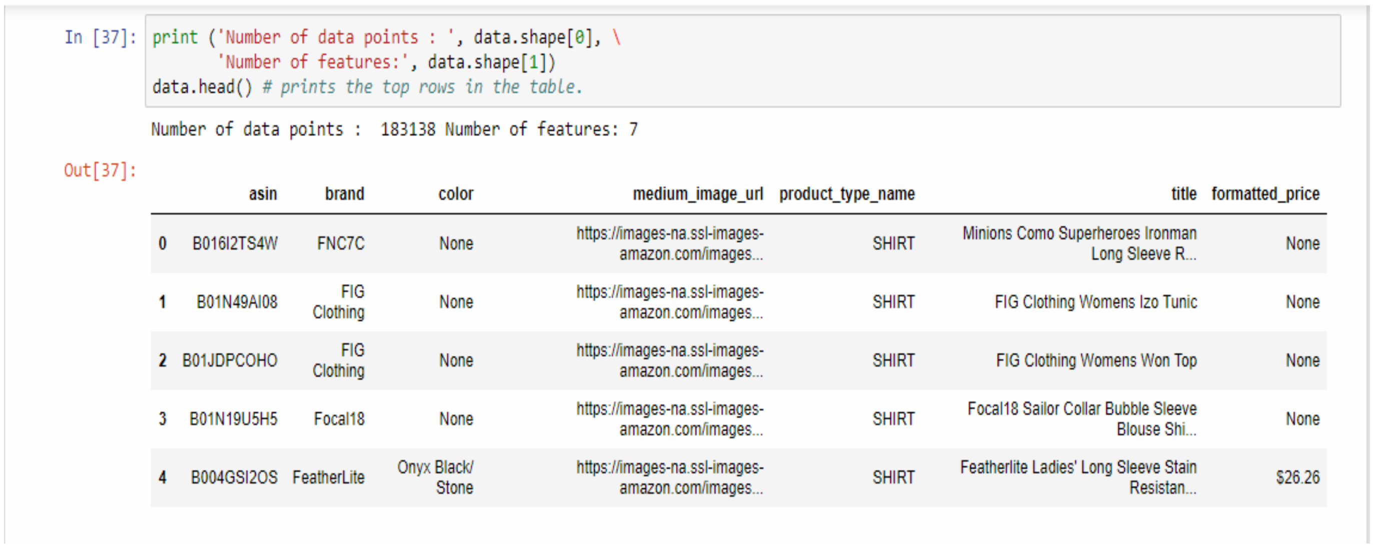
```
1. asin  ( Amazon standard identification number)
2. brand ( brand to which the product belongs to )
3. color ( Color information of apparel, it can contain many colors as   a value ex: red and black stripes )
4. product_type_name (type of the apperal, ex: SHIRT/TSHIRT )
5. medium_image_url  ( url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

```
In [36]: data = data[['asin', 'brand', 'color', 'medium_image_url', 'product_type_name', 'title', 'formatted_price']]
```

Fig 3.1 Details of the dataset

The features of interest in this project are 6, viz. asin, brand, color, product_type_name, medium_image_url, title, and formatted_price.

After selecting the required features from all the features, the data set reduces to a dataset of 183k points with 6 features representing each one of them. The details of a few products is in fig 3.2.



```
In [37]: print ('Number of data points : ', data.shape[0], \
               'Number of features:', data.shape[1])
         data.head() # prints the top rows in the table.

         Number of data points :  183138 Number of features: 7
Out[37]:
```

	asin	brand	color	medium_image_url	product_type_name	title	formatted_price
0	B016I2TS4W	FNC7C	None	https://images-na.ssl-images-amazon.com/images...	SHIRT	Minions Como Superheroes Ironman Long Sleeve R...	None
1	B01N49AI08	FIG Clothing	None	https://images-na.ssl-images-amazon.com/images...	SHIRT	FIG Clothing Womens Izo Tunic	None
2	B01JDPCOHO	FIG Clothing	None	https://images-na.ssl-images-amazon.com/images...	SHIRT	FIG Clothing Womens Won Top	None
3	B01N19U5H5	Focal18	None	https://images-na.ssl-images-amazon.com/images...	SHIRT	Focal18 Sailor Collar Bubble Sleeve Blouse Shi...	None
4	B004GSI2OS	FeatherLite	Onyx Black/ Stone	https://images-na.ssl-images-amazon.com/images...	SHIRT	Featherlite Ladies' Long Sleeve Stain Resistan...	$26.26

Fig 3.2 Head of the Dataset, showing values of features of interest

3.2 DATA CLEANING

The data that is selected is full of inconsistencies such as NULL values, and needs to be cleaned, for efficient and error free training of the model. The steps involved are:

3.2.1 MISSING DATA IN FEATURES:

In the data set, there are certain data points, for which the values of one or more features are missing. These missing values pose to be a hindrance in model training.

Upon inspection of the feature of "product_type_name", we can see:

Table 3.1 Details of the data sets corresponding to the feature product_type_name

Count	**183138**
Unique	**72**
Top	**SHIRT**
Freq	**167794**
Name, dtype	**product_type_name, object**

Thus, this feature has no missing value.

For the feature, “brand”, the details are:

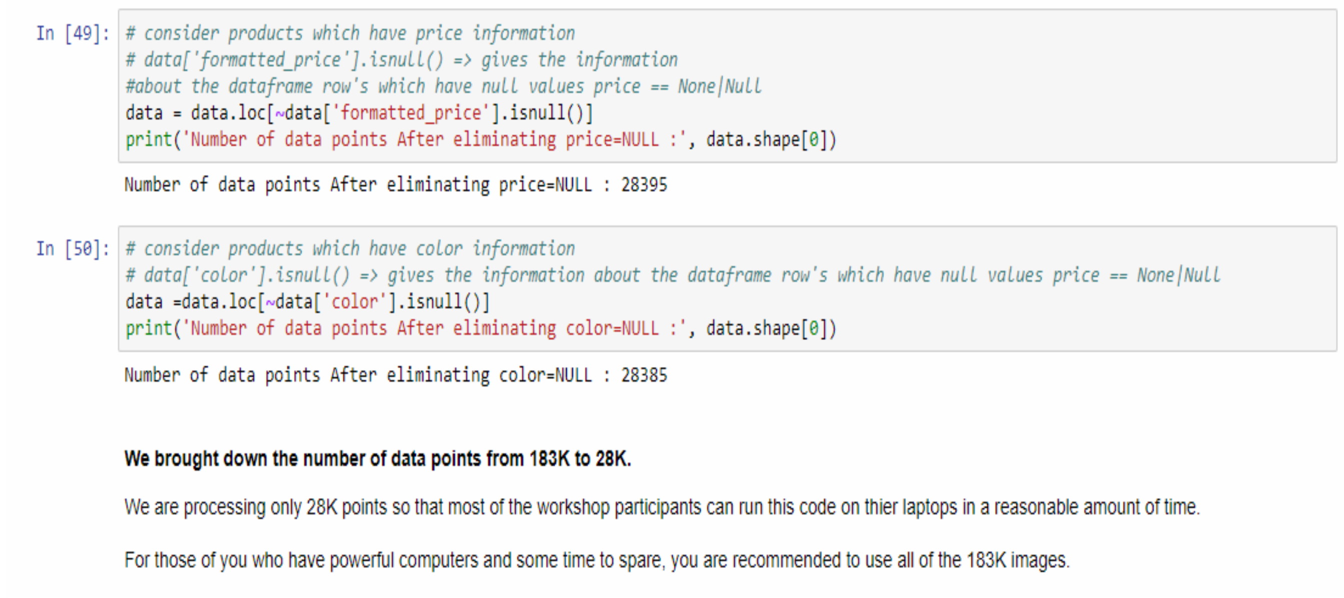
Table 3.2 Details of the data sets corresponding to the feature brand

Count	**182987**
Unique	**10577**
Top	**Zago**
Freq	**223**
Name, dtype	**brand, object**

Thus, there are 151 missing values.

Similarly, for feature color the count is 64956, for formatted_price the count is 28395, and so on

The data rows that have null values are removed, from the data set. This leads to unique data rows being reduced to 28k



```
In [49]: # consider products which have price information
         # data['formatted_price'].isnull() => gives the information
         #about the dataframe row's which have null values price == None|Null
         data = data.loc[~data['formatted_price'].isnull()]
         print('Number of data points After eliminating price=NULL :', data.shape[0])

         Number of data points After eliminating price=NULL : 28395

In [50]: # consider products which have color information
         # data['color'].isnull() => gives the information about the dataframe row's which have null values price == None|Null
         data =data.loc[~data['color'].isnull()]
         print('Number of data points After eliminating color=NULL :', data.shape[0])

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

We brought down the number of data points from 183K to 28K.

We are processing only 28K points so that most of the workshop participants can run this code on thier laptops in a reasonable amount of time.

For those of you who have powerful computers and some time to spare, you are recommended to use all of the 183K images.

