

## ▼ Amazon Apparel Recommendations

```
#import all the necessary packages.

from PIL import Image
import requests
from io import BytesIO
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import warnings
from bs4 import BeautifulSoup
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk
import math
import time
import re
import os
import seaborn as sns
from collections import Counter
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import pairwise_distances
from matplotlib import gridspec
from scipy.sparse import hstack
import plotly
import plotly.figure_factory as ff
from plotly.graph_objs import Scatter, Layout

plotly.offline.init_notebook_mode(connected=True)
warnings.filterwarnings("ignore")
```



## ▼ [4.3] Overview of the data

```
# we have give a json file which consists of all information about
# the products
# loading the data using pandas' read_json file.
data = pd.read_json('tops_fashion.json')
```

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



Number of data points : 183138 Number of features/variables: 19

## ▼ Terminology:

What is a dataset?

Rows and columns

Data-point

Feature/variable

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



```
Index(['asin', 'author', 'availability', 'availability_type', 'brand', 'color',
       'editorial_review', '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 belongs to ) 3. color ( Color information of apparel, it can contain many colors as a value ex: red and black strip SHIRT/TSHIRT ) 5. medium\_image\_url ( url of the image ) 6. title (title of the product.) 7. formatted\_price (price

```
data = data[['asin', 'brand', 'color', 'medium_image_url', 'product_type_name', 'title', 'format
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

	asin	brand	color	medium_image_url
0	B016I2TS4W	FNC7C	None	https://images-na.ssl-images-amazon.com/ima...
1	B01N49AI08	FIG Clothing	None	https://images-na.ssl-images-amazon.com/ima...
2	B01JDPCOHO	FIG Clothing	None	https://images-na.ssl-images-amazon.com/ima...
3	B01N19U5H5	Focal18	None	https://images-na.ssl-images-amazon.com/ima...
4	B004GSI2OS	FeatherLite	Onyx Black/ Stone	https://images-na.ssl-images-amazon.com/ima...

## ▼ [5.1] Missing data for various features

```
# We have total 72 unique type of product_type_names
print(data['product_type_name'].describe())
```

# 91.62% (167794/183138) of the products are shirts,



```
count    183138
unique     72
top        SHIRT
freq     167794
Name: product_type_name, dtype: object
```

```
# names of different product types
print(data['product_type_name'].unique())
```



```
['SHIRT' 'SWEATER' 'APPAREL' 'OUTDOOR_RECREATION_PRODUCT'
 'BOOKS_1973_AND_LATER' 'PANTS' 'HAT' 'SPORTING_GOODS' 'DRESS' 'UNDERWEAR'
 'SKIRT' 'OUTERWEAR' 'BRA' 'ACCESSORY' 'ART_SUPPLIES' 'SLEEPWEAR'
 'ORCA_SHIRT' 'HANDBAG' 'PET_SUPPLIES' 'SHOES' 'KITCHEN' 'ADULT_COSTUME'
 'HOME_BED_AND_BATH' 'MISC_OTHER' 'BLAZER' 'HEALTH_PERSONAL_CARE'
 'TOYS_AND_GAMES' 'SWIMWEAR' 'CONSUMER_ELECTRONICS' 'SHORTS' 'HOME'
 'AUTO_PART' 'OFFICE_PRODUCTS' 'ETHNIC_WEAR' 'BEAUTY'
 'INSTRUMENT_PARTS_AND_ACCESSORIES' 'POWERSPORTS_PROTECTIVE_GEAR' 'SHIRTS'
 'ABIS_APPAREL' 'AUTO_ACCESSORY' 'NONAPPARELMISC' 'TOOLS' 'BABY_PRODUCT'
 'SOCKSHOSIERY' 'POWERSPORTS RIDING SHIRT' 'EYEWEAR' 'SUIT'
 'OUTDOOR_LIVING' 'POWERSPORTS RIDING JACKET' 'HARDWARE' 'SAFETY_SUPPLY'
 'ABIS_DVD' 'VIDEO_DVD' 'GOLF_CLUB' 'MUSIC_POPULAR_VINYL'
 'HOME_FURNITURE_AND_DECOR' 'TABLET_COMPUTER' 'GUILD_ACCESSORIES'
 'ABIS_SPORTS' 'ART_AND_CRAFT_SUPPLY' 'BAG' 'MECHANICAL_COMPONENTS'
 'SOUND_AND_RECORDING_EQUIPMENT' 'COMPUTER_COMPONENT' 'JEWELRY'
 'BUILDING_MATERIAL' 'LUGGAGE' 'BABY_COSTUME' 'POWERSPORTS_VEHICLE_PART'
 'PROFESSIONAL_HEALTHCARE' 'SEEDS_AND_PLANTS' 'WIRELESS_ACCESSORY']
```

```
# find the 10 most frequent product_type_names.
product_type_count = Counter(list(data['product_type_name']))
product_type_count.most_common(10)
```



```
[('SHIRT', 167794),
 ('APPAREL', 3549),
 ('BOOKS_1973_AND_LATER', 3336),
 ('DRESS', 1584),
 ('SPORTING_GOODS', 1281),
 ('SWEATER', 837),
 ('OUTERWEAR', 796),
 ('OUTDOOR_RECREATION_PRODUCT', 729),
 ('ACCESSORY', 636),
 ('UNDERWEAR', 425)]
```

## ▼ Basic stats for the feature: brand

```
# there are 10577 unique brands
print(data['brand'].describe())

# 183138 - 182987 = 151 missing values.
```



count	182987
unique	10577
top	Zago
freq	223
Name:	brand, dtype: object

```
brand_count = Counter(list(data['brand']))
brand_count.most_common(10)
```



```
[('Zago', 223),
 ('XQS', 222),
 ('Yayun', 215),
 ('YUNY', 198),
 ('XiaoTianXin-women clothes', 193),
 ('Generic', 192),
 ('Boohoo', 190),
 ('Alion', 188),
 ('Abetteric', 187),
 ('TheMogan', 187)]
```

## ▼ Basic stats for the feature: color

```
print(data['color'].describe())
```

```
# we have 7380 unique colors
# 7.2% of products are black in color
# 64956 of 183138 products have brand information. That's approx 35.4%.
```



```
count      64956
unique     7380
top        Black
freq       13207
Name: color, dtype: object
```

```
color_count = Counter(list(data['color']))
color_count.most_common(10)
```



```
[(None, 118182),
 ('Black', 13207),
 ('White', 8616),
 ('Blue', 3570),
 ('Red', 2289),
 ('Pink', 1842),
 ('Grey', 1499),
 ('*', 1388),
 ('Green', 1258),
 ('Multi', 1203)]
```

## ▼ Basic stats for the feature: formatted\_price

```
print(data['formatted_price'].describe())
```

```
# Only 28,395 (15.5% of whole data) products with price information
```



```

count      28395
unique     3135
top       $19.99
freq       945
Name: formatted_price, dtype: object

price_count = Counter(list(data['formatted_price']))
price_count.most_common(10)

```

👤 [(None, 154743),  
('19.99', 945),  
('\$9.99', 749),  
('\$9.50', 601),  
('\$14.99', 472),  
('\$7.50', 463),  
('\$24.99', 414),  
('\$29.99', 370),  
('\$8.99', 343),  
('\$9.01', 336)]

## ▼ Basic stats for the feature: title

```

print(data['title'].describe())

# All of the products have a title.
# Titles are fairly descriptive of what the product is.
# We use titles extensively in this workshop
# as they are short and informative.

```

👤 count 183138  
unique 175985  
top Nakoda Cotton Self Print Straight Kurti For Women  
freq 77  
Name: title, dtype: object

```
data.to_pickle('pickels/180k_apparel_data').
```

We save data files at every major step in our processing in "pickle" files. If you are stuck anywhere (or) if some code is not working, use the pickle files we give you to speed things up.

```

# 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

```

# consider products which have color information
# data['color'].isnull() => gives the information about the dataframe row's which have null values color == 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 their laptops  
For those of you who have powerful computers and some time to spare, you are recommended to use all of the

```
data.to_pickle('pickels/28k_apparel_data')
```

```
# You can download all these 28k images using this code below.  
# You do NOT need to run this code and hence it is commented.
```

...

```
from PIL import Image
import requests
from io import BytesIO

for index, row in images.iterrows():
    url = row['large_image_url']
    response = requests.get(url)
    img = Image.open(BytesIO(response.content))
    img.save('images/28k_images/'+row['asin']+'.jpeg')
```

...



```
"\nfrom PIL import Image\nimport requests\nfrom io import BytesIO\n\nfor index, row i
```

### ▼ [5.2] Remove near duplicate items

#### ▼ [5.2.1] Understand about duplicates

```
# read data from pickle file from previous stage
data = pd.read_pickle('pickels/28k_apparel_data')

# find number of products that have duplicate titles.
print(sum(data.duplicated('title')))

# we have 2325 products which have same title but different color
```



2325

#### ▼ [5.2.2] Remove duplicates : Part 1

```
# read data from pickle file from previous stage
data = pd.read_pickle('pickels/28k_apparel_data')

data.head()
```



	asin	brand	color	
4	B004GS12OS	FeatherLite	Onyx Black/ Stone	https://images-na.ssl-images-
6	B012YX2ZPI	HX-Kingdom Fashion T-shirts	White	https://images-na.ssl-images-
11	B001LOUGE4	Fitness Etc.	Black	https://images-na.ssl-images-
15	B003BSRPB0	FeatherLite	White	https://images-na.ssl-images-
24	B014ICEDNA	ENCTC	Brown	https://images-na.ssl-images-

```
# Remove All products with very few words in title
data_sorted = data[data['title'].apply(lambda x: len(x.split())>4)]
print("After removal of products with short description:", data_sorted.shape[0])
```



After removal of products with short description: 27949

```
# Sort the whole data based on title (alphabetical order of title)
data_sorted.sort_values('title', inplace=True, ascending=False)
data_sorted.head()
```



	asin	brand	color	medium_image_u
61973	B06Y1KZ2WB	Éclair	Black/Pink	https://images-na.ssl-images-amazon.com/images
133820	B010RV33VE	xiaoming	Pink	https://images-na.ssl-images-amazon.com/images
81461	B01DDSDLNS	xiaoming	White	https://images-na.ssl-images-amazon.com/images
75995	B00X5LYO9Y	xiaoming	Red Anchors	https://images-na.ssl-images-amazon.com/images
151570	B00WPJG35K	xiaoming	White	https://images-na.ssl-images-amazon.com/images

## ▼ Some examples of duplicate titles that differ only in the last few words

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 Hea
- 62. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Hea
- 63. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Hea
- 64. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Print Hea

```

indices = []
for i, row in data_sorted.iterrows():
    indices.append(i)

import itertools
stage1_dedupe_asins = []
i = 0
j = 0
num_data_points = data_sorted.shape[0]
while i < num_data_points and j < num_data_points:

    previous_i = i

    # store the list of words of ith string in a, ex: a = ['tokidoki', 'The', 'Queen', 'of', 'Di
    a = data['title'].loc[indices[i]].split()

    # search for the similar products sequentially
    j = i+1
    while j < num_data_points:

        # store the list of words of jth string in b, ex: b = ['tokidoki', 'The', 'Queen', 'of',
        b = data['title'].loc[indices[j]].split()

        # store the maximum length of two strings
        length = max(len(a), len(b))

        # count is used to store the number of words that are matched in both strings
        count = 0

        # itertools.zip_longest(a,b): will map the corresponding words in both strings, it will
        # example: a = ['a', 'b', 'c', 'd']
        # b = ['a', 'b', 'd']
        # itertools.zip_longest(a,b): will give [('a', 'a'), ('b', 'b'), ('c', 'd'), ('d', None)]
        for k in itertools.zip_longest(a, b):
            if (k[0] == k[1]):
                count += 1

        # if the number of words in which both strings differ are > 2 , we are considering it as
        # if the number of words in which both strings differ are < 2 , we are considering it as
        if (length - count) > 2: # number of words in which both sentences differ
            # if both strings differ by more than 2 words we include the 1st string index
            stage1_dedupe_asins.append(data_sorted['asin'].loc[indices[i]])

        # if the comparison between is between num_data_points, num_data_points-1 strings a
        if j == num_data_points-1: stage1_dedupe_asins.append(data_sorted['asin'].loc[indices[j]])

        # start searching for similar appearances corresponds 2nd string
        i = j
        break
    else:
        j += 1
if previous_i == i:
    break

```

```
data = data.loc[data['asin'].isin(stage1_dedupe_asins)]
```

- ▼ We removed the duplicates which differ only at the end.

