Duplicate product detection

Machine learning approach to detect duplicates in product database

Raghu Cheekatla, raghu5rvm@gmail.com,

Ph: 9912026326, madivala-Banglore.

http://www.github.com/raghu5rvm/infilectme

Table of contents:

Index of figures

0. Introduction

There could be a lot of reasons why duplicate data exists, still it is not desirable to have redundant data in most cases (except cases like Backups, hdfs…). The idea of filtering duplicates involves a lot of heuristic application and correct understanding of data. This reduces the amount of data to process in a substantial quantity and is quite useful in dealing with large datasets. Recommendation systems with high accuracy and low response time is useful to no one, as the user surfs too fast and might see recommendations to an item that they viewed 5min back. Bearing in mind all these constraints data processing should be done efficiently as well as quickly.

1. Motivation and objective

As a recent graduate and a machine learning enthusiast, it would be a great start for my career to work with you. My exposure to machine learning and Artificial intelligence is driving me to learn and experiment more in those fields. It was fun and I really like doing the task first-of-all and hope to have more fun with you in the coming days. Looking forward to hear from you!!!

The object is to identify duplicate product details in given data from an e-commerce database. Other than identifying it is expected to calculate the similarity between two products. Finally the observations and results are to be documented and saved for further references.

2. Assumptions

- Product- a product is a physical entity which has definite description, parameters, cost and other sufficient details.
- Different sellers uploading same product are not evaluated for similarity (from objective page guidelines)
- Duplicate product – a product is considered to be duplicate of another if it satisfies below criteria
    - Both have similar description

- Both look alike
- Key features of both products match to most extent
- Difference in sizes but same look
- Difference in color but same look

3. Data preparation

3.1. Data accusation

The data set is provided by Infilect, in a compressed format. It was then extracted to obtain raw data of around 40lakh rows and 32 columns in a csv format of 5GiB.

3.2. Technology stack

- Python
- R

3.3 Data Visualization

It is very essential to understand data before trying to operate on it. Insights from data visualization help us reduce the amount of processing burden and sometimes avoid complex things which are not needed. The given data set is loaded and view in R-Studio to gain hidden insights from it. R is a great tool for data visualization and processing too. The “largeData.csv” file consists of xyz rows and 32 columns. The columns are named as follows

- Pid
  A unique id of reference to each product in database
- Title
  Product title, one short sentence describing what product is
- Description
  This specifies the products short description like where it is suitable and on what other apparels
- imageUrlStr
  Multiple links separated by a semi-colon which are original images of product in multiple sizes
- mrp
  Maximum retail price of product
- sellingPrice
  Selling price of product. This could vary from mrp (it is very much large in some cases in this dataset)
- specialPrice

Special price of product in cases of festive offers or other promotional offers

- productUrl
  A link pointing to the actual product in e-commerce website(flipkart,amazon…)
- categories
  Denotes the product category, this comprises of sub-categories too
- productBrand
  Product manufacturing brand name.
- productFamily
  List of products which are similar and this is useful in identifying duplicates
- inStock
  Whether the product is available in stock and ready for shipment or not.
- codeAvailable
  any promotional codes available or not
- offers
  Description of any offer related information.
- Discount
  Discount on actual price .
- shippingCharges
  Any shipping charges if apply
- size
  Size of appropriate product (in multiple standards)
- deliveryTime
  Expected delivery time of the product if ordered today
- color
  Color specifications of product
- sizeUnit
  The size could be in UK standards or in US standards
- storage
  A Boolean valued column specifying info related to storage.
- displaySize
  When multiple sizes are available, what is the size to show first when user visits the page
- keySepcsStr
  Key specifications of a product
- detailedSpecsStr
  Detailed specifications of a product
- specificationList
  More specification describing the product
- sellerName
  Seller name of the particular product

- sellerAvgRating
  Average user rating of particular seller
- sellerNoOfRating
  Number of ratings which constitute to the average ratings of seller
- Sleeve
  Sleeve related info of a product
- Neck
  Neck related info of a product
- idealFor
  Whether the product is ideal for men or for women