Fig 3.3 Shape of the Data Set after Handling Missing Values

3.2.2 REMOVING DUPLICATE VALUES

There are certain products in the data rows, which can be considered as duplicates. If all the details except the size of two or more products are same, they are duplicates (fig 4). The titles of certain products are identical except for a few last words, and hence be removed (fig 5).

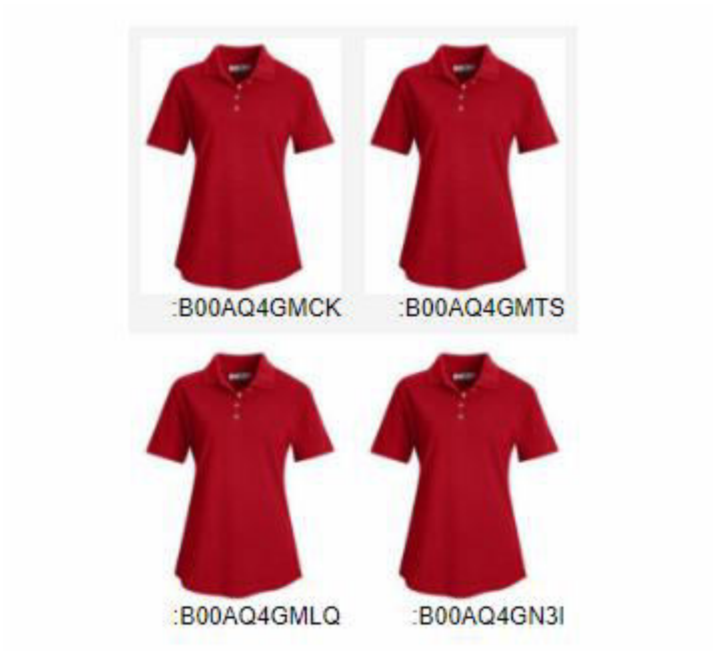
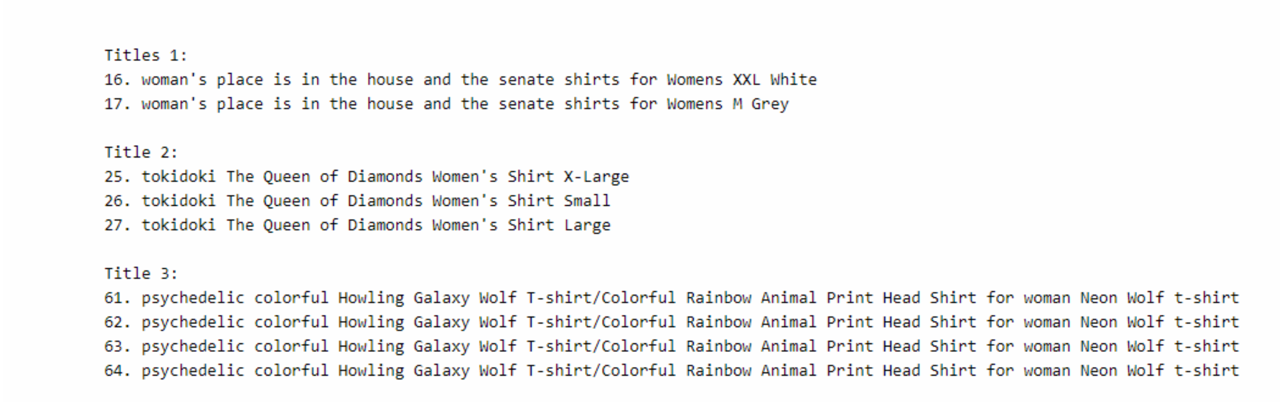


Fig 3.4 Images of example of duplicate products, that differ only in size



```
Titles 1:
16. woman's place is in the house and the senate shirts for Womens XXL White
17. woman's place is in the house and the senate shirts for Womens M Grey

Title 2:
25. tokidoki The Queen of Diamonds Women's Shirt X-Large
26. tokidoki The Queen of Diamonds Women's Shirt Small
27. tokidoki The Queen of Diamonds Women's Shirt Large

Title 3:
61. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt
62. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt
63. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt
64. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Head Shirt for woman Neon Wolf t-shirt
```

Fig 3.5 Example titles of duplicate products that have different sizes

Also, for products that are identical except for their color are also considered to be duplicates (fig 3.6)

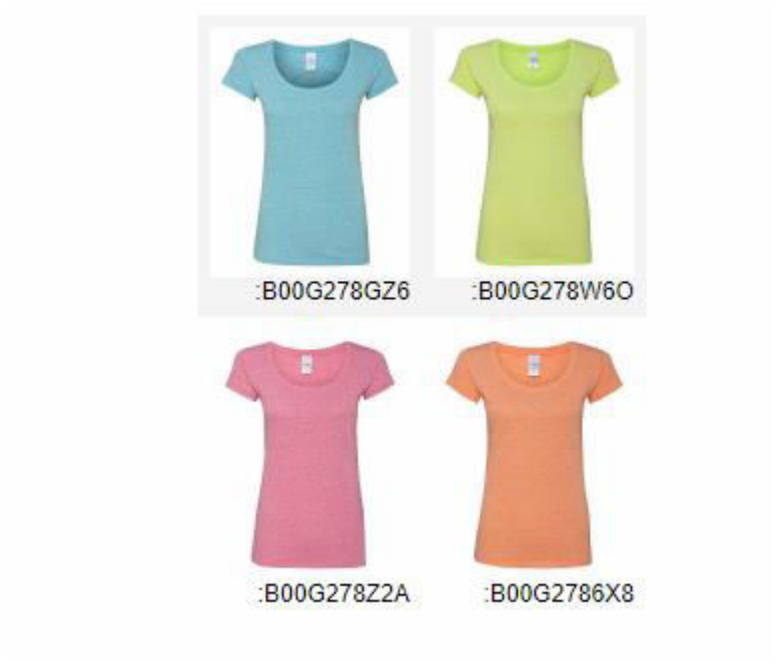


Fig 3.6 Images of Example of duplicate products that differ

The duplicates should be removed from the data sets. After removal of these duplicate items, we are left with 16k products. The resultant data set is free from duplicate products, and the items with empty values or NULL values are also removed from the dataset for easier processing.

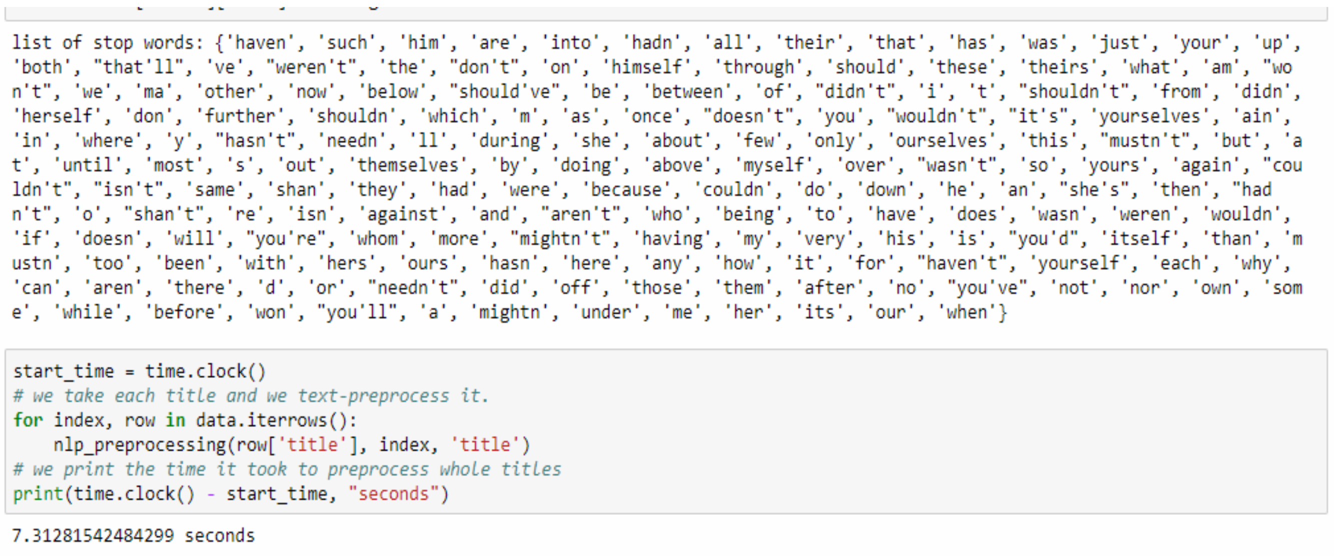
3.3 TEXT PREPROCESSING

The titles of the products are in the form of natural language. Thus, for efficient training of model, we need to adopt traditional natural language processing techniques, such as tokenization, stop word removal, and stemming.