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

 Number of data points : 17593

```
data.to_pickle('pickels/17k_apperial_data')
```

### ▼ [5.2.3] Remove duplicates : Part 2

In the previous cell, we sorted whole data in alphabetical order of titles. Then, we r very similar title

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

Examples:

Titles-1

86261. UltraClub Women's Classic Wrinkle-Free Long Sleeve Oxford Shirt, Pink, XX-Large  
115042. UltraClub Ladies Classic Wrinkle-Free Long-Sleeve Oxford Light Blue XXL

TITles-2

75004. EVALY Women's Cool University Of UTAH 3/4 Sleeve Raglan Tee  
109225. EVALY Women's Unique University Of UTAH 3/4 Sleeve Raglan Tees  
120832. EVALY Women's New University Of UTAH 3/4-Sleeve Raglan Tshirt

```
data = pd.read_pickle('pickels/17k_apperial_data')
```

```
# This code snippet takes significant amount of time.
# O(n^2) time.
# Takes about an hour to run on a decent computer.
```

```
indices = []
for i, row in data.iterrows():
    indices.append(i)

stage2_dedupe_asins = []
while len(indices)!=0:
    i = indices.pop()
    stage2_dedupe_asins.append(data['asin'].loc[i])
    # consider the first apparel's title
    a = data['title'].loc[i].split()
    # store the list of words of ith string in a, ex: a = ['tokidoki', 'The', 'Queen', 'of', 'Di
    for j in indices:

        b = data['title'].loc[j].split()
        # store the list of words of jth string in b, ex: b = ['tokidoki', 'The', 'Queen', 'of',
        length = max(len(a),len(b))

        # count is used to store the number of words that are matched in both strings
        count = 0

        # itertools.zip_longest(a,b): will map the corresponding words in both strings, it will
        # example: a = ['a', 'b', 'c', 'd']
        # b = ['a', 'b', 'd']
        # itertools.zip_longest(a,b): will give [('a','a'), ('b','b'), ('c','d'), ('d', None)]
```

```

for k in itertools.zip_longest(a,b):
    if (k[0]==k[1]):
        count += 1

# if the number of words in which both strings differ are < 3 , we are considering it as
if (length - count) < 3:
    indices.remove(j)

# from whole previous products we will consider only
# the products that are found in previous cell
data = data.loc[data['asin'].isin(stage2_dedupe_asins)]


print('Number of data points after stage two of dedupe: ',data.shape[0])
# from 17k apperals we reduced to 16k apperals

```

 Number of data points after stage two of dedupe: 16435

```

data.to_pickle('pickels/16k_apperal_data')
# Storing these products in a pickle file
# candidates who wants to download these files instead
# of 180K they can download and use them from the Google Drive folder.

```

## ▼ 6. Text pre-processing

```

data = pd.read_pickle('pickels/16k_apperal_data')

# NLTK download stop words. [RUN ONLY ONCE]
# goto Terminal (Linux/Mac) or Command-Prompt (Window)
# In the temrinal, type these commands
# $python3
# $import nltk
# $nltk.download()

```

data.shape

 (16435, 7)

```

# we use the list of stop words that are downloaded from nltk lib.
stop_words = set(stopwords.words('english'))
print ('list of stop words:', stop_words)

def nlp_preprocessing(total_text, index, column):
    if type(total_text) is not int:
        string = ""
        for words in total_text.split():
            # remove the special chars in review like '"#$@!%^&*()_+-~?>< etc.
            word = ("").join(e for e in words if e.isalnum()))
            # Conver all letters to lower-case
            word = word.lower()
            # stop-word removal
            if not word in stop_words:
                string += word + " "
        data[column][index] = string

```



```
list of stop words: {'should', 'couldn', 'weren', 'o', 'didn', 'just', 'm', 'have',
```

```
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")
```

 7.976727174402514 seconds

```
data.head()
```

	asin	brand	color	
4	B004GSI2OS	FeatherLite	Onyx Black/ Stone	https://images-na.ssl-images
6	B012YX2ZPI	HX-Kingdom Fashion T-shirts	White	https://images-na.ssl-images
15	B003BSRPB0	FeatherLite	White	https://images-na.ssl-images
27	B014ICEJ1Q	FNC7C	Purple	https://images-na.ssl-images
46	B01NACPBG2	Fifth Degree	Black	https://images-na.ssl-images

```
data.to_pickle('pickels/16k_apperial_data_preprocessed')
```

## ▼ Stemming

```
from nltk.stem.porter import *
stemmer = PorterStemmer()
print(stemmer.stem('arguing'))
print(stemmer.stem('fishing'))
```

```
# We tried using stemming on our titles and it didnot work very well.
```

 argu  
fish

## ▼ [8] Text based product similarity

```
data = pd.read_pickle('pickels/16k_apperial_data_preprocessed')
data.head()
```



	asin	brand	color
4	B004GSI2OS	FeatherLite	Onyx Black/ Stone
6	B012YX2ZPI	HX-Kingdom Fashion T-shirts	White

data.shape

(16042, 7)

# Utility Functions which we will use through the rest of the workshop.

```
#Display an image
def display_img(url,ax,fig):
    # we get the url of the apparel and download it
    response = requests.get(url)
    img = Image.open(BytesIO(response.content))
    # we will display it in notebook
    plt.imshow(img)

#plotting code to understand the algorithm's decision.
def plot_heatmap(keys, values, labels, url, text):
    # keys: list of words of recommended title
    # values: len(values) == len(keys), values(i) represents the occurrence of the word keys
    # labels: len(labels) == len(keys), the values of labels depends on the model we are using
        # if model == 'bag of words': labels(i) = values(i)
        # if model == 'tfidf weighted bag of words': labels(i) = tfidf(keys(i))
        # if model == 'idf weighted bag of words': labels(i) = idf(keys(i))
    # url : apparel's url

    # we will devide the whole figure into two parts
    gs = gridspec.GridSpec(2, 2, width_ratios=[4,1], height_ratios=[4,1])
    fig = plt.figure(figsize=(25,3))

    # 1st, plotting heat map that represents the count of commonly occurred words in title2
    ax = plt.subplot(gs[0])
    # it displays a cell in white color if the word is intersection(list of words of title1 and title2)
    ax = sns.heatmap(np.array([values]), annot=np.array([labels]))
    ax.set_xticklabels(keys) # set that axis labels as the words of title
    ax.set_title(text) # apparel title

    # 2nd, plotting image of the apparel
    ax = plt.subplot(gs[1])
    # we don't want any grid lines for image and no labels on x-axis and y-axis
    ax.grid(False)
    ax.set_xticks([])
    ax.set_yticks([])

    # we call dispaly_img based with paramete url
    display_img(url, ax, fig)

    # displays combine figure ( heat map and image together)
    plt.show()

def plot_heatmap_image(doc_id, vec1, vec2, url, text, model):
    # doc_id : index of the title1
    # vec1 : input apparel's vector, it is of a dict type {word:count}
    # vec2 : recommended apparel's vector, it is of a dict type {word:count}
    # url : apparel's image url
    # text: title of recomended apparel (used to keep title of image)
    # model, it can be any of the models,
        # 1. bag_of_words
        # 2. tfidf
```

```

# 3. idf

# we find the common words in both titles, because these only words contribute to the distance
intersection = set(vec1.keys()) & set(vec2.keys())

# we set the values of non intersecting words to zero, this is just to show the difference in
for i in vec2:
    if i not in intersection:
        vec2[i]=0

# for labeling heatmap, keys contains list of all words in title2
keys = list(vec2.keys())
# if ith word in intersection(list of words of title1 and list of words of title2): values(i)
values = [vec2[x] for x in vec2.keys()]

# labels: len(labels) == len(keys), the values of labels depends on the model we are using
# if model == 'bag of words': labels(i) = values(i)
# if model == 'tfidf weighted bag of words':labels(i) = tfidf(keys(i))
# if model == 'idf weighted bag of words':labels(i) = idf(keys(i))

if model == 'bag_of_words':
    labels = values
elif model == 'tfidf':
    labels = []
    for x in vec2.keys():
        # tfidf_title_vectorizer.vocabulary_ it contains all the words in the corpus
        # tfidf_title_features[doc_id, index_of_word_in_corpus] will give the tfidf value of word
        if x in tfidf_title_vectorizer.vocabulary_:
            labels.append(tfidf_title_features[doc_id, tfidf_title_vectorizer.vocabulary_[x]])
        else:
            labels.append(0)
elif model == 'idf':
    labels = []
    for x in vec2.keys():
        # idf_title_vectorizer.vocabulary_ it contains all the words in the corpus
        # idf_title_features[doc_id, index_of_word_in_corpus] will give the idf value of word
        if x in idf_title_vectorizer.vocabulary_:
            labels.append(idf_title_features[doc_id, idf_title_vectorizer.vocabulary_[x]])
        else:
            labels.append(0)

plot_heatmap(keys, values, labels, url, text)

# this function gets a list of words along with the frequency of each
# word given "text"
def text_to_vector(text):
    word = re.compile(r'\w+')
    words = word.findall(text)
    # words stores list of all words in given string, you can try 'words = text.split()' this will
    return Counter(words) # Counter counts the occurrence of each word in list, it returns dict t

def get_result(doc_id, content_a, content_b, url, model):
    text1 = content_a
    text2 = content_b

    # vector1 = dict{word11:#count, word12:#count, etc.}
    vector1 = text_to_vector(text1)

    # vector2 = dict{word21:#count, word22:#count, etc.}
    vector2 = text_to_vector(text2)

    plot_heatmap_image(doc_id, vector1, vector2, url, text2, model)

```

## ▼ [8.2] Bag of Words (BoW) on product titles.

```

from sklearn.feature_extraction.text import CountVectorizer
title_vectorizer = CountVectorizer()
title_features = title_vectorizer.fit_transform(data['title'])
title_features.get_shape() # get number of rows and columns in feature matrix.
# title_features.shape = #data_points * #words_in_corpus
# CountVectorizer().fit_transform(corpus) returns
# the a sparse matrix of dimensions #data_points * #words_in_corpus

# What is a sparse vector?

# title_features[doc_id, index_of_word_in_corpus] = number of times the word occurred in that doc

```



(16042, 12609)

```

def bag_of_words_model(doc_id, num_results):
    # doc_id: apparel's id in given corpus

    # pairwise_dist will store the distance from given input apparel to all remaining apparels
    # the metric we used here is cosine, the coside distance is measured as K(X, Y) = <X, Y> / (|X| * |Y|)
    # http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
    pairwise_dist = pairwise_distances(title_features,title_features[doc_id])

    # np.argsort will return indices of the smallest distances
    indices = np.argsort(pairwise_dist.flatten())[0:num_results]
    #pdists will store the smallest distances
    pdists = np.sort(pairwise_dist.flatten())[0:num_results]

    #data frame indices of the 9 smallest distace's
    df_indices = list(data.index[indices])

    for i in range(0,len(indices)):
        # we will pass 1. doc_id, 2. title1, 3. title2, url, model
        get_result(indices[i],data['title'].loc[df_indices[0]], data['title'].loc[df_indices[i]])
        print('ASIN :',data['asin'].loc[df_indices[i]])
        print ('Brand:', data['brand'].loc[df_indices[i]])
        print ('Title:', data['title'].loc[df_indices[i]])
        print ('Euclidean similarity with the query image :', pdists[i])
        print('='*60)

#call the bag-of-words model for a product to get similar products.
bag_of_words_model(12566, 20) # change the index if you want to.
# In the output heat map each value represents the count value
# of the label word, the color represents the intersection
# with inputs title.

#try 12566
#try 931