3.4 Filtering appropriate data

As specified in the objective, we have to find duplicate "tops" from given dataset. For this, we better separate the details of tops from other products. This makes our analysis easier and reduces the amount of data to analyze. For filtering data such that products which are "tops" as specified in "categories" filed a python script could be useful and below is the sample code

```
import csv
cList=[0]*100
topI=[]
with open('data/largeData.csv', 'rb') as f:
	dataSet = csv.reader(f,delimiter=',')
	with open('data/filteredLarge.csv', 'w+') as opf:
		writer = csv.writer(opf)
		i=0
		j=0
		topCount=0
		bugs=[]
		for row in dataSet:
			#print (str(i)+"\t"+row[0]+"\t"+row[8]+"\t"+str(len(subs)-
			#1)+"\n")
			#print subs[len(subs)-1]
			cList[len(row)]+=1
			if(i==0):
				writer.writerow(row)
			elif(len(row)<32):
				print (i," some ",len(row),"columns found out of 32\n")
				j+=1
			else:
				subs = row[8].split(">")
				if(len(subs)>0 and subs[len(subs)-1] == 'Tops'):
					writer.writerow(row)
					topCount+=1
					topI.append(i)
			i+=1
```

This python script reads each row from given dataset and scans for the presence of “tops” under “category” column. If the result is positive, the appropriate row is written into a new csv file which can be used further for analysis. The original data of dimensions [4057189] x [32] is reduced to [347694] x [32]] after successful filtering.

```
>
> dim(largeData)
[1] 4057189      32
>
```

Dimensions of original data fig3.4

3.5 Summarizing filtered data

After careful analysis of data which was obtained after proper filtering, below are the summaries of each column given in data set which are found useful. The following insights are derived after loading and analyzing data in R

- PID
  .1. Unique values of alpha numeric string
  .2. 10 rows with missing pids
  .3. Key field in generating output data file
- Title
  .1. String describing product title
  .2. May not be so useful to identify duplicates since multiple products have same title
  .3. 7 rows with missing titles
- Description
  .1. String describing product summary in short
  .2. 1.3Lakh rows with missing description
  .3. Not useful since, most rows lack description
- imageUrlStr
  .1. multiple links to images of product samples separated by semi-colons
  .2. Useful to cross check the results and can be used in advanced techniques to find similarity
  .3. 18 rows with missing image url links
- Mrp
  .1. Numerical values representing max retail prices of products
  .2. May be useful in finding products in similar range of prices

.3. 9 rows with missing mrps

- Selling Price
  .1. Numerical values representing actual selling prices of products
  .2. May be useful in finding products in similar range of prices
  .3. 7 rows with missing selling prices
- Specialprice
  .1. Numerical values representing special prices of products
  .2. May be useful in finding products in similar range of prices
  .3. 8 rows with missing special prices
- ProductUrl
  .1. Links denoting actual product online in e-commerce site
  .2. Might be useful in checking product details for verification and testing
  .3. 10 rows with missing product urls
- Categories
  .1. Categorization of given product into sub-categories till an atomic level which describes a product
  .2. Useful in finding similar products
  .3. 2 missing values
- productBrand
  .1. Brand could point to identify similar products
  .2. 8 missing values
- productFamily
  .1. This is found useful in identifying similar products from a seller
  .2. Consists of multiple product ids
  .3. 17 missing values
- inStock
  .1. Duplicates of a product may not depend on this factor, hence ignored
  .2. Most of the rows are missing too
- codeAvailable
  .1. May not be useful for duplicate detection, ignored
  .2. 11 missing values
- Offers
  .1. May not be affecting duplicate detection
  .2. Mostly missing from all rows
- Discount
  .1. May/may not be so useful for duplicate detection
  .2. 18 missing values
- Shipping charges
  .1. May/may not be useful for duplicate detection
  .2. 17 missing values