3.3.1 Tokenization and Stop Word Removal:

In Natural Language processing a token is a small piece of text. There are diverse tokenization procedures, yet you need to remember that the procedure comprises in cutting an amount of words into littler sacks of word. The stop words

Stop words are the words we need to sift through before preparing the classifier. These are generally high recurrence words that aren't giving any extra data to our naming. Truth be told, they really confound our classifier. In English, they could be the, is, at, which, and on - the list is in fig 3.7



```
list of stop words: {'haven', 'such', 'him', 'are', 'into', 'hadn', 'all', 'their', 'that', 'has', 'was', 'just', 'your', 'up',
'both', "that'll", 've', "weren't", 'the', "don't", 'on', 'himself', 'through', 'should', 'these', 'theirs', 'what', 'am', "wo
n't", 'we', 'ma', 'other', 'now', 'below', "should've", 'be', 'between', 'of', "didn't", 'i', 't', "shouldn't", 'from', 'didn',
'herself', 'don', 'further', 'shouldn', 'which', 'm', 'as', 'once', "doesn't", 'you', "wouldn't", "it's", 'yourselves', 'ain',
'in', 'where', 'y', "hasn't", 'needn', 'll', 'during', 'she', 'about', 'few', 'only', 'ourselves', 'this', "mustn't", 'but', 'a
t', 'until', 'most', 's', 'out', 'themselves', 'by', 'doing', 'above', 'myself', 'over', "wasn't", 'so', 'yours', 'again', "cou
ldn't", "isn't", 'same', 'shan', 'they', 'had', 'were', 'because', 'couldn', 'do', 'down', 'he', 'an', "she's", 'then', "had
n't", 'o', "shan't", 're', 'isn', 'against', 'and', "aren't", 'who', 'being', 'to', 'have', 'does', 'wasn', 'weren', 'wouldn',
'if', 'doesn', 'will', "you're", 'whom', 'more', "mightn't", 'having', 'my', 'very', 'his', 'is', "you'd", 'itself', 'than', 'm
ustn', 'too', 'been', 'with', 'hers', 'ours', 'hasn', 'here', 'any', 'how', 'it', 'for', "haven't", 'yourself', 'each', 'why',
'can', 'aren', 'there', 'd', 'or', "needn't", 'did', 'off', 'those', 'them', 'after', 'no', "you've", 'not', 'nor', 'own', 'som
e', 'while', 'before', 'won', "you'll", 'a', 'mightn', 'under', 'me', 'her', 'its', 'our', 'when'}
```

```
start_time = time.clock()
# we take each title and we text-preprocess it.
for index, row in data.iterrows():
    nlp_preprocessing(row['title'], index, 'title')
# we print the time it took to preprocess whole titles
print(time.clock() - start_time, "seconds")
```

```
7.31281542484299 seconds
```

Fig 3.7 List of Stop Words

For our unfiltered content, in the principal line, we load an array of stop words utilizing stopwords library. We at that point split the first content in words — additionally getting out images and digits — and sift through the words in the stopwords cluster, and after that change over everything back to a string.

3.3.2 STEMMING

“Stemming is the process for reducing inflected words to their word stem (base form).” Words like *banks* and *banking* progress toward becoming bank, and contributing and contributed move toward becoming contribute. The classifier doesn't comprehend that the verbs contributing and contributed are the same, and regards them as various words with various frequencies. By stemming them, it bunches the frequencies of various affectations to only one term — for this situation, contribute. Stemming did not prove useful for this project and hence was not used.

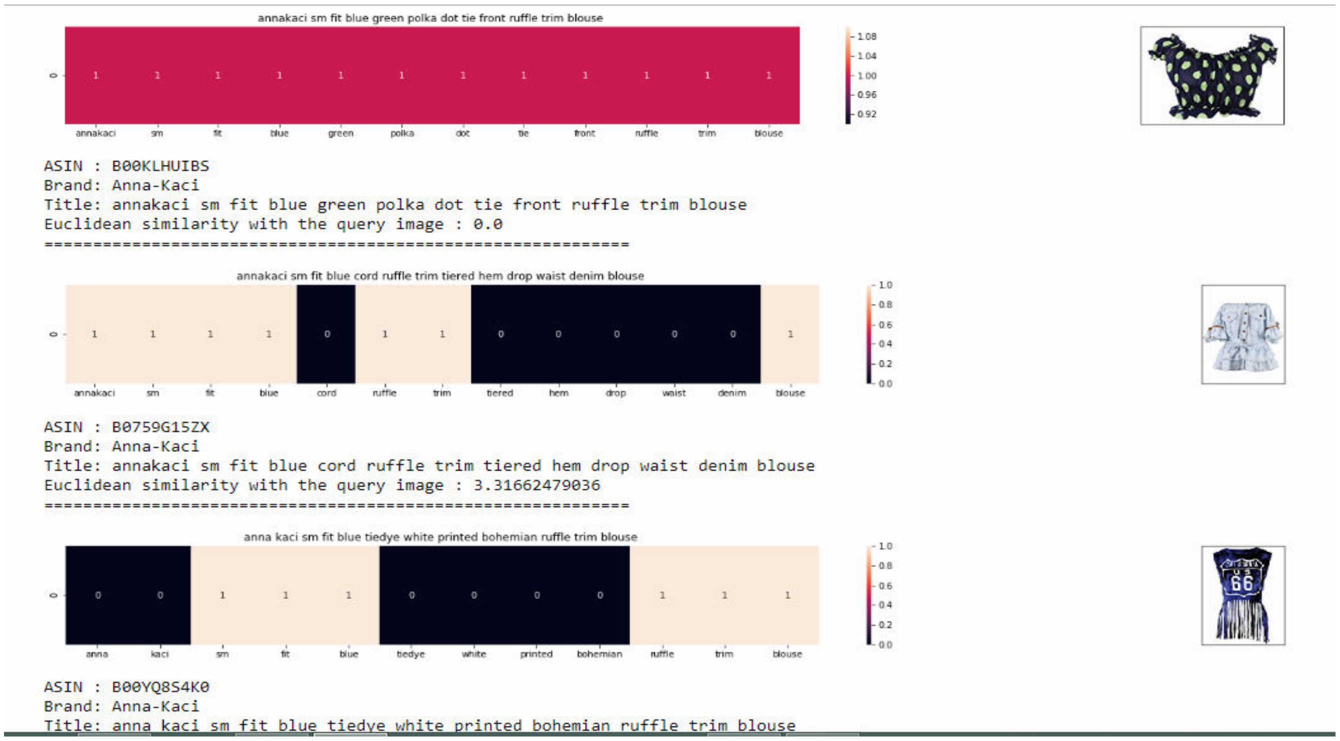
3.4 TEXT BASED PRODUCT SIMILARITY

The text based similarity is evaluated using the product titles. In this case the method adopted focus mainly on the text based similarity, without any focus on semantic structure of the sentence.

3.4.1 BAG OF WORDS MODEL (BoW)

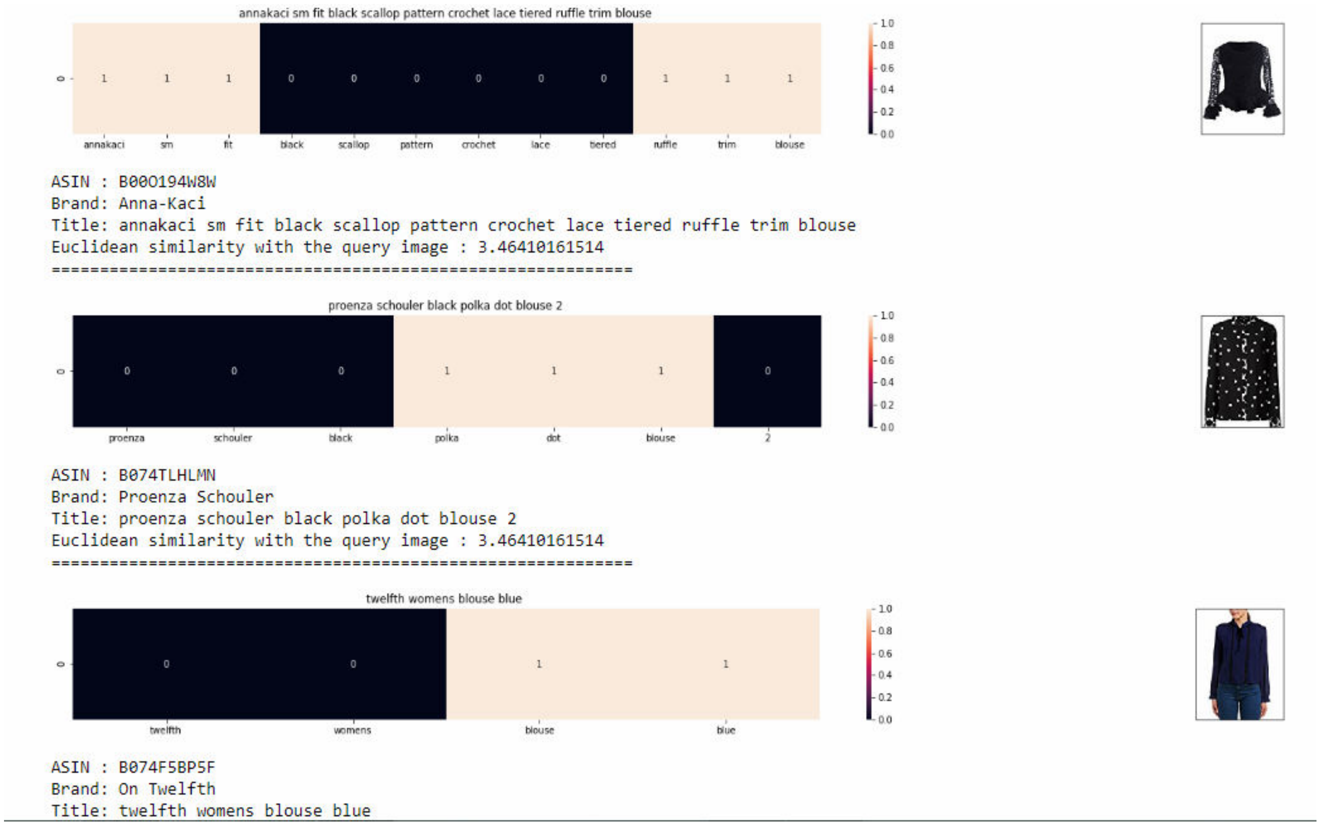
The bag-of-words model is a simplifying portrayal utilized as a part of natural language processing and data recovery (IR). It is also known as the vector space model. In this model, a text (for example, a sentence or a record) is spoken to as the pack (multiset) of its words, dismissing language structure and even word arrangement however keeping multiplicity.
A bag-of-words model, or BoW for short, is a method for removing highlights from content for use in modeling, for example, with machine learning algorithms. The approach is exceptionally basic and adaptable, and can be utilized as a part of a bunch of routes for removing highlights from records. A bag of-words is a portrayal of content that depicts the event of words inside a record. It includes two things: A vocabulary of known words, and a measure of the nearness of known words.