```



burnt umber tiger tshirt zebra stripes xl xxl



ASIN : B00JXQB5FQ

Brand: Si Row

Title: burnt umber tiger tshirt zebra stripes xl xxl

Euclidean similarity with the query image : 0.0

---

pink tiger tshirt zebra stripes xl xxl



ASIN : B00JXQASS6

Brand: Si Row

Title: pink tiger tshirt zebra stripes xl xxl

Euclidean similarity with the query image : 1.7320508075688772

---

brown white tiger tshirt tiger stripes xl xxl



ASIN : B00JXQCWT0

Brand: Si Row

Title: brown white tiger tshirt tiger stripes xl xxl

Euclidean similarity with the query image : 2.449489742783178

---

yellow tiger tshirt tiger stripes l



ASIN : B00JXQCUIC

Brand: Si Row

Title: yellow tiger tshirt tiger stripes l

Euclidean similarity with the query image : 2.6457513110645907

---



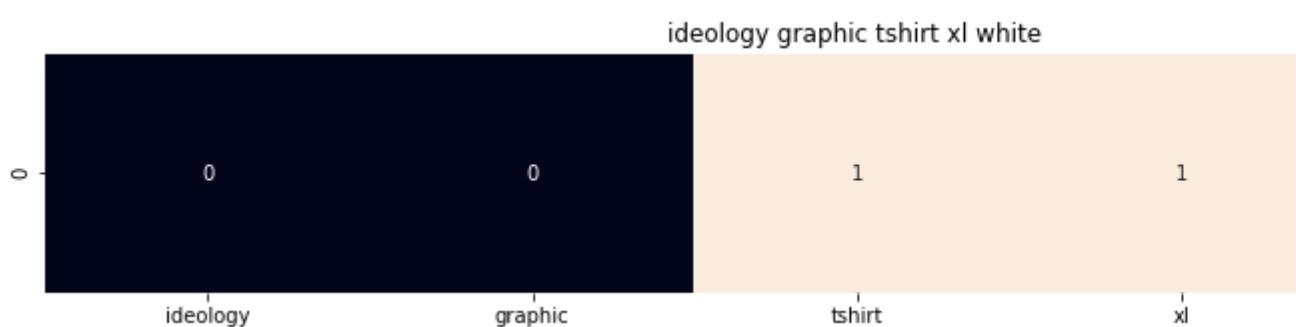
ASIN : B07568NZX4

Brand: Rustic Grace

Title: believed could tshirt

Euclidean similarity with the query image : 3.0

---



ASIN : B01NB0NKRO

Brand: Ideology

Title: ideology graphic tshirt xl white

Euclidean similarity with the query image : 3.0

---



ASIN : B00JXQAFZ2

Brand: Si Row

Title: grey white tiger tank top tiger stripes xl xxl

Euclidean similarity with the query image : 3.0

---



ASIN : B01CLS8LMW

Brand: Awake

Title: morning person tshirt troll picture xl  
 Euclidean similarity with the query image : 3.1622776601683795

---

merona green gold stripes



ASIN : B01KVZUB6G

Brand: Merona

Title: merona green gold stripes

Euclidean similarity with the query image : 3.1622776601683795

blvd womens graphic tshirt l



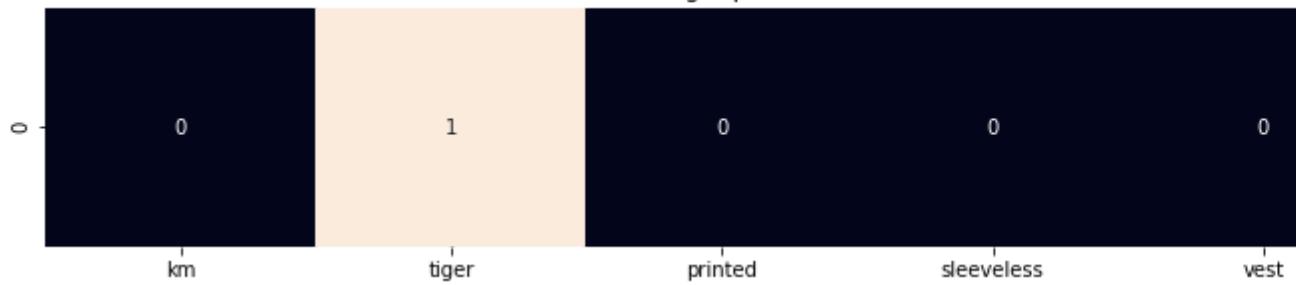
ASIN : B0733R2CJK

Brand: BLVD

Title: blvd womens graphic tshirt l

Euclidean similarity with the query image : 3.1622776601683795

km tiger printed sleeveless vest tshirt



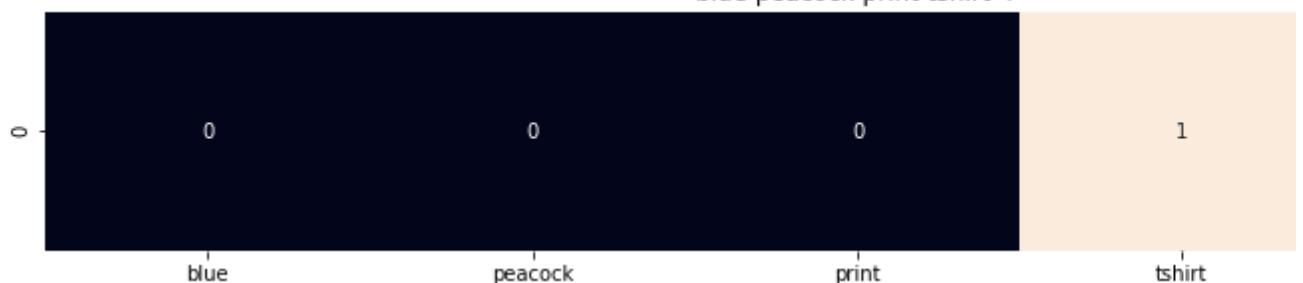
ASIN : B012VQLT6Y

Brand: KM T-shirt

Title: km tiger printed sleeveless vest tshirt

Euclidean similarity with the query image : 3.1622776601683795

blue peacock print tshirt l



ASIN : B000JXQL8L6

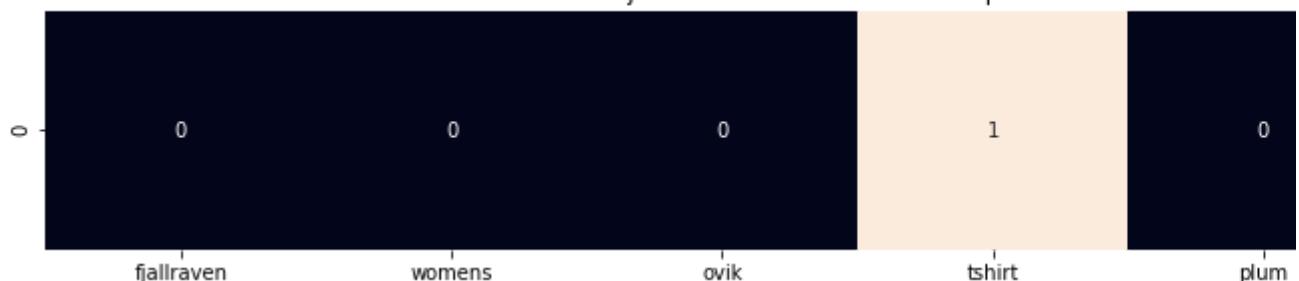
Brand: Si Row

Title: blue peacock print tshirt 1

Euclidean similarity with the query image : 3.1622776601683795

=====

fjallraven womens ovik tshirt plum xxl



ASIN : B06XC3CZF6

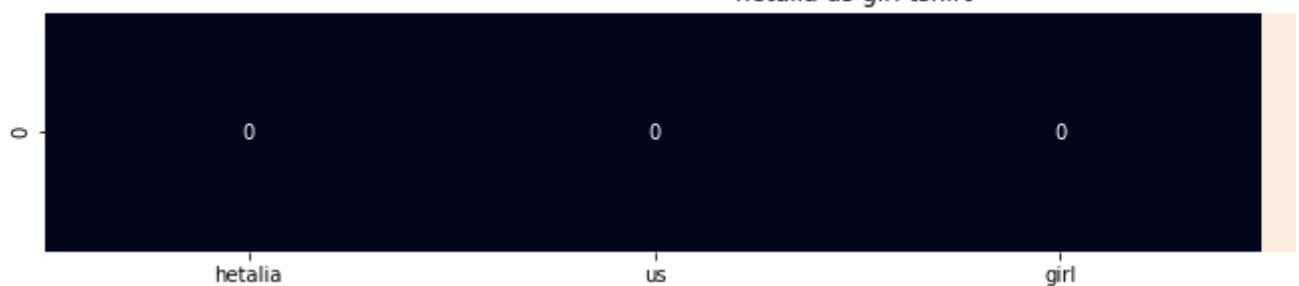
Brand: Fjallraven

Title: fjallraven womens ovik tshirt plum xxl

Euclidean similarity with the query image : 3.1622776601683795

=====

hetalia us girl tshirt



ASIN : B005IT80BA

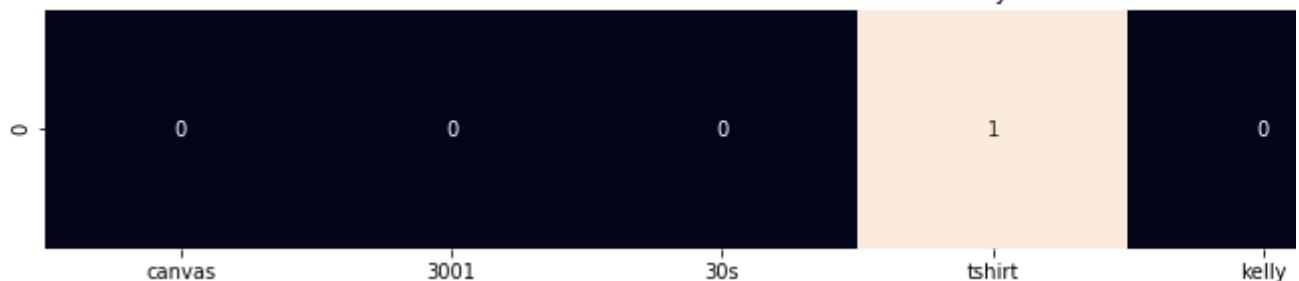
Brand: Hetalia

Title: hetalia us girl tshirt

Euclidean similarity with the query image : 3.1622776601683795

=====

canvas 3001 30s tshirt kelly xl



ASIN : B0088PN0LA

Brand: Red House

Title: canvas 3001 30s tshirt kelly xl

Euclidean similarity with the query image : 3.1622776601683795

=====

brunello cucinelli tshirt women white xl



brunello

cucinelli

tshirt

women

white

ASIN : B06X99V6WC

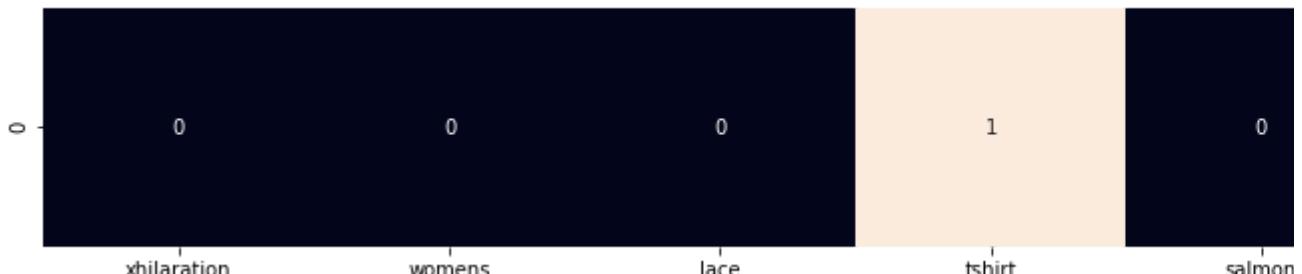
Brand: Brunello Cucinelli

Title: brunello cucinelli tshirt women white xl

Euclidean similarity with the query image : 3.1622776601683795

=====

xhilaration womens lace tshirt salmon xxl



ASIN : B06Y1JPW1Q

Brand: Xhilaration

Title: xhilaration womens lace tshirt salmon xxl

Euclidean similarity with the query image : 3.1622776601683795

=====

animal oceania tshirt yellow



ASIN : B06X6GX6WG

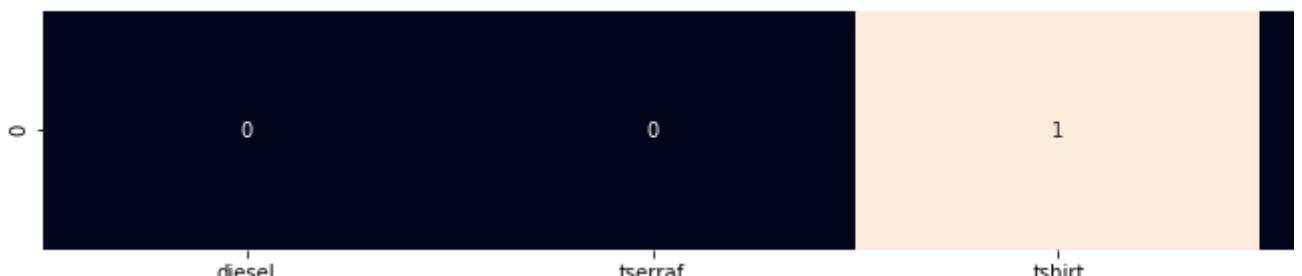
Brand: Animal

Title: animal oceania tshirt yellow

Euclidean similarity with the query image : 3.1622776601683795

=====

diesel tserraf tshirt black



ASIN : B017X8PW9U

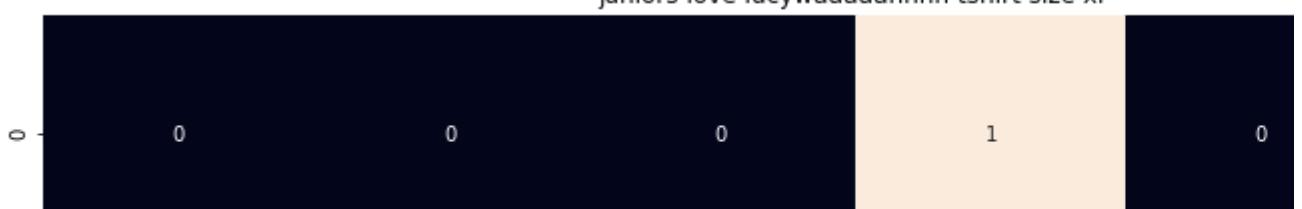
Brand: Diesel

Title: diesel tserraf tshirt black

Euclidean similarity with the query image : 3.1622776601683795

=====

juniors love lucywaaaaahhhh tshirt size xl



```

juniors          love          lucywaaaaahhhh      tshirt        size
ASIN : B00IAAA4JIQ
Brand: I Love Lucy
Title: juniors love lucywaaaaahhhh tshirt size xl
Euclidean similarity with the query image : 3.1622776601683795
=====