- Size
  .1. Could be useful in determining similar products
  .2. 21 missing values from rows
- Delivery time
  .1. May not be useful since most rows are missing hence, ignored
- Color
  .1. Could be a factor to identify similar products in different color
  .2. 666 rows with missing colors
- Size unit
  .1. May not be so useful in duplicate detection
  .2. 3.4 lakh rows with blank values for this
- Storage
  .1. No values in most rows hence, not useful
- Display Size
  .1. Not useful since most(3.4Lash) missing values in all rows
- keySpecsStr
  .1. Might be useful in identifying duplicates
  .2. 1033 missing values
- detailedSpecstr
  .1. May/may not be so useful in duplicate detection
  .2. Most of them are same as key specs from observed data
  .3. 1056 missing row values
- Specification List
  .1. Missing most values (3 Lakh) hence, not useful and ignored
- sellerName
  .1. Useful in identifying duplicated uploaded by a seller
  .2. 26 missing values
- sellerAvgRating & sellerNoOfReviews
  .1. May not be so useful since they are not related to product
- Sleeve
  .1. Could be useful to compare two products
  .2. 1897 missing values
- Neck
  .1. Could be useful like sleeve information
  .2. 5744 values missing
- Ideal for
  .1. Mostly missing from all rows hence, not useful and ignored

```
                                                                                                        :341233
      mrp          sellingPrice       specialPrice
 Min.   :    0   Min.   :    0.0   Min.   :    0.0
 1st Qu.:  895   1st Qu.:  449.0   1st Qu.:  431.0
 Median : 1098   Median :  581.0   Median :  559.0
 Mean   : 1216   Mean   :  674.9   Mean   :  664.7
 3rd Qu.: 1399   3rd Qu.:  799.0   3rd Qu.:  797.0
 Max.   :99999   Max.   :49990.0   Max.   :49990.0

                                                                                                                                  productUrl
 http://dl.flipkart.com/dl/anekdote-casual-3-4th-sleeve-solid-women-s-blue-white-top/p/itmenv7gmnuyqnz2?pid=TOPEKUW2FGBSVN3H           :     2
 http://dl.flipkart.com/dl/athena-casual-full-sleeve-printed-women-s-white-black-top/p/itmenv3svukss8nr?pid=TOPEG697C4GPFU8F          :     2
 http://dl.flipkart.com/dl/atheno-casual-sleeveless-solid-women-s-grey-top/p/itmenv6hjbwghcmb?pid=TOPEF5RVP2NMGGNK                    :     2
 http://dl.flipkart.com/dl/baloono-casual-3-4th-sleeve-printed-women-s-white-red-top/p/itmenv6uqq4wavrf?pid=TOPEKK5EZXGYRZF7          :     2
 http://dl.flipkart.com/dl/baloono-casual-sleeveless-striped-women-s-black-white-top/p/itmenv7t62g2eg6h?pid=TOPEJB6RHYZTYMVC          :     2
 http://dl.flipkart.com/dl/bhama-couture-casual-full-sleeve-floral-print-women-s-black-grey-top/p/itmenv46h2dctg7g?pid=TOPECQKMDQZYRHSG:     2
 (Other)                                                                                                                              :341751
                                                 categories            productBrand
 Apparels>Kids>Girls>T-Shirts & Tops>Tops                   :     1   Vero Moda   :  8769
 Apparels>Kids>Infants>Baby Girls>T-Shirts & Tops>Tops      :     0   Uptown 18   :  7969
 Apparels>Women>Fusion Wear>Shirts, Tops & Tunics>Tops      : 12296   Only        :  7917
 Apparels>Women>Maternity Wear>Shirts, Tops & Tunics>Tops:   1256   Harpa       :  6805
 Apparels>Women>Western Wear>Shirts, Tops & Tunics>Tops     :328210   Frenchtrendz:  4037
                                                                      Raindrops   :  3990
                                                                      (Other)     :302276
```