It is known as a "bag" of words, on the grounds that any data about the request or structure of words in the report is disposed of. The model is just worried about whether known words happen in the record, not where in the report.


annakaci sm fit blue green polka dot tie front ruffle trim blouse
annakaci sm fit blue cord ruffle trim tiered hem drop waist denim blouse
anna kaci sm fit blue tiedye white printed bohemian ruffle trim blouse

```
ASIN : B00KLHUIBS
Brand: Anna-Kaci
Title: annakaci sm fit blue green polka dot tie front ruffle trim blouse
Euclidean similarity with the query image : 0.0
============================================================
ASIN : B0759G15ZX
Brand: Anna-Kaci
Title: annakaci sm fit blue cord ruffle trim tiered hem drop waist denim blouse
Euclidean similarity with the query image : 3.31662479036
============================================================
ASIN : B00YQ8S4K0
Brand: Anna-Kaci
Title: anna kaci sm fit blue tiedye white printed bohemian ruffle trim blouse
```

Fig 3.8 Results of Text Based Similarity of Products using BoW

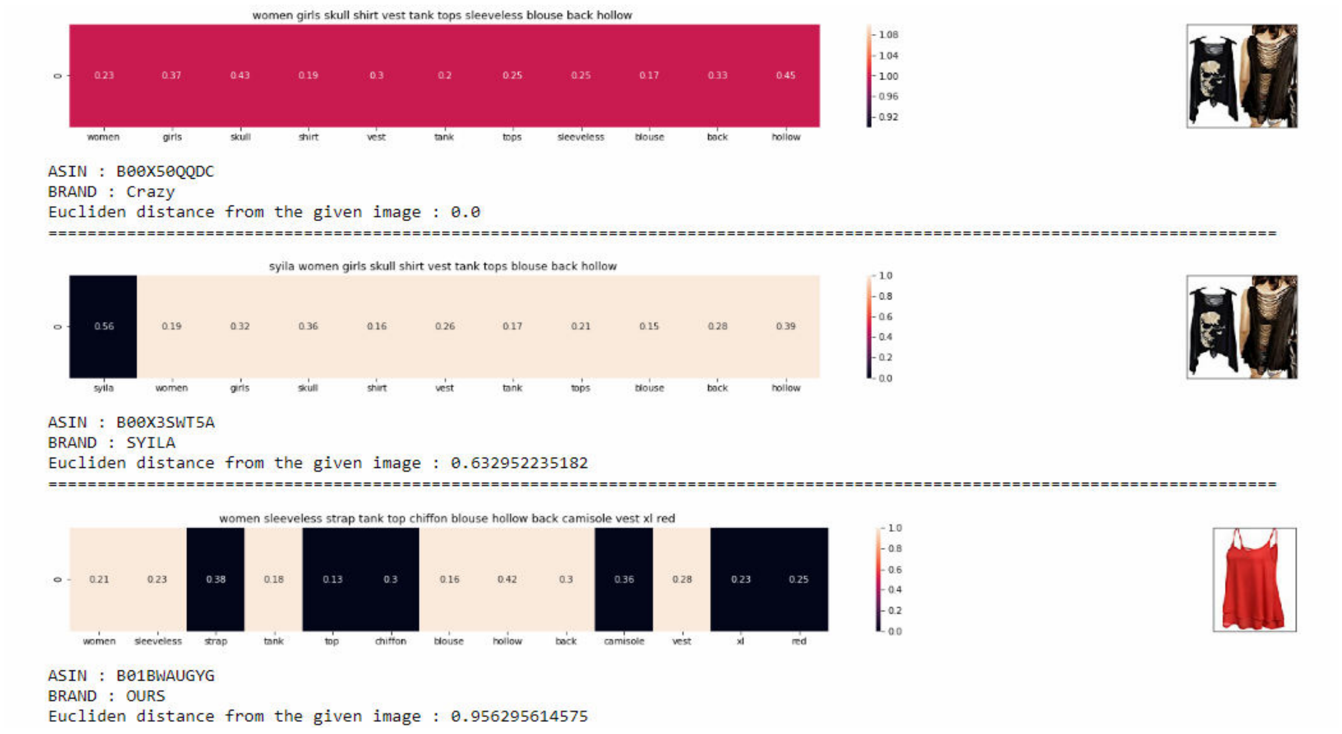

annakaci sm fit black scallop pattern crochet lace tiered ruffle trim blouse
proenza schouler black polka dot blouse 2
twelfth womens blouse blue

```
ASIN : B00O194W8W
Brand: Anna-Kaci
Title: annakaci sm fit black scallop pattern crochet lace tiered ruffle trim blouse
Euclidean similarity with the query image : 3.46410161514
============================================================
ASIN : B074TLHLMN
Brand: Proenza Schouler
Title: proenza schouler black polka dot blouse 2
Euclidean similarity with the query image : 3.46410161514
============================================================
ASIN : B074F5BP5F
Brand: On Twelfth
Title: twelfth womens blouse blue
```

Fig 3.9 Results of Text Based Similarity of Products using BoW(cont)

3.4.2 TF-IDF Based Product Similarity

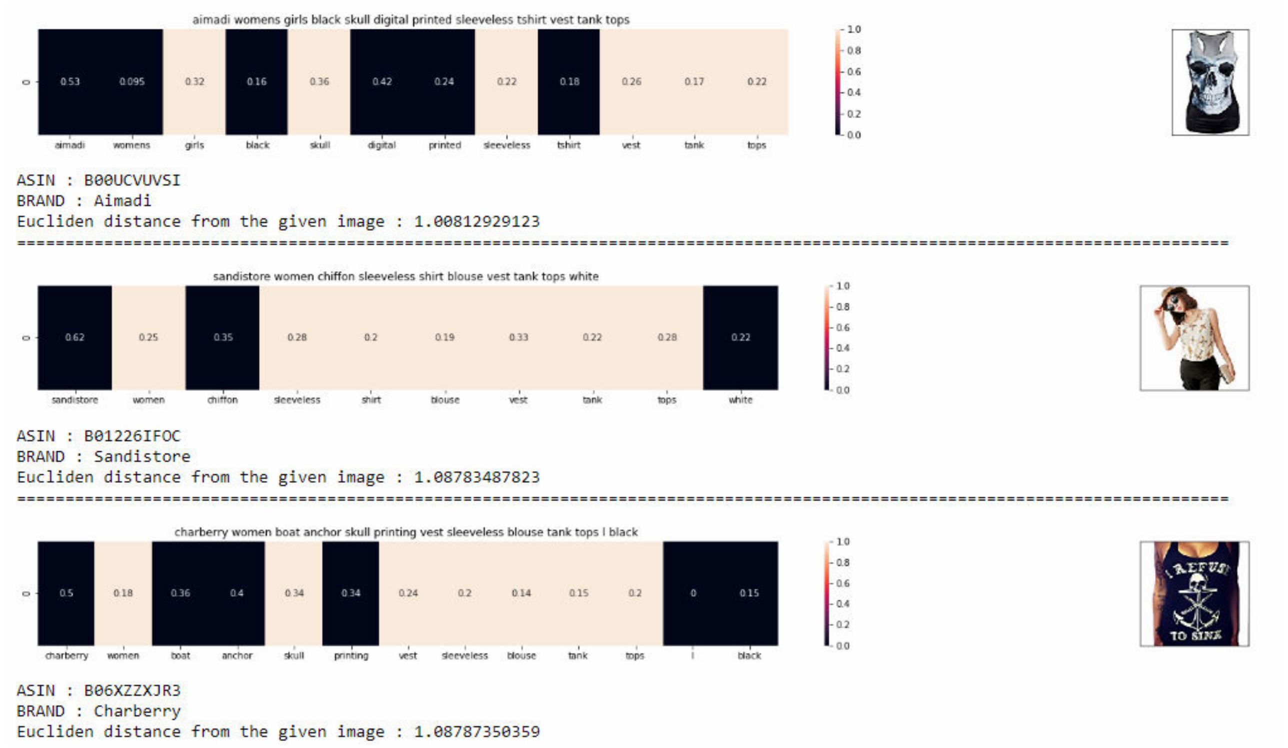
TF-IDF, short for term frequency– inverse document frequency, is a numerical measurement that is planned to reflect how critical a word is to a document in a corpus. It is frequently utilized as a weighting factor in ventures of information retrieval, text mining, and user modeling. The tf-idf value increases relatively to the number of times a word shows up in the record and is counterbalanced by the recurrence of the word in the corpus.