```

## ▼ [8.5] TF-IDF based product similarity

```

tfidf_title_vectorizer = TfidfVectorizer(min_df = 0)
tfidf_title_features = tfidf_title_vectorizer.fit_transform(data['title'])
# tfidf_title_features.shape = #data_points * #words_in_corpus
# CountVectorizer().fit_transform(courpus) returns the a sparase matrix of dimensions #data_poin
# tfidf_title_features[doc_id, index_of_word_in_corpus] = tfidf values of the word in given doc

def tfidf_model(doc_id, num_results):
    # doc_id: apparel's id in given corpus

    # pairwise_dist will store the distance from given input apparel to all remaining apparels
    # the metric we used here is cosine, the coside distance is mesured as K(X, Y) = <X, Y> / (|
    # http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
    pairwise_dist = pairwise_distances(tfidf_title_features,tfidf_title_features[doc_id])

    # np.argsort will return indices of 9 smallest distances
    indices = np.argsort(pairwise_dist.flatten())[0:num_results]
    #pdists will store the 9 smallest distances
    pdists = np.sort(pairwise_dist.flatten())[0:num_results]

    #data frame indices of the 9 smallest distace's
    df_indices = list(data.index[indices])

    for i in range(0,len(indices)):
        # we will pass 1. doc_id, 2. title1, 3. title2, url, model
        get_result(indices[i], data['title'].loc[df_indices[0]], data['title'].loc[df_indices[i]])
        print('ASIN :',data['asin'].loc[df_indices[i]])
        print('BRAND :',data['brand'].loc[df_indices[i]])
        print ('Eucliden distance from the given image :', pdists[i])
        print('='*125)
tfidf_model(12566, 20)
# in the output heat map each value represents the tfidf values of the label word, the color rep

```



burnt umber tiger tshirt zebra stripes xl xxl

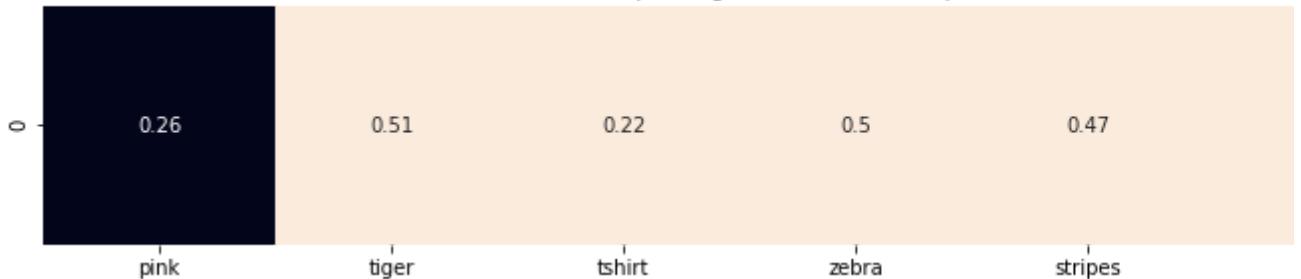


ASIN : B00JXQB5FQ

BRAND : Si Row

Eucliden distance from the given image : 0.0

pink tiger tshirt zebra stripes xl xxl

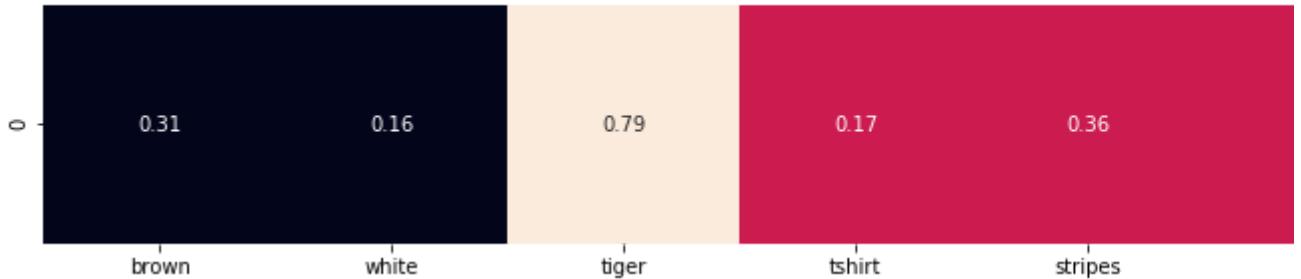


ASIN : B00JXQASS6

BRAND : Si Row

Eucliden distance from the given image : 0.7536331912451361

brown white tiger tshirt tiger stripes xl xxl

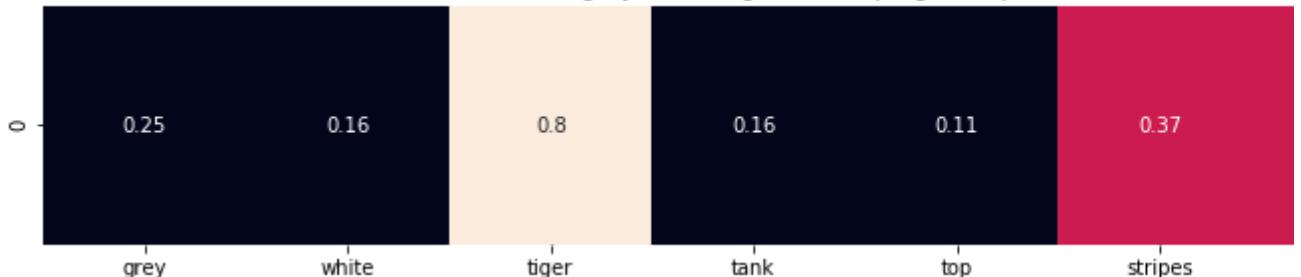


ASIN : B00JXQCWTO

BRAND : Si Row

Eucliden distance from the given image : 0.9357643943769645

grey white tiger tank top tiger stripes xl xxl



ASIN : B00JXQAFZ2

BRAND : Si Row

Eucliden distance from the given image : 0.9586153524200749

yellow tiger tshirt tiger stripes l

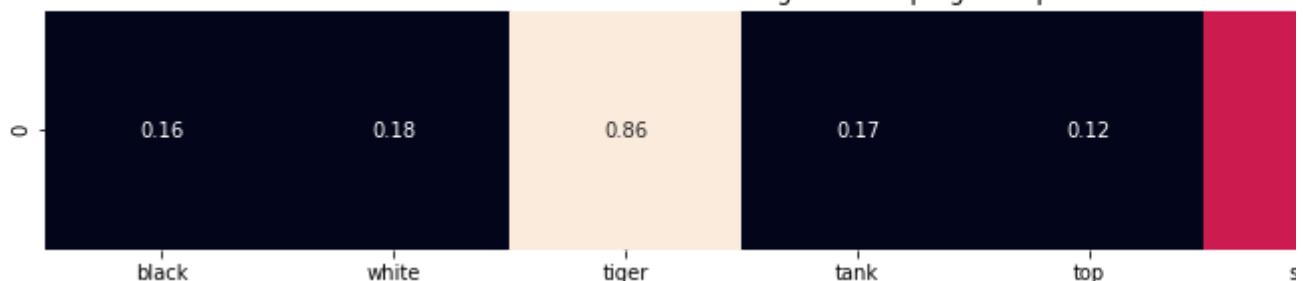


ASIN : B00JXQCUIC

BRAND : Si Row

Eucliden distance from the given image : 1.000074961446881

black white tiger tank top tiger stripes |



ASIN : B00JXQA094

BRAND : Si Row

Eucliden distance from the given image : 1.023215552457452

yellow tiger tank top tiger stripes |



ASIN : B00JXQUWA

BRAND : Si Row

Eucliden distance from the given image : 1.031991846303421

studio womens burnt orange dolman top size medium



ASIN : B06XSCVFT5

BRAND : Studio M

Eucliden distance from the given image : 1.2106843670424716

boundaries juniors racerback ribbed tank pink zebra xxl

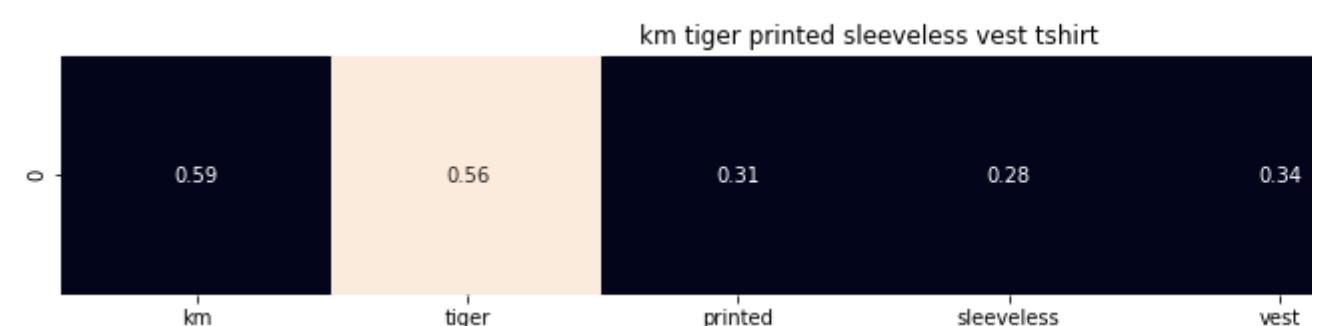




ASIN : B06Y2GTYPM

BRAND : No Boundaries

Euclidean distance from the given image : 1.212168381072083



ASIN : B012VQLT6Y

BRAND : KM T-shirt

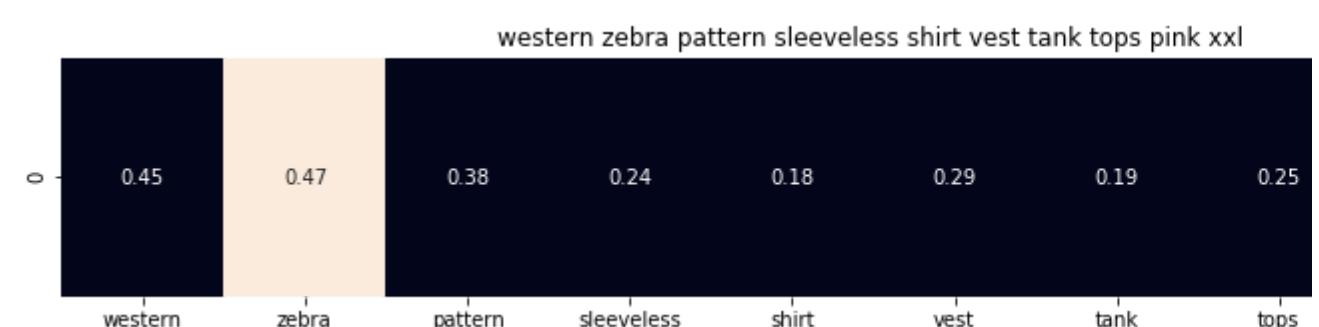
Euclidean distance from the given image : 1.219790640280982