Summary of data fig3.5a

```
                                                                                                      :341751
    discount      shippingCharges        size             color
 Min.   : 0.00   Min.   :  0.00   M        :72205   Black     : 42255
 1st Qu.:30.00   1st Qu.:  0.00   L        :72098   Blue      : 27463
 Median :49.00   Median :  0.00   S        :67510   Multicolor: 26656
 Mean   :41.97   Mean   : 12.89   XL       :62577   White     : 26583
 3rd Qu.:58.00   3rd Qu.:  0.00   XS       :22768   Pink      : 16935
 Max.   :94.00   Max.   :160.00   2XL      :13885   Red       : 14330
                                  (Other):30720   (Other)   :187541
                                                                      keySpecsStr
 Round Neck, Short Sleeve;Fabric: Cotton;Pattern: Printed;Type: Top;Pack of 1       :  7353
 Round Neck, Sleeveless;Fabric: Cotton;Pattern: Solid;Type: Top;Pack of 1           :  5906
 Round Neck, Short Sleeve;Fabric: Cotton;Pattern: Solid;Type: Top;Pack of 1         :  5881
 Round Neck, Short Sleeve;Fabric: Cotton;Pattern: Printed;Type: Crop top;Pack of 1:  5682
 Round Neck, Full Sleeve;Fabric: Cotton;Pattern: Solid;Type: Top;Pack of 1          :  4832
 Round Neck, Sleeveless;Fabric: Cotton;Pattern: Printed;Type: Top;Pack of 1         :  4216
 (Other)                                                                            :307893
                                                                   detailedSpecsStr                           sellerName
 Round Neck, Short Sleeve;Fabric: Cotton;Pattern: Printed;Type: Top;Pack of 1       :  7353   MAyur Karwa                 : 20262
 Round Neck, Sleeveless;Fabric: Cotton;Pattern: Solid;Type: Top;Pack of 1           :  5906   Kapsons Agencies Pvt. Ltd.  : 16816
 Round Neck, Short Sleeve;Fabric: Cotton;Pattern: Solid;Type: Top;Pack of 1         :  5881   Satvinder Singh             :  8462
 Round Neck, Short Sleeve;Fabric: Cotton;Pattern: Printed;Type: Crop top;Pack of 1:  5682   Hindustan Online Trade Pvt Ltd:  7942
 Round Neck, Full Sleeve;Fabric: Cotton;Pattern: Solid;Type: Top;Pack of 1          :  4832   Abhijeet Kumar              :  6456
 Round Neck, Sleeveless;Fabric: Cotton;Pattern: Printed;Type: Top;Pack of 1         :  4216   SUNIL KUMAR                 :  5800
 (Other)                                                                            :307893   (Other)                     :276025
          sleeve               neck
 Sleeveless    :109336   Round Neck:223219
 Short Sleeve  : 87193   V-Neck    : 41700
 3/4th Sleeve  : 68667   Boat Neck : 16974
 Full Sleeve   : 50718   U Neck    :  9176
 Cap Sleeve    : 13886   High Neck :  8051
 Roll-up Sleeve:  3527   V Neck    :  5005
 (Other)       :  8436   (Other)   : 37638
```

Summary of data fig3.5b

3.6 Data cleaning

3.6.1 Deleting useless columns

Delete all useless columns from the observations in data summary and visualization. Keep the rows which are of significant use in identifying duplicates form given product database. Below is the R code to exclude rows which are of very less/no priority for given objecting

```
> notUsefulIndex <- c(3,12,13,14,17,20,21,22,25,27,28,29,32)
#("inStock","codAvailable","offers","deliveryTime","sizeUnit","storage","disp
laySize","sellerAverageRati#-
ng","sellerNoOfRatings","sellerNoOfReviews","idealFor","specificationList"

> data.usefulColsOnly <- data.spacesFilled[,-notUsefulIndex]
```