women girls skull shirt vest tank tops sleeveless blouse back hollow
0.23 0.37 0.43 0.19 0.3 0.2 0.25 0.25 0.17 0.33 0.45
women girls skull shirt vest tank tops sleeveless blouse back hollow
1.08 1.04 1.00 0.96 0.92
syila women girls skull shirt vest tank tops blouse back hollow
0.56 0.19 0.32 0.36 0.16 0.26 0.17 0.21 0.15 0.28 0.39
syila women girls skull shirt vest tank tops blouse back hollow
1.0 0.8 0.6 0.4 0.2 0.0
women sleeveless strap tank top chiffon blouse hollow back camisole vest xl red
0.21 0.23 0.38 0.18 0.13 0.3 0.16 0.42 0.3 0.36 0.28 0.23 0.25
women sleeveless strap tank top chiffon blouse hollow back camisole vest xl red
1.0 0.8 0.6 0.4 0.2 0.0

```
ASIN : B00X50QQDC
BRAND : Crazy
Eucliden distance from the given image : 0.0
============================================================================================================================
ASIN : B00X3SWT5A
BRAND : SYILA
Eucliden distance from the given image : 0.632952235182
============================================================================================================================
ASIN : B01BWAUGYG
BRAND : OURS
Eucliden distance from the given image : 0.956295614575
```

Fig. 3.10 Results of Text Based Similarity of Products using TFIDF

Regularly, the tf-idf weight is created by two terms: the first registers the standardized Term Frequency (TF), otherwise known as. the time a word shows up in a record, isolated by the aggregate number of words in that archive; the second term is the Inverse Document Frequency (IDF), processed as the logarithm of the quantity of the reports in the corpus partitioned by the quantity of archives where the particular term shows up.


aimadi womens girls black skull digital printed sleeveless tshirt vest tank tops
0.53 0.095 0.32 0.16 0.36 0.42 0.24 0.22 0.18 0.26 0.17 0.22
aimadi womens girls black skull digital printed sleeveless tshirt vest tank tops
sandistore women chiffon sleeveless shirt blouse vest tank tops white
0.62 0.25 0.35 0.28 0.2 0.19 0.33 0.22 0.28 0.22
sandistore women chiffon sleeveless shirt blouse vest tank tops white
charberry women boat anchor skull printing vest sleeveless blouse tank tops l black
0.5 0.18 0.36 0.4 0.34 0.34 0.24 0.2 0.14 0.15 0.2 0 0.15
charberry women boat anchor skull printing vest sleeveless blouse tank tops l black
1.0 0.8 0.6 0.4 0.2 0.0

```
ASIN : B00UCVUVSI
BRAND : Aimadi
Eucliden distance from the given image : 1.00812929123
==============================================================================================================================
ASIN : B01226IFOC
BRAND : Sandistore
Eucliden distance from the given image : 1.08783487823
==============================================================================================================================
ASIN : B06XZZXJR3
BRAND : Charberry
Eucliden distance from the given image : 1.08787350359
```

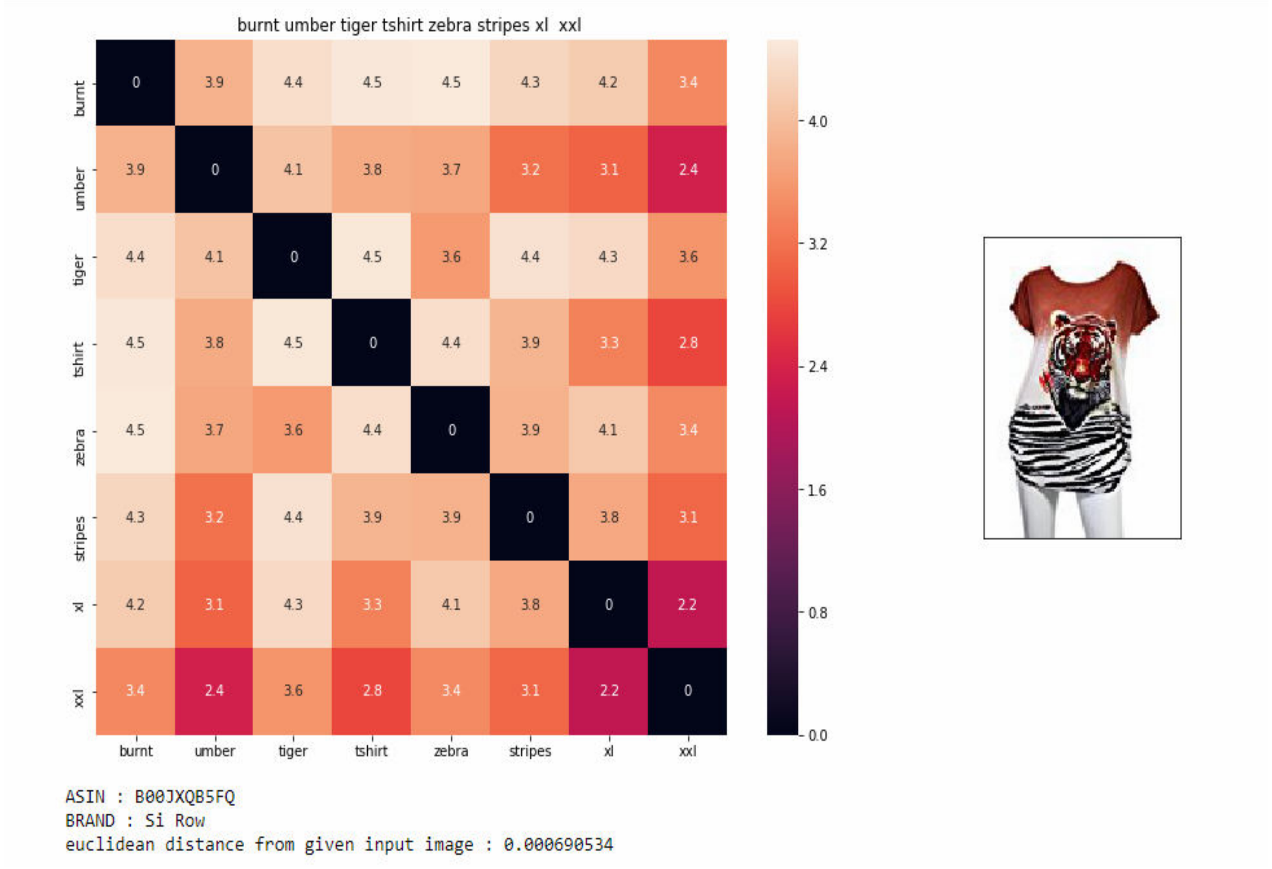
Fig 3.11 Results of Text Based Similarity of Products using TFIDF(cont)

3.5 TEXT SEMANTICS BASED PRODUCT SIMILARITY

In this approach, the semantics behind the product's title is also taken into consideration. A very popular approach is Word2Vec model.

3.5.1 Word2Vec Model

Word2vec is a gathering of related models that are utilized to deliver word embeddings. These models are shallow, two-layer neural systems that are prepared to remake etymological settings of words. Word2vec takes as its info a substantial corpus of content and delivers a vector space, commonly of a few hundred measurements, with every interesting word in the corpus being allocated a relating vector in the space. Word vectors are situated in the vector space with the end goal that words that offer normal settings in the corpus are situated in nearness to each other in the space.