ASIN : B06Y1VN8WQ

BRAND : Black Swan

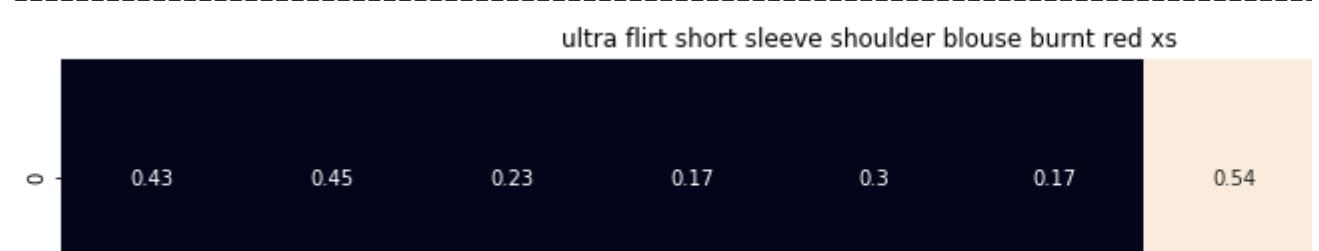
Euclidean distance from the given image : 1.2206849659998316



ASIN : B00Z6HEXWI

BRAND : Black Temptation

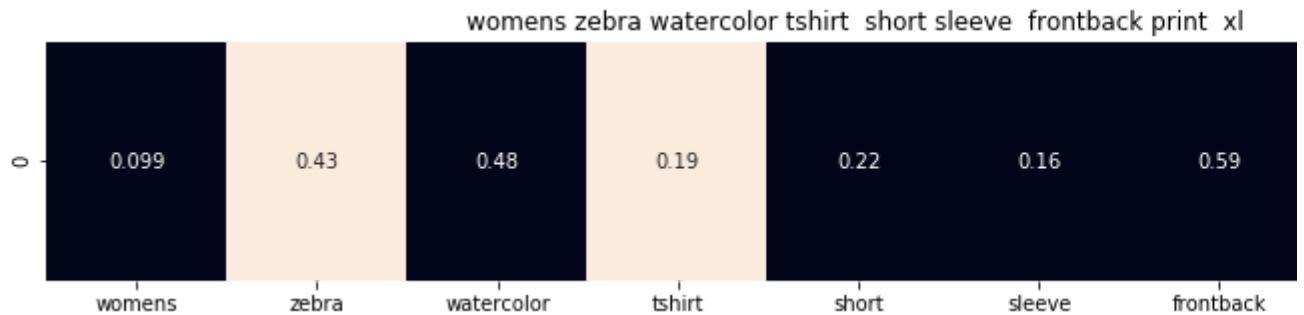
Euclidean distance from the given image : 1.221281392120943



ultra      flirt      short      sleeve      shoulder      blouse      burnt

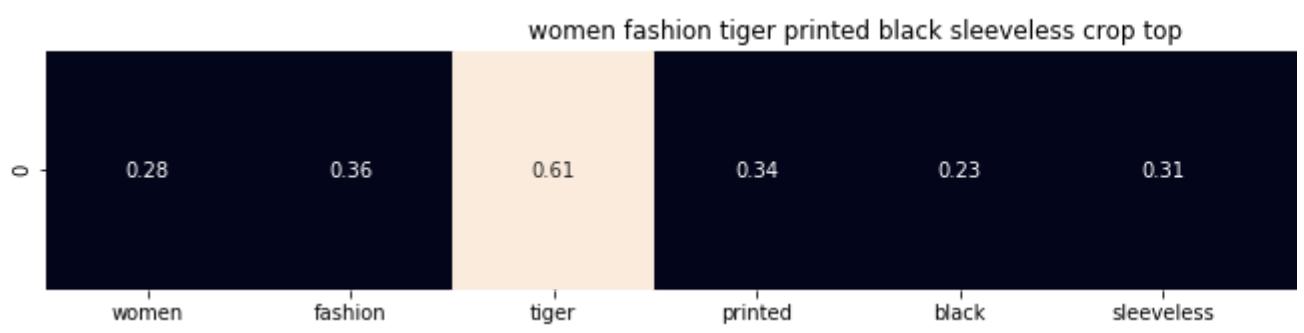
ASIN : B074TR12BH  
 BRAND : Ultra Flirt  
 Eucliden distance from the given image : 1.2313364094597743

---



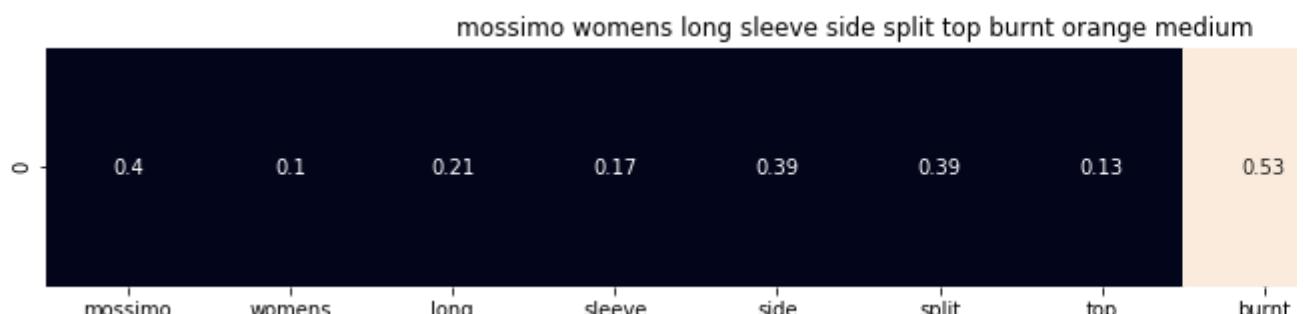
ASIN : B072R2JXKW  
 BRAND : WHAT ON EARTH  
 Eucliden distance from the given image : 1.2318451972624518

---



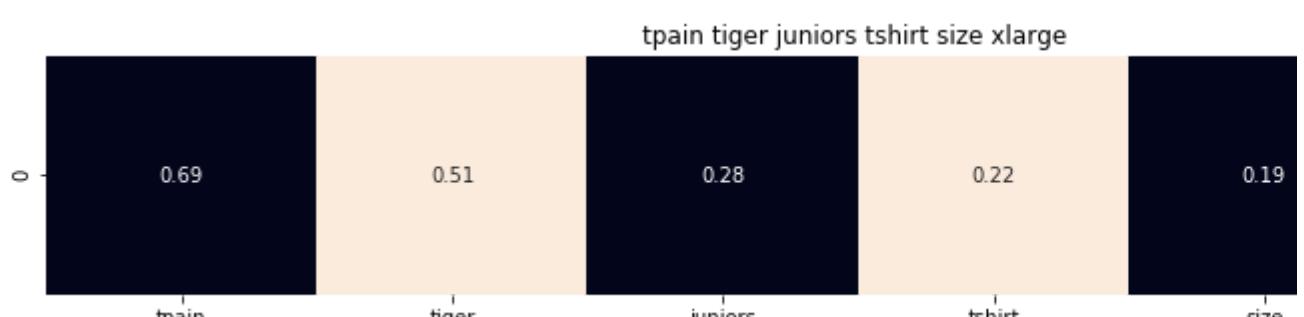
ASIN : B074T8ZYGX  
 BRAND : MKP Crop Top  
 Eucliden distance from the given image : 1.2340607457359425

---



ASIN : B071ZDF6T2  
 BRAND : Mossimo  
 Eucliden distance from the given image : 1.2352785577664824

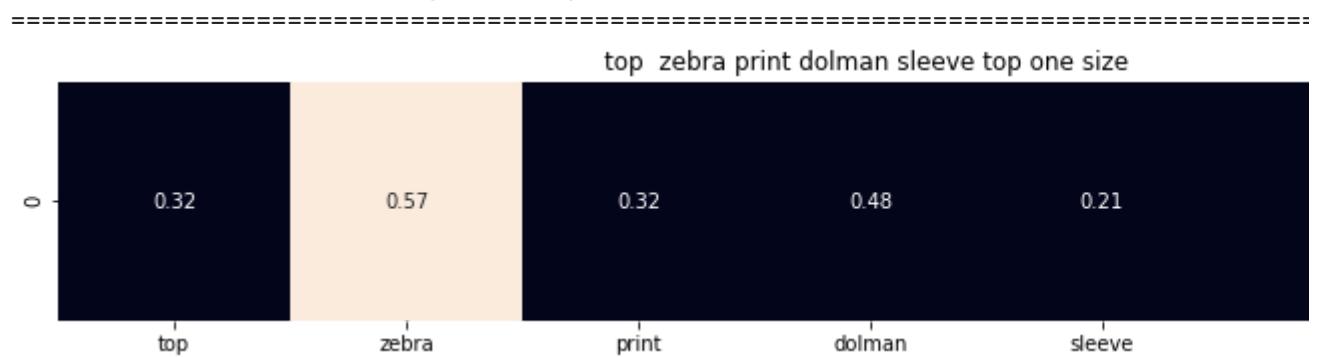
---



ASIN : B01K0H020G

BRAND : Tultex

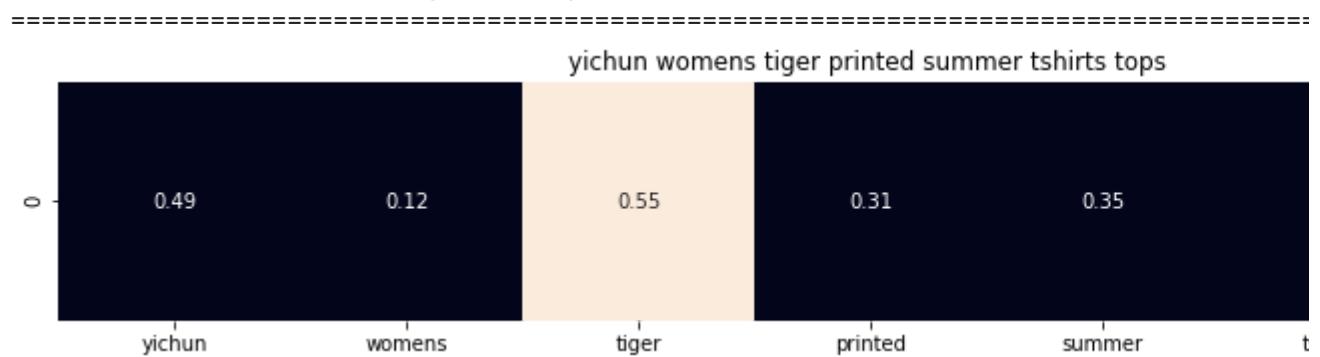
Eucliden distance from the given image : 1.236457298812782



ASIN : B00H8A6ZLI

BRAND : Vivian's Fashions

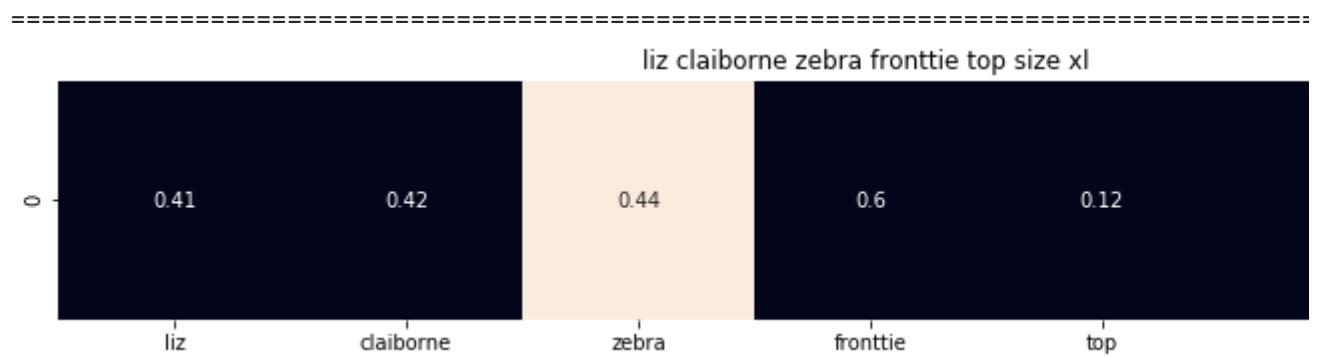
Eucliden distance from the given image : 1.24996155052848