3.6.2 Replace blank values with NAs

Blank values are not easy to track in R. So, all blank columns are replaced with NAs and then are analyzed further.

```
> oldData[oldData==""] <- NA
```

productId	title	description
TOPE9ABBZU3HZRHN	Citrine Casual Short Sleeve Printed Women's Pink, ...	This beautiful printed modal top from Citrine is soft ...
TOPE9ABBBTJYDSQE	Citrine Casual Short Sleeve Printed Women's Pink, ...	This beautiful printed modal top from Citrine is soft ...
TOPE9AZZSMSZFYAM	Leelan Casual Short Sleeve Solid Women's Black Top	
TOPE6ZCYFCQ3H6EV	Cottinfab Casual Sleeveless Solid Women's Purple, ...	Round neck, sleeveless stylish top with pack of 3 sets.
TOPE6ZCYHTJEMZMW	Cottinfab Casual Sleeveless Solid Women's Purple, ...	V Neck with black net on front yoke, sleeveless, soli...
TOPE6XZPUVT9C7RU	Butterfly Wears Casual Short Sleeve Solid Women's ...	
TOPE6Y7HSDDXPHZN	Butterfly Wears Casual Short Sleeve Solid Women's ...	
TOPE6XZPXBP5APH9	Butterfly Wears Casual Short Sleeve Solid Women's ...	
TOPE6XZPRUAFWPBH	Butterfly Wears Casual Short Sleeve Solid Women's ...	
TOPE6XZP5XW5NHZA	Butterfly Wears Casual Full Sleeve Solid Women's Ye...	
TOPE8BZACHSUMG6U	Taurus Casual Sleeveless Self Design Women's Gree...	
TOPE8FZ32WFGWAUF	Nagpal Radio Corp Casual Sleeveless Solid Women's...	
TOPE7CD4ETPFHCDX	Color Cocktail Casual Full Sleeve Printed Women's M...	
TOPE7CD4FZXYEY2F	Color Cocktail Casual Full Sleeve Printed Women's M...	
TOPE6GAURXUQSGHN	Aarr Casual Sleeveless Polka Print Women's Orange ...	This beautiful top For all those who want a simply st...
TOPE8M6R2XZCZG8Z	Taurus Casual Sleeveless Printed Women's Green Top	
TOPE8M6RMN7SBFVG	Taurus Casual Sleeveless Printed Women's Green Top	
TOPE74AWETGFAHGN	Namakh Casual 3/4th Sleeve Solid Women's Blue Top	Get a stylish casual look with this great top from Na...
TOPE74AWYBAAUUCQ	Namakh Casual Sleeveless Printed Women's Pink Top	Get a stylish casual look with this great top from Na...
TOPE7BDHRF6CHJ7G	Vivante by VSA Casual Full Sleeve Printed Women's ...	Beautiful abstract flowers in wildly beautiful colors o...

Original data with blank columns fig3.6a

TOPE9ABBZU3HZRHN	Citrine Casual Short Sleeve Printed Women's Pink, ...	This beautiful printed modal top from Citrine is soft ...
TOPE9ABBBTJYDSQE	Citrine Casual Short Sleeve Printed Women's Pink, ...	This beautiful printed modal top from Citrine is soft ...
TOPE9AZZSMSZFYAM	Leelan Casual Short Sleeve Solid Women's Black Top	*NA*
TOPE6ZCYFCQ3H6EV	Cottinfab Casual Sleeveless Solid Women's Purple, ...	Round neck, sleeveless stylish top with pack of 3 sets.
TOPE6ZCYHTJEMZMW	Cottinfab Casual Sleeveless Solid Women's Purple, ...	V Neck with black net on front yoke, sleeveless, soli...
TOPE6XZPUVT9C7RU	Butterfly Wears Casual Short Sleeve Solid Women's ...	*NA*
TOPE6Y7HSDDXPHZN	Butterfly Wears Casual Short Sleeve Solid Women's ...	*NA*
TOPE6XZPXBP5APH9	Butterfly Wears Casual Short Sleeve Solid Women's ...	*NA*
TOPE6XZPRUAFWPBH	Butterfly Wears Casual Short Sleeve Solid Women's ...	*NA*
TOPE6XZP5XW5NHZA	Butterfly Wears Casual Full Sleeve Solid Women's Ye...	*NA*
TOPE8BZACHSUMG6U	Taurus Casual Sleeveless Self Design Women's Gree...	*NA*
TOPE8FZ32WFGWAUF	Nagpal Radio Corp Casual Sleeveless Solid Women's...	*NA*
TOPE7CD4ETPFHCDX	Color Cocktail Casual Full Sleeve Printed Women's M...	*NA*
TOPE7CD4FZXYEY2F	Color Cocktail Casual Full Sleeve Printed Women's M...	*NA*
TOPE6GAURXUQSGHN	Aarr Casual Sleeveless Polka Print Women's Orange ...	This beautiful top For all those who want a simply st...
TOPE8M6R2XZCZG8Z	Taurus Casual Sleeveless Printed Women's Green Top	*NA*
TOPE8M6RMN7SBFVG	Taurus Casual Sleeveless Printed Women's Green Top	*NA*
TOPE74AWETGFAHGN	Namakh Casual 3/4th Sleeve Solid Women's Blue Top	Get a stylish casual look with this great top from Na...
TOPE74AWYBAAUUCQ	Namakh Casual Sleeveless Printed Women's Pink Top	Get a stylish casual look with this great top from Na...
TOPE7BDHRF6CHJ7G	Vivante by VSA Casual Full Sleeve Printed Women's ...	Beautiful abstract flowers in wildly beautiful colors o...