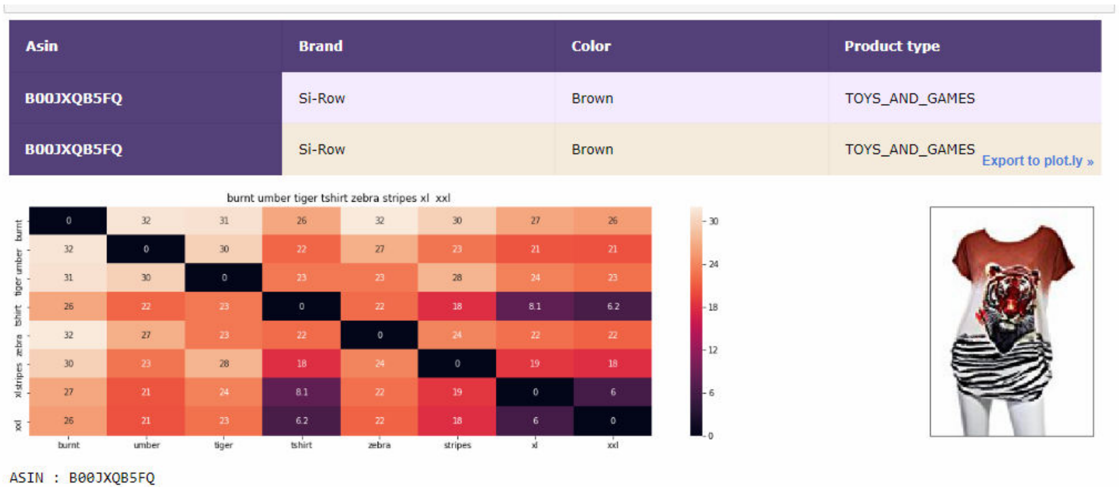
```
ASIN : B00JXQB5FQ
BRAND : Si Row
euclidean distance from given input image : 0.000690534
```

Fig. 3.12 Text Semantics Based Similarity using word2vec

3.5.2 WEIGHTED SIMILARITY USING BRANDS AND COLORS

The color and brands of the product are transformed using one-hot encoding. These vectors of color and brands are concatenated to the matrix of titles. The similarity measure is evaluated using this combined matrix.

Varieties of the tf– idf weighting plan are frequently utilized via web search tools as a focal apparatus in scoring and positioning an archive's pertinence given a client inquiry. tf– idf can be effectively utilized for stop-words separating in different subject fields, including content outline and order.

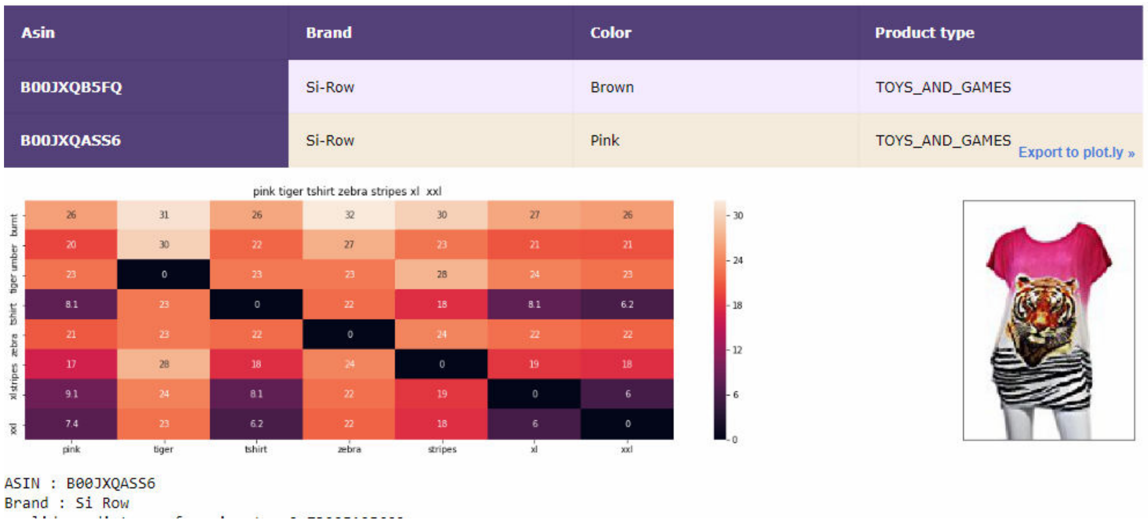


Asin	Brand	Color	Product type
B00JXQB5FQ	Si-Row	Brown	TOYS_AND_GAMES
B00JXQB5FQ	Si-Row	Brown	TOYS_AND_GAMES

Export to plot.ly »
burnt umber tiger tshirt zebra stripes xl xxl
burnt: 0 32 31 26 32 30 27 26
umber: 32 0 30 22 27 23 21 21
tiger: 31 30 0 23 23 28 24 23
tshirt: 26 22 23 0 22 18 8.1 6.2
zebra: 32 27 23 22 0 24 22 22
stripes: 30 23 28 18 24 0 19 18
xl: 27 21 24 8.1 22 19 0 6
xxl: 26 21 23 6.2 22 18 6 0
burnt umber tiger tshirt zebra stripes xl xxl
30 24 18 12 6 0

```
ASIN : B00JXQB5FQ
```

Fig. 3.13 Text Semantics Based Similarity using Brand and Color



Asin	Brand	Color	Product type
B00JXQB5FQ	Si-Row	Brown	TOYS_AND_GAMES
B00JXQASS6	Si-Row	Pink	TOYS_AND_GAMES

Export to plot.ly »
pink tiger tshirt zebra stripes xl xxl
burnt: 26 31 26 32 30 27 26
umber: 20 30 22 27 23 21 21
tiger: 23 0 23 23 28 24 23
tshirt: 8.1 23 0 22 18 8.1 6.2
zebra: 21 23 22 0 24 22 22
stripes: 17 28 18 24 0 19 18
xl: 9.1 24 8.1 22 19 0 6
xxl: 7.4 23 6.2 22 18 6 0
pink tiger tshirt zebra stripes xl xxl
30 24 18 12 6 0

```
ASIN : B00JXQASS6
Brand : Si Row
```

Fig. 3.14 Text Semantics Based Similarity using Brand and Color(Cont)

3.6 DEEP LEARNING BASED VISUAL PRODUCT SIMILARITY

The visual product similarity is evaluated using the images of the product. To achieve this successfully, a popular deep learning concept vis. Neural Networks were used.

3.6.1 NEURAL NETWORKS

Neural Networks are an endeavor at demonstrating the data handling abilities of sensory systems [28]. A Neural Network depends on a gathering of associated units or hubs called artificial neurons.

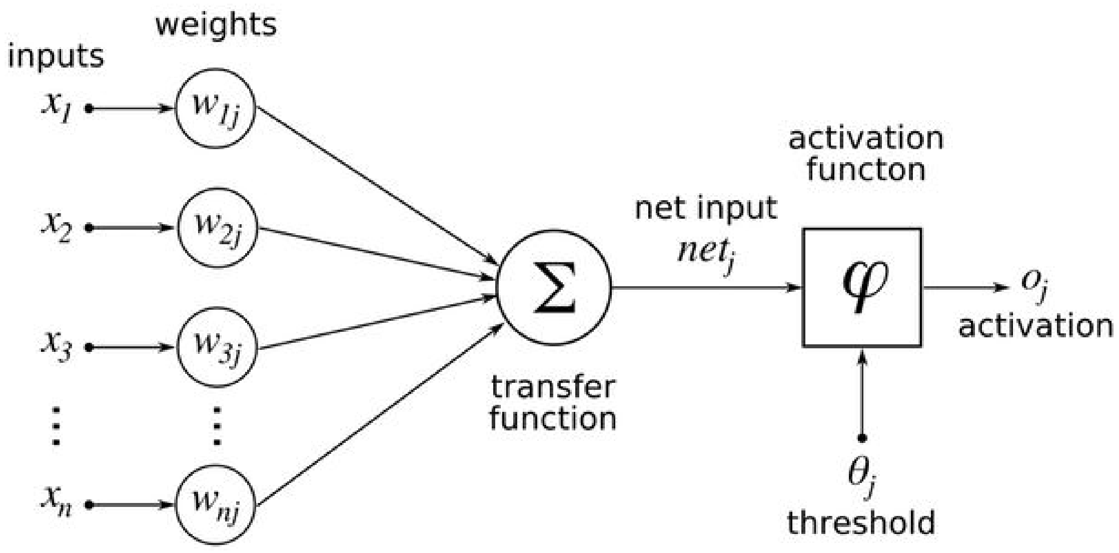


Fig 3.15 Artificial Neural Network: Representation and Activation Function

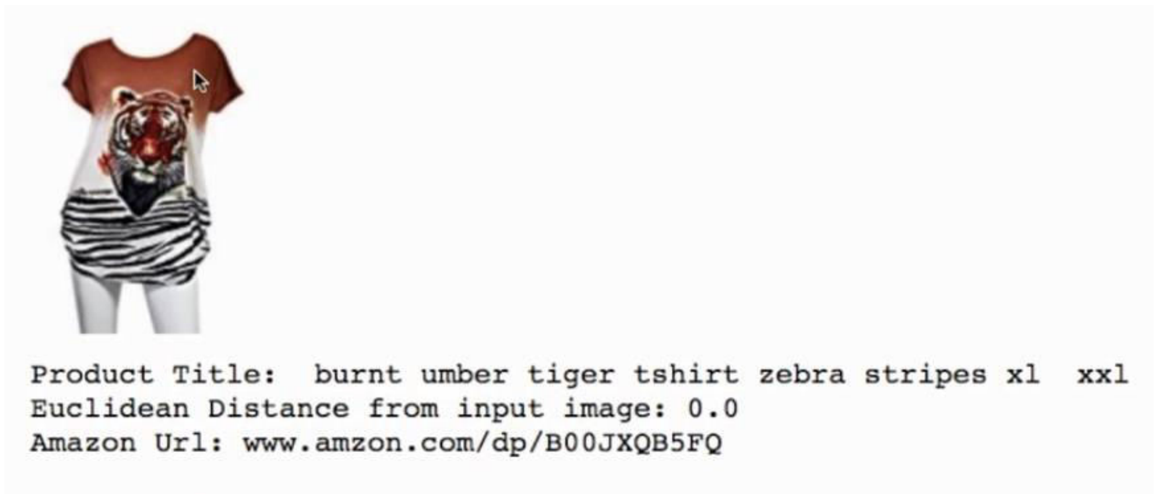
The activation function/transfer function of a node characterizes the yield of that node given an input or set of sources of inputs.