ASIN : B010NN9RX0

BRAND : YICHUN

Eucliden distance from the given image : 1.25354614208561



ASIN : B06XY5QXL

BRAND : Liz Claiborne

Eucliden distance from the given image : 1.2538832938357722

## ▼ [8.5] IDF based product similarity

```
idf_title_vectorizer = CountVectorizer()
idf_title_features = idf_title_vectorizer.fit_transform(data['title'])

# idf_title_features.shape = #data_points * #words_in_corpus
# CountVectorizer().fit_transform(courpus) returns the a sparase matrix of dimensions #data_poin
# idf_title_features[doc_id, index_of_word_in_corpus] = number of times the word occured in that

def nContaining(word):
```

```

# return the number of documents which had the given word
return sum(1 for blob in data['title'] if word in blob.split())

def idf(word):
    # idf = log(#number of docs / #number of docs which had the given word)
    return math.log(data.shape[0] / (n_containing(word)))

# we need to convert the values into float
idf_title_features = idf_title_features.astype(np.float)

for i in idf_title_vectorizer.vocabulary_.keys():
    # for every word in whole corpus we will find its idf value
    idf_val = idf(i)

    # to calculate idf_title_features we need to replace the count values with the idf values of
    # idf_title_features[:, idf_title_vectorizer.vocabulary_[i]].nonzero()[0] will return all do
    for j in idf_title_features[:, idf_title_vectorizer.vocabulary_[i]].nonzero()[0]:
        # we replace the count values of word i in document j with idf_value of word
        # idf_title_features[doc_id, index_of_word_in_corpus] = idf value of word
        idf_title_features[j,idf_title_vectorizer.vocabulary_[i]] = idf_val


def idf_model(doc_id, num_results):
    # doc_id: apparel's id in given corpus

    # pairwise_dist will store the distance from given input apparel to all remaining apparels
    # the metric we used here is cosine, the coside distance is mesured as K(X, Y) = <X, Y> / (|X| |Y|)
    # http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
    pairwise_dist = pairwise_distances(idf_title_features,idf_title_features[doc_id])

    # np.argsort will return indices of 9 smallest distances
    indices = np.argsort(pairwise_dist.flatten())[0:num_results]
    #pdists will store the 9 smallest distances
    pdists = np.sort(pairwise_dist.flatten())[0:num_results]

    #data frame indices of the 9 smallest distace's
    df_indices = list(data.index[indices])

    for i in range(0,len(indices)):
        get_result(indices[i],data['title'].loc[df_indices[0]], data['title'].loc[df_indices[i]])
        print('ASIN :',data['asin'].loc[df_indices[i]])
        print('Brand :',data['brand'].loc[df_indices[i]])
        print ('euclidean distance from the given image :', pdists[i])
        print('='*125)

idf_model(12566,20)
# in the output heat map each value represents the idf values of the label word, the color repre

```



burnt umber tiger tshirt zebra stripes xl xxl

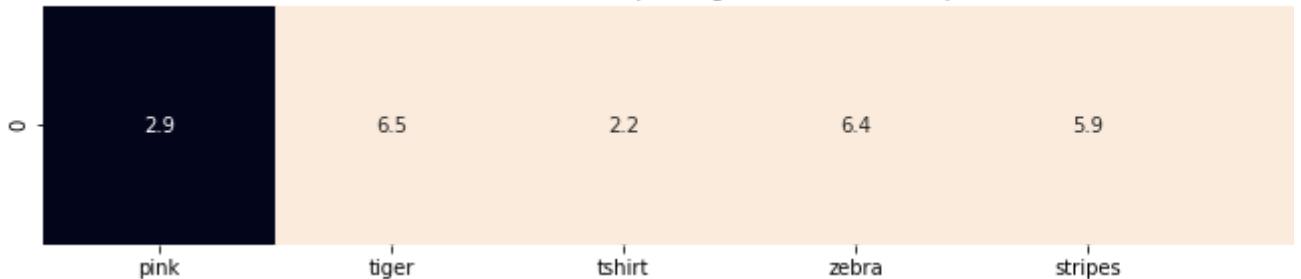


ASIN : B00JXQB5FQ

Brand : Si Row

euclidean distance from the given image : 0.0

pink tiger tshirt zebra stripes xl xxl

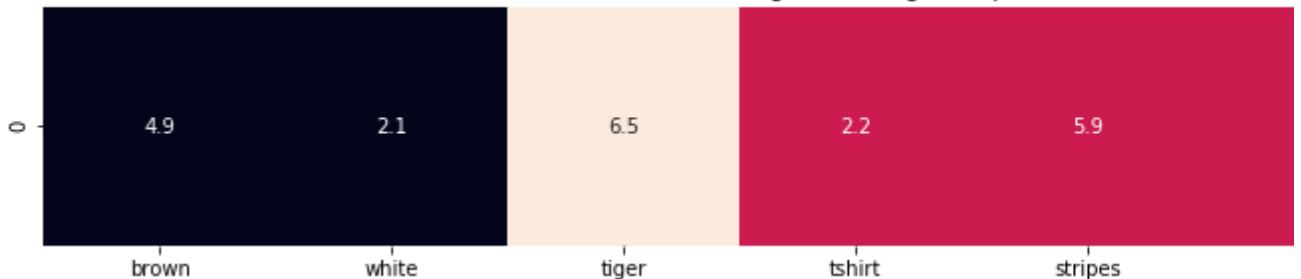


ASIN : B00JXQASS6

Brand : Si Row

euclidean distance from the given image : 12.20507131122177

brown white tiger tshirt tiger stripes xl xxl



ASIN : B00JXQCWTO

Brand : Si Row

euclidean distance from the given image : 14.468362685603465

grey white tiger tank top tiger stripes xl xxl

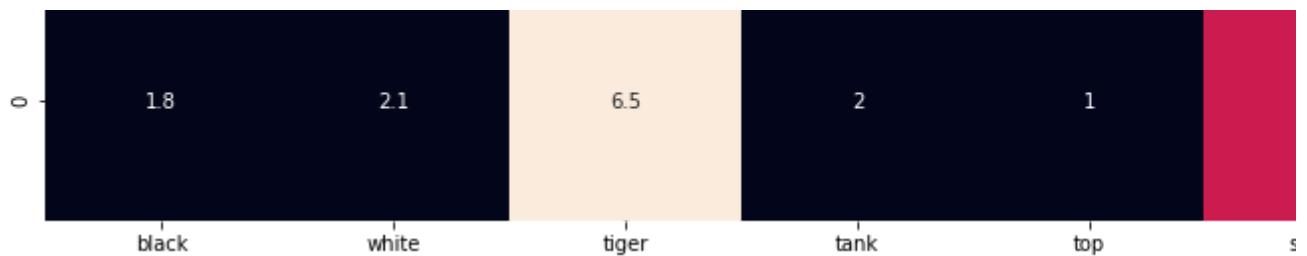


ASIN : B00JXQAFZ2

Brand : Si Row

euclidean distance from the given image : 14.486832924778964

black white tiger tank top tiger stripes l



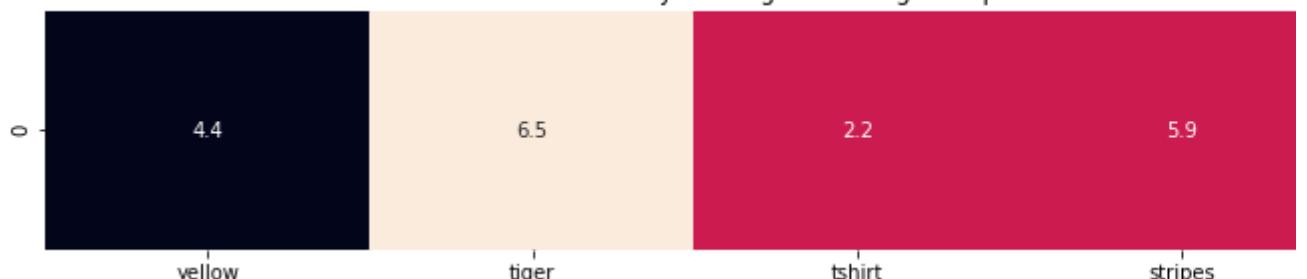
ASIN : B00JXQA094

Brand : Si Row

euclidean distance from the given image : 14.833392966672909

---

yellow tiger tshirt tiger stripes l



ASIN : B00JXQCUIC

Brand : Si Row

euclidean distance from the given image : 14.898744516719225

---

yellow tiger tank top tiger stripes l



ASIN : B00JXQUWA

Brand : Si Row

euclidean distance from the given image : 15.224458287343769

---

women fashion tiger printed black sleeveless crop top



ASIN : B074T8ZYGX

Brand : MKP Crop Top

euclidean distance from the given image : 17.080812955631995

---

long sleeve top blouse tshirt





ASIN : B00KF2N5PU

Brand : Vietsbay

euclidean distance from the given image : 17.090168125645416

---

womens tank top white



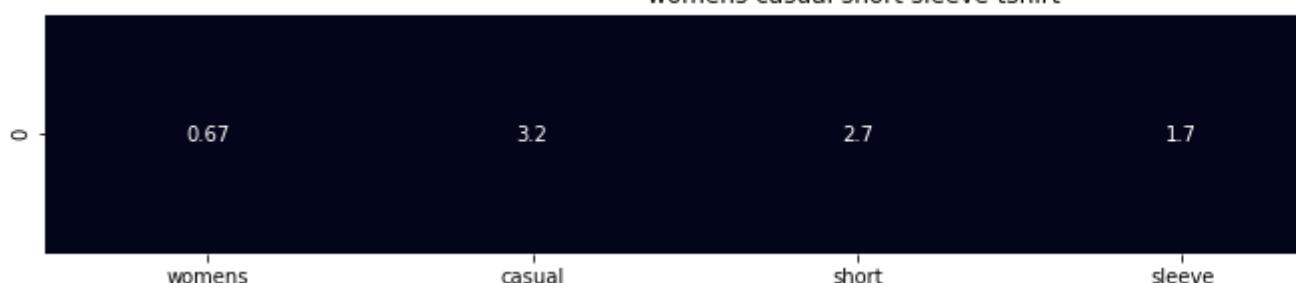
ASIN : B00JPOZ9GM

Brand : Sofra

euclidean distance from the given image : 17.153215337562703

---

womens casual short sleeve tshirt



ASIN : B074T9KG9Q

Brand : Rain

euclidean distance from the given image : 17.33671523874989

---

top zebra print dolman sleeve top one size



ASIN : B00H8A6ZLI

Brand : Vivian's Fashions

euclidean distance from the given image : 17.410075941001253

---

white top blouse tank shirt sleeveless



white top blouse tank shirt

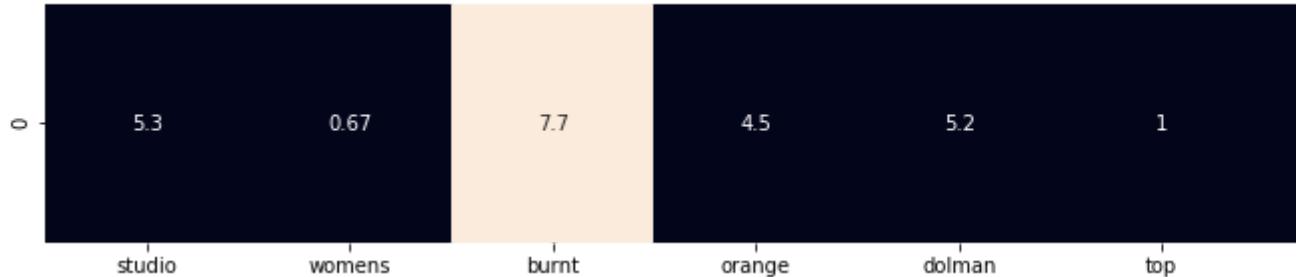
ASIN : B074G5G5RK

Brand : ERMANNO SCERVINO

euclidean distance from the given image : 17.539921335459557

---

===== studio womens burnt orange dolman top size medium =====



ASIN : B06XSCVFT5

Brand : Studio M

euclidean distance from the given image : 17.61275854366134

---

===== wear womens vneck blouse black xxl =====



ASIN : B06Y6FH453

Brand : Who What Wear

euclidean distance from the given image : 17.623745282500135

---

===== womens casual vneck short sleeve tshirt =====



ASIN : B074V45DCX

Brand : Rain

euclidean distance from the given image : 17.634342496835046

---

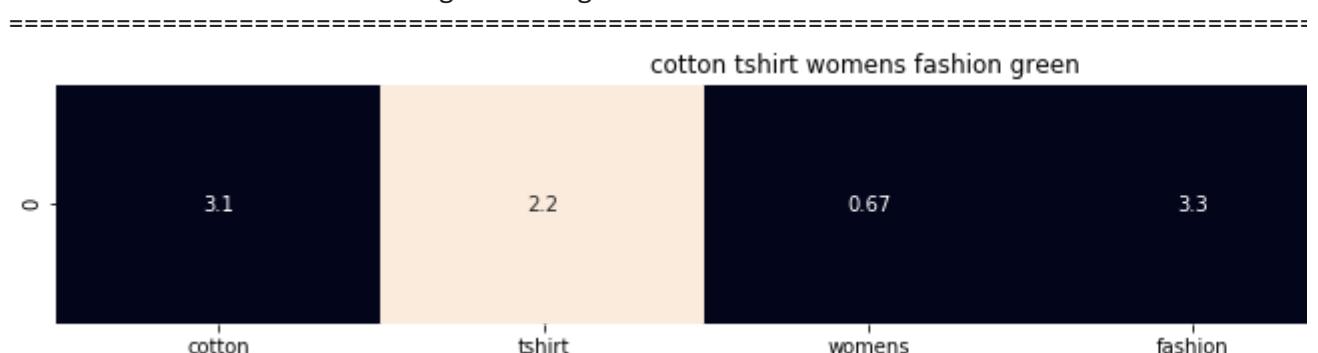
===== j womens shoulder blue small =====