Data after filling blanks with NAs fig3.6a

3.6.3 Remove rows with incomplete data

Below is the code snippet to remove rows with insufficient data or rows which have NAs in fields which are keys in finding duplicates.

```
dropIncomplete <- function(data, variables){
  completeIndexes <- complete.cases(data[,variables])
  return(data[completeIndexes, ])
}
```

3.6.4 Correct data types

Filtered data contains every value in String format. Hence we need to format them so that each column has appropriate data types.

```
> data.correctDataTypes$mrp <- as.numeric(data.correctDataTypes$mrp)

> data.correctDataTypes$sellingPrice <-
as.numeric(data.correctDataTypes$sellingPrice)

> data.correctDataTypes$specialPrice <-
as.numeric(data.correctDataTypes$specialPrice)

> data.correctDataTypes$discount <-
as.numeric(data.correctDataTypes$discount)

> data.correctDataTypes$shippingCharges <-
as.numeric(data.correctDataTypes$shippingCharges)

> data.correctDataTypes$shippingCharges <-
as.numeric(data.correctDataTypes$shippingCharges)
```

3.6.5 Sort data

The objective is to identify duplicate product details uploaded by any seller in same or multiple e-commerce sites. Hence, it is required to sort data set accordingly for faster search of duplicated. Below is the code to sort data by multiple columns

```
> data.sorted <- data.correctDataTypes[with(data.correctDataTypes,
order(sellerName,productBrand,mrp)),]
```

3.6.6 Export to csv

After cleaning data from basic analysis, check further for presence of any other errors/missing data. This could be done by randomly selecting data from whole data set and analyzing it. After successful cleaning of data, it is now ready to be exported. Below is the sample code to export data into csv from R.

```
> write.csv(data.sorted,file="/home/raghu/Desktop/assign/dataCleanedLarge.csv",
sep=",",col.names=TRUE)
```

4 Data processing

4.1 Reading data into python

The data obtained after filtering and cleaning is ready to be analyzed. We are using python(2.7) for this purpose. Below is a simple code to read csv file into a python script

```
with open('data/largeData.csv', 'rb') as f:
      dataSet = csv.reader(f,delimiter=',')
            for row in dataset:
                  print (row)
```