3.6.2 CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (CNN or ConvNet) is a class of deep, feed-forward artificial neural networks that has effectively been connected to investigating visual imagery. CNNs utilize a variety of multilayer perceptrons intended to require insignificant preprocessing [29]. They are otherwise called move invariant or space invariant simulated neural systems (SIANN).

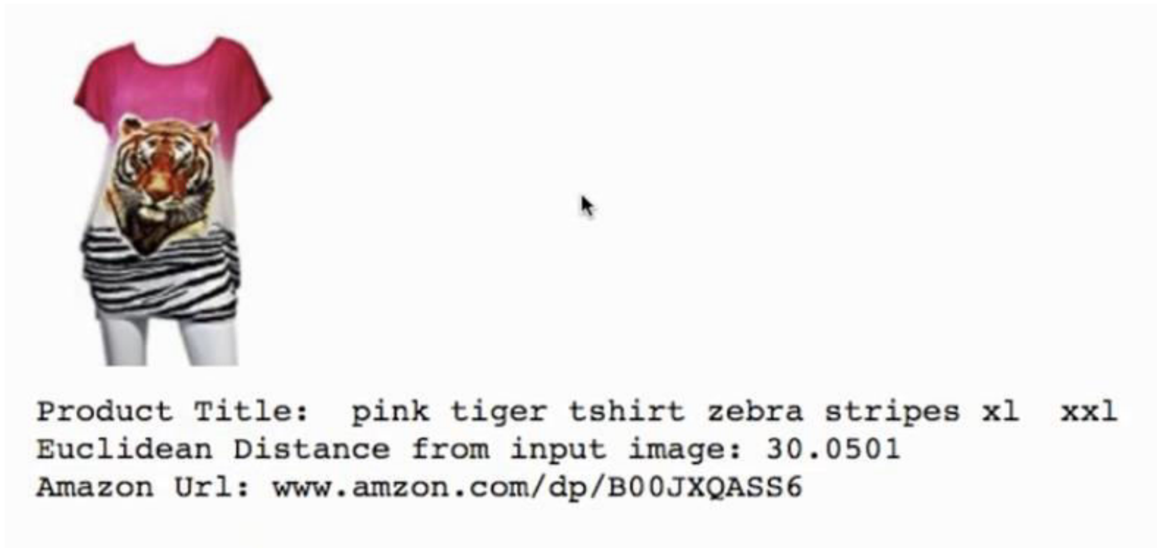
3.6.3 VISUAL BASED PRODUCT SIMILARITY

To evaluate the visual similarity of images of the products, few deep learning techniques are adopted. The image of each product is transformed in to dense matrix. Using matrices of each product image, Euclidean distance is calculated.



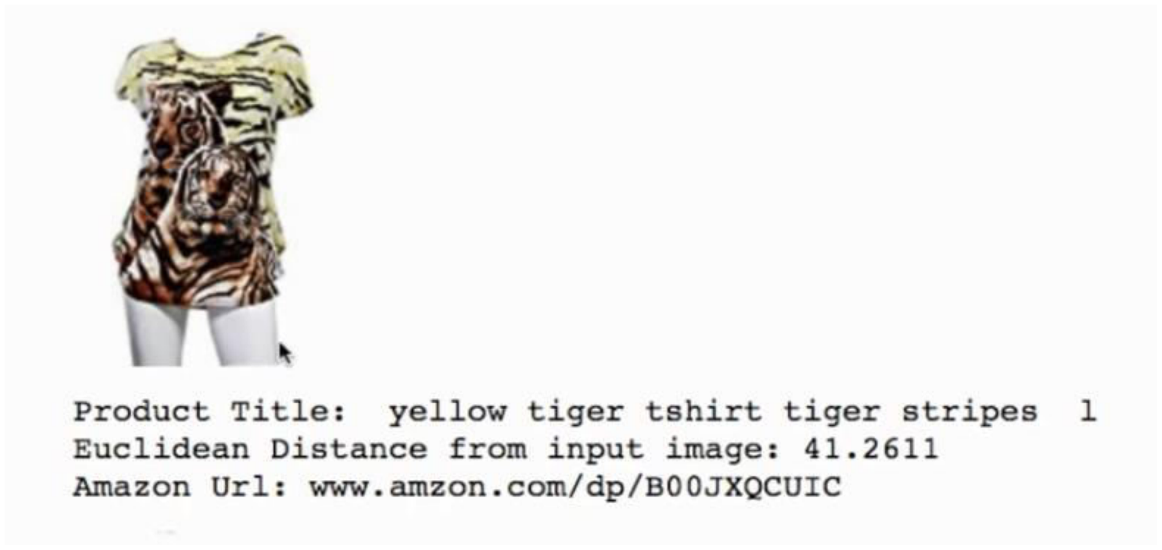
```
Product Title:  burnt umber tiger tshirt zebra stripes xl  xxl
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B00JXQB5FQ
```

Fig. 3.16 Visual Based Similarity Results (Target Product)



```
Product Title:  pink tiger tshirt zebra stripes xl  xxl
Euclidean Distance from input image: 30.0501
Amazon Url: www.amzon.com/dp/B00JXQASS6
```

Fig 3.17 Visual Based Similarity Results (Recommended Product)



```
Product Title:  yellow tiger tshirt tiger stripes  1
Euclidean Distance from input image: 41.2611
Amazon Url: www.amzon.com/dp/B00JXQCUIC
```

Fig. 3.18 Visual Based Similarity Results (Recommended Product)

CHAPTER 4
RESULTS AND DISCUSSION

4.1 INTRODUCTION

To determine the most similar match to a given product, the algorithm constructs a similar-items table by finding the Euclidean Distance. One could construct an item-to-item matrix by iterating through all possible item pairs and computing the similarity metric for each pair. It's possible to compute the similarity between two items in various ways. Popular approaches include Euclidean Distance and Cosine Measure. Euclidean Distance is the geometrical distance between the two products in question. In Cosine Measure each item is represented as a vector in n-Dimensional Space. This offline computation of this similarity matrix is computationally intensive, with $O(N2M)$ being the worst case.

4.2 SCALABILITY

Amazon.com has in excess of 29 million clients and a few million listed things. Other major retailers have similarly huge information sources. While this information offers opportunity, it's additionally a curse, crushing the spirits of calculations outlined for data sets three times of size smaller. All current calculations were assessed over little informational collections. For instance, the MovieLens information set [30] contains 35,000 clients and 3,000 things, what's more, the EachMovie information set [31] contains 4,000 clients and 1,600 things. For very large data sets, a scalable recommendation algorithm must perform the most expensive calculations offline [32].

4.3 IMPLEMENTATION DETAILS

The primary objective of this project was to construct a content based filtering system, which recommends similar products to the user based on their profile. The original data set contained roughly 160k products. The list was reduced to roughly 16k after removal of duplicate products, and removal of NULL valued rows. The titles of these products was simplified for the purpose of this project using traditional Natural Language Processing techniques, such as, tokenization, stop word removal etc. The similarity of the titles was evaluated using Euclidean Distance. The text based visual similarity was used to recommend the products whose title were the closest to the product in question. Text Semantics based similarity focused in sentence structure as well as the semantics behind it. Visual Similarity was evaluated by adopting Deep Learning techniques, vis. Convolutional Neural Networks.

CHAPTER 5
CONCLUSION AND FUTURE WORKS

5.1 INTRODUCTION

Recommendation algorithms provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. For large retailers like Amazon.com, a good recommendation algorithm is scalable over very large customer bases and product catalogs, requires only subsecond processing time to generate online recommendations, is able to react immediately to changes in a user's data, and makes compelling recommendations for all users regardless of the number of purchases and ratings. Unlike other algorithms, item-to-item collaborative filtering is able to meet this challenge. In the future, we expect the retail industry to more broadly apply recommendation algorithms for targeted marketing, both online and offline. While e-commerce businesses have the easiest vehicles for personalization, the technology's increased conversion rates as compared with traditional broad-scale approaches will also make it compelling to offline retailers for use in postal mailings, coupons, and other forms of customer communication.