ASIN : B07583CQFT

Brand : Very J

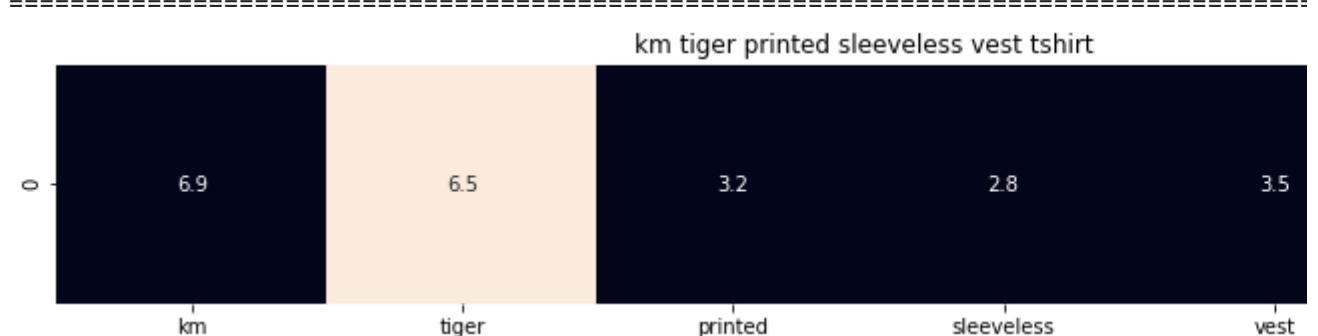
euclidean distance from the given image : 17.63753712743611



ASIN : B073GJGVBN

Brand : Ivan Levi

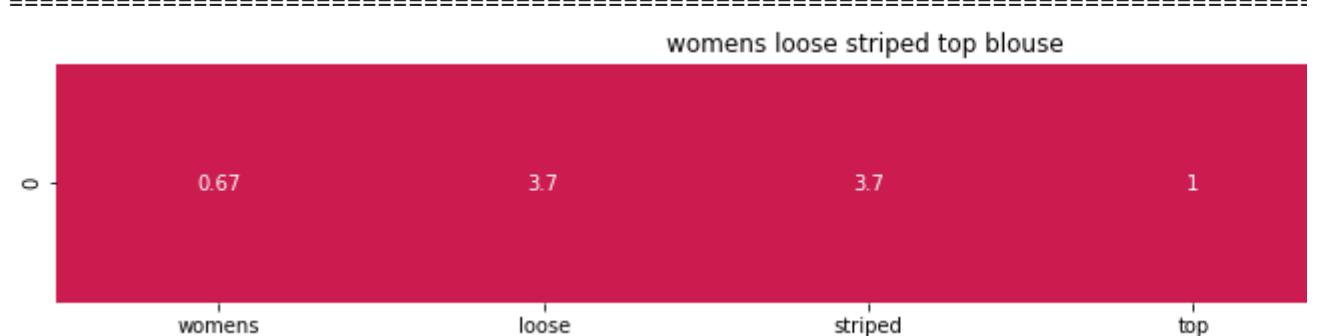
euclidean distance from the given image : 17.7230738913371



ASIN : B012VQLT6Y

Brand : KM T-shirt

euclidean distance from the given image : 17.762588561202364



ASIN : B00ZZMYBRG

Brand : HP-LEISURE

euclidean distance from the given image : 17.779536864674238

## ▼ [9] Text Semantics based product similarity

```
# credits: https://www.kaggle.com/c/word2vec-nlp-tutorial#part-2-word-vectors
# Custom Word2Vec using your own text data.
# Do NOT RUN this code.
# It is meant as a reference to build your own Word2Vec when you have
# lots of data.

...
# Set values for various parameters
num_features = 300      # Word vector dimensionality
```

```

min_word_count = 1      # Minimum word count
num_workers = 4          # Number of threads to run in parallel
context = 10             # Context window size
downsampling = 1e-3       # Downsample setting for frequent words

# Initialize and train the model (this will take some time)
from gensim.models import Word2Vec
print ("Training model...")
model = Word2Vec(sen_corpus, workers=num_workers, \
                  size=num_features, min_count = min_word_count, \
                  window = context)

...

```

 \n# Set values for various parameters\nnum\_features = 300 # Word vector dimension

```

import gensim
from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle

# in this project we are using a pretrained model by google
# its 3.3G file, once you load this into your memory
# it occupies ~9Gb, so please do this step only if you have >12G of ram
# we will provide a pickle file which contains a dict ,
# and it contains all our corpus words as keys and model[word] as values
# To use this code-snippet, download "GoogleNews-vectors-negative300.bin"
# from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTtSS21pQmM/edit
# it's 1.9GB in size.

...
model = KeyedVectors.load_word2vec_format('GoogleNews-vectors-negative300.bin', binary=True)
...

```

#if you do NOT have RAM >= 12GB, use the code below.
with open('word2vec\_model', 'rb') as handle:
 model = pickle.load(handle)

# Utility functions

```

def get_word_vec(sentence, doc_id, m_name):
    # sentence : title of the apparel
    # doc_id: document id in our corpus
    # m_name: model information it will take two values
    # if m_name == 'avg', we will append the model[i], w2v representation of word i
    # if m_name == 'weighted', we will multiply each w2v[word] with the idf(word)
    vec = []
    for i in sentence.split():
        if i in vocab:
            if m_name == 'weighted' and i in idf_title_vectorizer.vocabulary_:
                vec.append(idf_title_features[doc_id, idf_title_vectorizer.vocabulary_[i]] * mod
            elif m_name == 'avg':
                vec.append(model[i])
        else:
            # if the word in our corpus is not there in the google word2vec corpus, we are just
            vec.append(np.zeros(shape=(300,)))
    # we will return a numpy array of shape (#number of words in title * 300 ) 300 = len(w2v_mod
    # each row represents the word2vec representation of each word (weighted/avg) in given senta
    return np.array(vec)

def get_distance(vec1, vec2):
    # vec1 = np.array(#number_of_words_title1 * 300), each row is a vector of length 300 corresp
    # vec2 = np.array(#number_of_words_title2 * 300), each row is a vector of length 300 corresp

```

```

final_dist = []
# for each vector in vec1 we calculate the distance(euclidean) to all vectors in vec2
for i in vec1:
    dist = []
    for j in vec2:
        # np.linalg.norm(i-j) will result the euclidean distance between vectors i, j
        dist.append(np.linalg.norm(i-j))
    final_dist.append(np.array(dist))
# final_dist = np.array(#number of words in title1 * #number of words in title2)
# final_dist[i,j] = euclidean distance between vectors i, j
return np.array(final_dist)

def heat_map_w2v(sentence1, sentence2, url, doc_id1, doc_id2, model):
    # sentance1 : title1, input apparel
    # sentance2 : title2, recommended apparel
    # url: apparel image url
    # doc_id1: document id of input apparel
    # doc_id2: document id of recommended apparel
    # model: it can have two values, 1. avg 2. weighted

    s1_vec = np.array(#number_of_words_title1 * 300), each row is a vector(weighted/avg) of len
    s1_vec = get_word_vec(sentence1, doc_id1, model)
    s2_vec = np.array(#number_of_words_title1 * 300), each row is a vector(weighted/avg) of len
    s2_vec = get_word_vec(sentence2, doc_id2, model)

    # s1_s2_dist = np.array(#number of words in title1 * #number of words in title2)
    # s1_s2_dist[i,j] = euclidean distance between words i, j
    s1_s2_dist = get_distance(s1_vec, s2_vec)

    # devide whole figure into 2 parts 1st part displays heatmap 2nd part displays image of appa
    gs = gridspec.GridSpec(2, 2, width_ratios=[4,1],height_ratios=[2,1])
    fig = plt.figure(figsize=(15,15))

    ax = plt.subplot(gs[0])
    # plotting the heap map based on the pairwise distances
    ax = sns.heatmap(np.round(s1_s2_dist,4), annot=True)
    # set the x axis labels as recommended apparels title
    ax.set_xticklabels(sentence2.split())
    # set the y axis labels as input apparels title
    ax.set_yticklabels(sentence1.split())
    # set title as recommended apparels title
    ax.set_title(sentence2)

    ax = plt.subplot(gs[1])
    # we remove all grids and axis labels for image
    ax.grid(False)
    ax.set_xticks([])
    ax.set_yticks([])
    display_img(url, ax, fig)

    plt.show()

# vocab = stores all the words that are there in google w2v model
# vocab = model.wv.vocab.keys() # if you are using Google word2Vec

vocab = model.keys()
# this function will add the vectors of each word and returns the avg vector of given sentance
def build_avg_vec(sentence, num_features, doc_id, m_name):
    # sentace: its title of the apparel
    # num_features: the lenght of word2vec vector, its values = 300
    # m_name: model information it will take two values
        # if m_name == 'avg', we will append the model[i], w2v representation of word i
        # if m_name == 'weighted', we will multiply each w2v[word] with the idf(word)

    featureVec = np.zeros((num_features,), dtype="float32")
    """
    .....
    """

```

```
# we will initialize a vector of size 300 with all zeros
# we add each word2vec(wordi) to this fatureVec
nwords = 0

for word in sentence.split():
    nwords += 1
    if word in vocab:
        if m_name == 'weighted' and word in idf_title_vectorizer.vocabulary_:
            featureVec = np.add(featureVec, idf_title_features[doc_id, idf_title_vectorizer.
        elif m_name == 'avg':
            featureVec = np.add(featureVec, model[word])
if(nwords>0):
    featureVec = np.divide(featureVec, nwords)
# returns the avg vector of given sentance, its of shape (1, 300)
return featureVec
```