4.2 Gensim and word2vec

Gensim is a free Python library designed to automatically extract semantic topics from documents, as efficiently (computer-wise) and painlessly (human-wise) as possible. **Gensim** is designed to process raw, unstructured digital texts. It includes implementations of tf-idf, random projections, word2vec and document2vec algorithms, hierarchical Dirichlet processes (HDP), latent semantic analysis (LSA, LSI, SVD). Word2vec is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep nets can understand. Deeplearning4j implements a distributed form of Word2vec for Java and Scala, which works on Spark with GPUs. Word2vec's applications extend beyond parsing sentences in the wild. It can be applied just as well to genes, code, likes, playlists, social media graphs and other verbal or symbolic series in which patterns may be discerned. Given enough data, usage and contexts, Word2vec can make highly accurate guesses about a word's meaning based on past appearances . Those guesses can be used to establish a word's association with other words, or cluster documents and classify them by topic. Those clusters can form the basis of search, sentiment analysis and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management. The output of the Word2vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep-learning net or simply queried to detect relationships between words.

Here is a sample code which loads Google's genism and converts test to vectors

```
model =
gensim.models.KeyedVectors.load_word2vec_format('packages/gensim.bin',
binary=True)

index2word_set = set(model.wv.index2word)

def avg_feature_vector(sentence, model, num_features, index2word_set):

    words = sentence.split()

    feature_vec = np.zeros((num_features, ), dtype='float32')

    n_words = 0

    for word in words:

        if word in index2word_set:

            n_words += 1

            feature_vec = np.add(feature_vec, model[word])

    if (n_words > 0):

        feature_vec = np.divide(feature_vec, n_words)

    return feature_vec
```

4.3 Calculating product similarity

Using GENSIM and word2vec, each product is represented as a vector. When two products are to be compared, we use this vectors to calculate similarity. The product vector is created by passing a string which consists of product characteristics, size, color, mrp, etc. When two products are similar/equal their parameters won't wary much. Thus giving us two vectors which are close to each other and hence, values nearer to 1 when tried to calculate similarity. Below is the code to create vector from product parameters.

```
> p1str = (title1+" "+mrp1+" "+str(sellingPrice1)+" "+str(specialPrice1)+"
"+productUrl1+" "+cate1+" "+productBrand1+" "+str(discount1)+"
"+str(shippingCharges1)+" "+sleeve1+" "+neck1+" "+fabric1+" "+printPattern1+"
"+size1+" "+topType1) # this is strings of product1 with paramters

> p1StrV = avg_feature_vector(p1str, model=model, num_features=300,
index2word_set=index2word_set)  # this is product1 vector, similar thing is
#done to product2

>sim = 1 - spatial.distance.cosine(p1StrV, p2StrV)
```

3.4 Exporting to JSON file

At the end the product Ids and the respective products ids which are similar/equal are exported to a JSON file with json package in python. Below is the sample code

```
Import json

Data = [[“1”,”2”],[“3”,”4”,”5”]]

with open('output.json', 'w') as fp:
    json.dump(Data, fp,sort_keys=True, indent=5)
```

5 Observation and results

5.1 Result

- The resultant JSON file is of 57Mb in size and has 8k+ product ids and their corresponding duplicates with similarity value.
- Most of the duplicate data is generated by varying parameters like size, prices etc,.

5.2 Observations

- Proper filtration methods can substantially reduce data size that we need to analyze
- There are hidden error present in large data sets and hence they need to be checked thoroughly
- Blank columns doesn't mean they are empty, there could be a space value stored in it
- Pre-trained models(genism) help us a lot and reduces the burden of custom training
- Processing large data needs more processing power and more main memory
- Understanding of data is the primary thing to do before operating of large data sets.

With more processing power and computational capacity we can further improve this by comparing the similarity of images given in product description. Distributed computing can fasten the process when applied on a large cluster. The data may still need more filtering and cleaning but due to time constraints I'm forced to skip them.

6 References

[1] Word to vector methods

https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[2] Sentence similarity

https://stackoverflow.com/questions/22129943/how-to-calculate-the-sentence-similarity-using-word2vec-model-of-gensim-with-pyt

[3] Recommendation engines

https://www.analyticsvidhya.com/blog/2016/06/quick-guide-build-recommendation-engine-python/

https://www.analyticsvidhya.com/blog/2015/10/recommendation-engines/

[4] Gensim

https://radimrehurek.com/gensim/tutorial.html