5.2 CONCLUSION

In this project we reviewed the field of content based recommender systems, by giving an overview of the most imperative aspects portraying that sort of systems. Despite the fact that there is a group of recommender systems in various spaces, they share in common a method for representing items to be suggested and client profiles. We dissected the principle content recommender systems created in the past years, by featuring the purposes behind which a more mind boggling "semantic analysis" of content is required to go past the syntactic analysis of client interests given by keywords. An audit of the fundamental methodologies (and frameworks) received to present some semantics in the proposal procedure is done, by providing confirmation of the main part of etymological learning, regardless of whether a more particular learning is obligatory for a more profound comprehension and contextualization of the client interests in various application areas. The most recent issues in cutting edge content portrayal utilizing wellsprings of world information, for example, Wikipedia, have been featured, yet they have not yet utilized as a part of the setting of learning client profiles.

5.3 FUTURE WORKS

5.3.1 ROLE OF USER GENERATED CONTENT IN RECOMMENDATION PROCESS

Web 2.0 is a term used to describe the trend in the use of World Wide Web technology that focuses at promoting information sharing and collaboration among users. According to Tim O'Reilly, the term "Web 2.0" means putting the user in the center, designing software that critically depends on its users since the content, as in Flickr, Wikipedia, Del.icio.us, or YouTube, is contributed by thousands or millions of users. That is why Web 2.0 is also called the "participative Web". O'Reilly also defined Web 2.0 as "the design of systems that get better the more people use them".

One of the forms of User Generated Content (UGC) that has piqued the interest of the research community is folksonomy, a taxonomy generated by users who collaboratively annotate and categorize resources of interests with freely chosen keywords called tags.

Several methodologies have been thought for including user tagging activity within content-based recommendation systems. The user profile is represented in the form of a tag vector, with each element indicating the number of times a tag has been assigned to a document by that user. A more sophisticated approach is proposed in [33], which takes into account tag co-occurrence. The matching of profiles to information sources is achieved by using simple string matching. As the authors themselves foresee, the matching could be enhanced by adopting WORDNET.

Folksonomies give new openings and difficulties in the field of recommender systems. It ought to be researched whether they may be a significant source of data about client interests and whether they could be incorporated into client profiles. Undoubtedly, a few challenges of tagging systems have been recognized, for example, polysemy and synonymy of labels, or the diverse mastery and motivations behind labeling members that may bring about labels at different levels of deliberation to depict an asset, or the turbulent expansion of labels

5.3.1.1 SOCIAL TAGGING RECOMMENDER SYSTEMS

A few strategies have been proposed for considering client labeling movement inside content based recommender systems.

In [34], the client profile is spoken to as a tag vector, with every component showing the number of times a tag has been allocated to a record by that client. A more modern approach is proposed in [35], which considers tag co-occurrence. The coordinating of profiles to data sources is accomplished by utilizing straightforward string operations. As the creators themselves anticipate, the coordinating could be upgraded by embracing WORDNET.

In the work by Szomszor et al. [36], the creators portray a movie recommendation system constructed absolutely on the keywords doled out to films by means of collaborative labeling. Suggestions for the active client are delivered by calculations in light of the likeness between the keywords of a film and those of the tag-clouds of motion pictures she appraised. As the creators themselves express, their recommendation algorithms can be enhanced by joining tag-based profiling systems with more conventional content based recommender procedures.

.

.

REFERENCES

[1] Michael J. Pazzani and Daniel Billsus: "*Content-Based Recommendation Systems*" Chapter 10 (326-327)

[2] Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The Adaptive Web, pp. 291–324. Springer Berlin / Heidelberg (2007)

[3] Schafer, J.B., Konstan, J.A., Riedl, J.: E-commerce recommendation applications. Data Mining and Knowledge Discovery 5(1/2), 115–153 (2001)

[4] Francesco Ricci, Lior Rokach and Bracha Shapira: Recommender System Handbooks Second Edition (1-25)

[5] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley (1999)

[6] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro: Content-based Recommender Systems: State of the Art and Trends(76-80)

[7] Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: P.Brusilovsky, A. Kobsa, W. Nejdl (eds.) The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, pp. 325–341 (2007). ISBN 978-3-540-72078-2

[8] Salton, G.: Automatic Text Processing. Addison-Wesley (1989)

[9] Magnini, B., Strapparava, C.: Improving User Modelling with Content-based Techniques. In: Proceedings of the 8th International Conference of User Modeling, pp. 74–83. Springer (2001)

[10] Degemmis, M., Lops, P., Semeraro, G.: A Content-collaborative Recommender that Exploits WordNet-based User Profiles for Neighborhood Formation. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI) 17(3), 217–255(2007). Springer Science + Business Media B.V

[11] Semeraro, G., Basile, P., de Gemmis, M., Lops, P.: User Profiles for Personalizing Digital Libraries. In: Y.L. Theng, S. Foo, D.G.H. Lian, J.C. Na (eds.) Handbook of Research on Digital Libraries: Design, Development and Impact, pp. 149–158. IGI Global (2009). ISBN978-159904879-6

[12] Semeraro, G., Degemmis, M., Lops, P., Basile, P.: Combining Learning and Word Sense Disambiguation for Intelligent User Profiling. In: M.M. Veloso (ed.) Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 2856–2861 (2007). ISBN 978-I-57735-298-3

[13] Basile, P., Degemmis, M., Gentile, A., Lops, P., Semeraro, G.: UNIBA: JIGSAW algorithm for Word Sense Disambiguation. In: Proceedings of the 4th ACL 2007 International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, pp. 398–401. Association for Computational Linguistics (2007)

[14] Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological User Profiling in Recommender Systems. ACM Transactions on Information Systems 22(1), 54–88 (2004)

[15] Gabrilovich, E., Markovitch, S.: Overcoming the Brittleness Bottleneck using Wikipedia:Enhancing Text Categorization with Encyclopedic Knowledge. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, pp. 1301–1306. AAAI Press (2006)

[16] Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In: M.M. Veloso (ed.) Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)

[17] Giles, J.: Internet Encyclopaedias Go Head to Head. Nature 438, 900–901 (2005)

[18] Csomai, A., Mihalcea, R.: Linking Documents to Encyclopedic Knowledge. IEEE Intelligent Systems 23(5), 34–41 (2008)

[19] Mihalcea, R., Csomai, A.: Wikify!: Linking Documents to Encyclopedic Knowledge. In:Proceedings of the sixteenth ACM conference on Conference on

Information and Knowledge Management, pp. 233–242. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1321440.1321475. ISBN 978-1-59593-803-9

[20] Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34 (1) (2002)

[21] Billsus, D., Pazzani, M.: Learning Probabilistic User Models. In: Proceedings of the

Workshop on Machine Learning for User Modeling. Chia Laguna, IT (1997). http://citeseer.nj.nec.com/billsus96learning.html

[22] Lewis, D.D., Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 81–93. Las Vegas, US (1994)

[23] Rocchio, J.: Relevance Feedback Information Retrieval. In: G. Salton (ed.) The SMART retrieval system - experiments in automated document processing, pp. 313–323. Prentice-Hall, Englewood Cliffs, NJ (1971)

[24] Chen, L., Sycara, K.: WebMate: A Personal Agent for Browsing and Searching. In: K.P.

Sycara, M. Wooldridge (eds.) Proceedings of the 2nd International Conference on Autonomous Agents, pp. 9–13. ACM Press, New York (1998)

[25] Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34 (1) (2002)

[26] Montaner, M., Lopez, B., Rosa, J.L.D.L.: A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review 19 (4), 285–330 (2003)

[27] Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: P. Brusilovsky, A. Kobsa, W. Nejdl (eds.) The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, pp. 325–341 (2007). ISBN 978-3-540-72078-2

[28] R. Rojas: Neural Networks, Springer-Verlag, Berlin, 1996

[29] LeCun, Yann: LeNet-5, Convolutional Neural Networks". Retrieved 16 November2013

[30] B.M. Sarwarm et al., Analysis of Recommendation Algorithms for E-Commerce, *ACM Conf. Electronic Commerce*, ACM Press, 2000, pp.158-167.

[31] J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc. 14th Conf. Uncertainty in Artificial Intelligence*, Morgan Kaufmann, 1998, pp. 43-52

[32] Greg Linden, Brent Smith, and Jeremy York: Amazon.com Recommendations.

[33] Michlmayr, E., Cayzer, S.: Learning User Profiles from Tagging Data and Leveraging them for Personal(ized) Information Access. In: Proc. of the Workshop on Tagging and Metadata for Social Information Organization, Int. WWW Conf. (2007)

[34] Diederich, J., Iofciu, T.: Finding Communities of Practice from User Profiles Based On Folksonomies. In: Innovative Approaches for Learning and Knowledge Sharing, EC-TEL Workshop Proc., pp. 288–297 (2006)

[35] Michlmayr, E., Cayzer, S.: Learning User Profiles from Tagging Data and Leveraging them for Personal(ized) Information Access. In: Proc. of the Workshop on Tagging and Metadata for Social Information Organization, Int. WWW Conf. (2007)

[36] Szomszor, M., Cattuto, C., Alani, H., O'Hara, K., Baldassarri, A., Loreto, V., Servedio, V.D.P.: Folksonomies, the Semantic Web, and Movie Recommendation. In: Proceedings of the Workshop on Bridging the Gap between Semantic Web and Web 2.0 at the 4th ESWC (2007).