## ▼ [9.2] Average Word2Vec product similarity.

```
doc_id = 0
w2v_title = []
# for every title we build a avg vector representation
for i in data['title']:
    w2v_title.append(build_avg_vec(i, 300, doc_id, 'avg'))
    doc_id += 1

# w2v_title = np.array(# number of doc in courpus * 300), each row corresponds to a doc
w2v_title = np.array(w2v_title)

def avg_w2v_model(doc_id, num_results):
    # doc_id: apparel's id in given corpus

    # dist(x, y) = sqrt(dot(x, x) - 2 * dot(x, y) + dot(y, y))
    pairwise_dist = pairwise_distances(w2v_title, w2v_title[doc_id].reshape(1, -1))

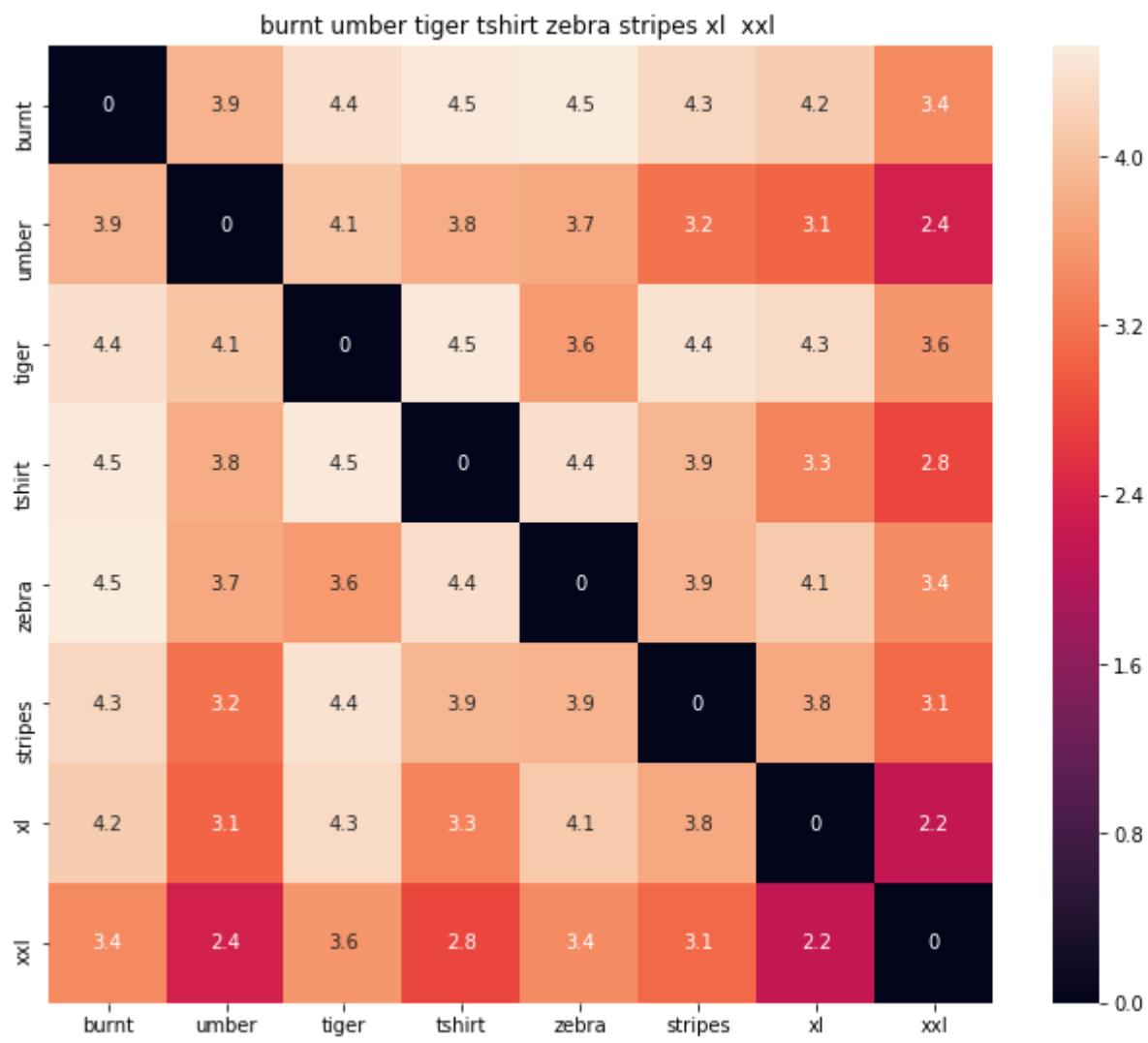
    # np.argsort will return indices of 9 smallest distances
    indices = np.argsort(pairwise_dist.flatten())[0:num_results]
    #pdists will store the 9 smallest distances
    pdists = np.sort(pairwise_dist.flatten())[0:num_results]

    #data frame indices of the 9 smallest distace's
    df_indices = list(data.index[indices])

    for i in range(0, len(indices)):
        heat_map_w2v(data['title'].loc[df_indices[0]], data['title'].loc[df_indices[i]], data['me
        print('ASIN :', data['asin'].loc[df_indices[i]])
        print('BRAND :', data['brand'].loc[df_indices[i]])
        print('euclidean distance from given input image :', pdists[i])
        print('='*125)

avg_w2v_model(12566, 20)
# in the give heat map, each cell contains the euclidean distance between words i, j
```

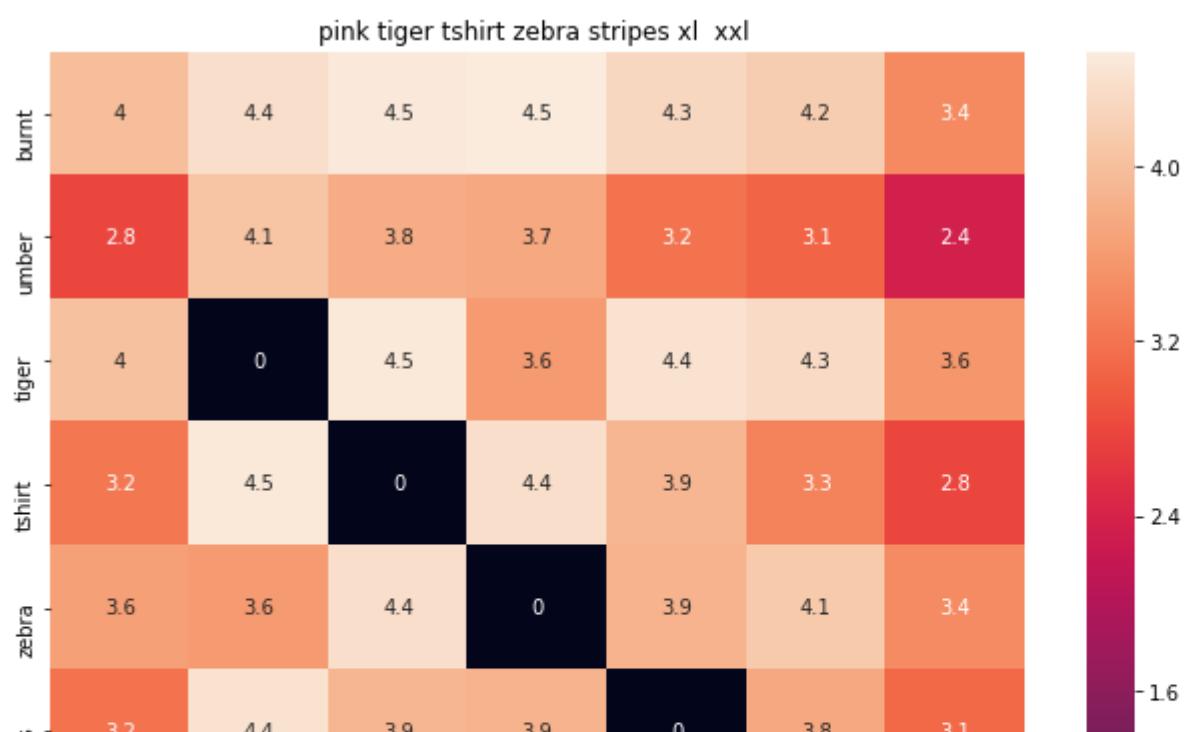


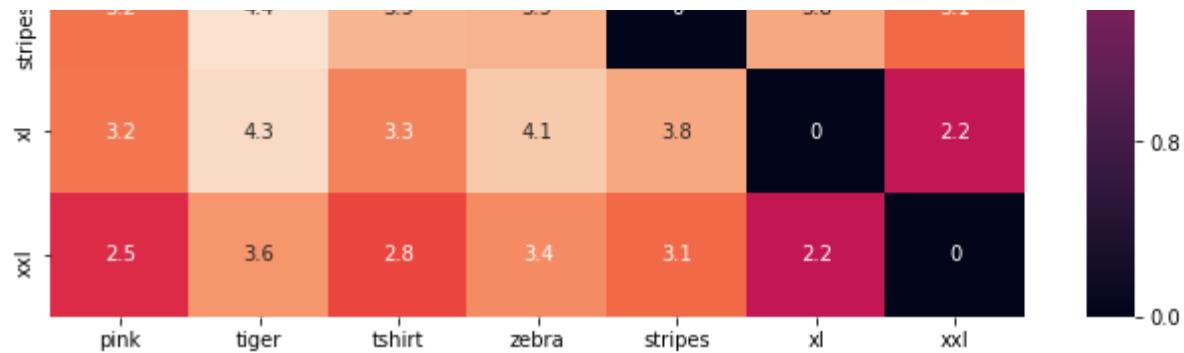


ASIN : B00JXQB5FQ

BRAND : Si Row

euclidean distance from given input image : 0.0

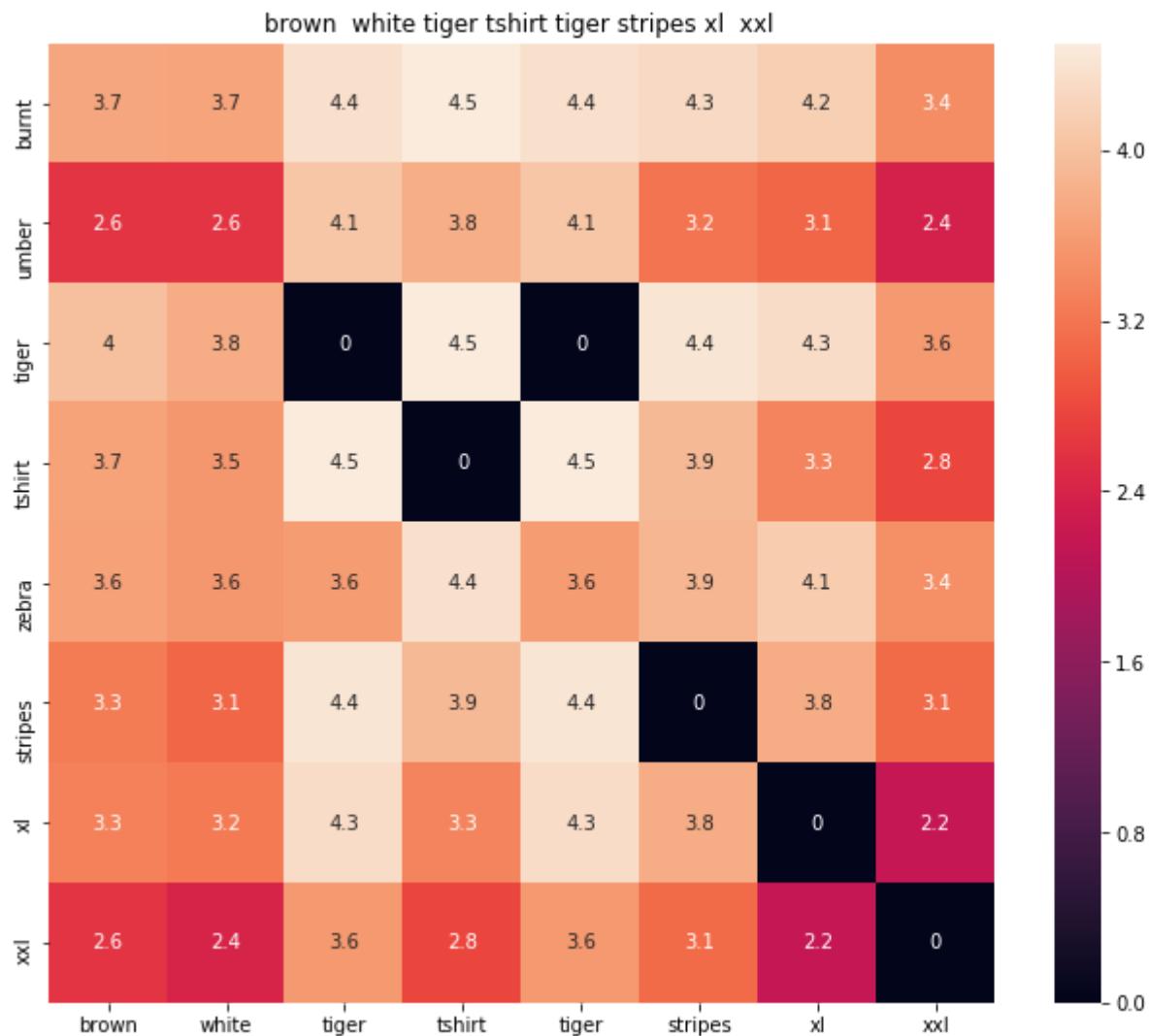




ASIN : B00JXQASS6

BRAND : Si Row

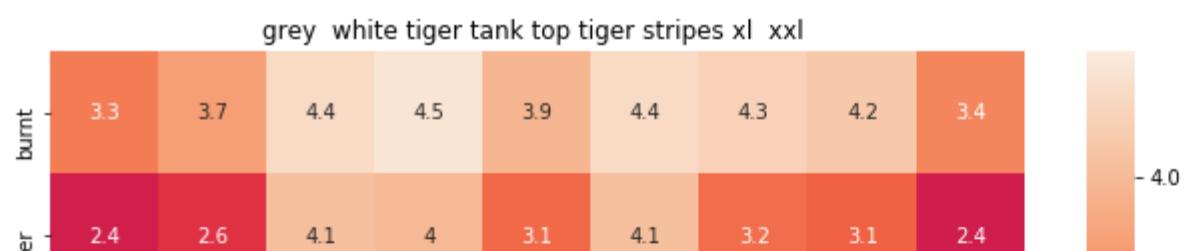
euclidean distance from given input image : 0.5891928

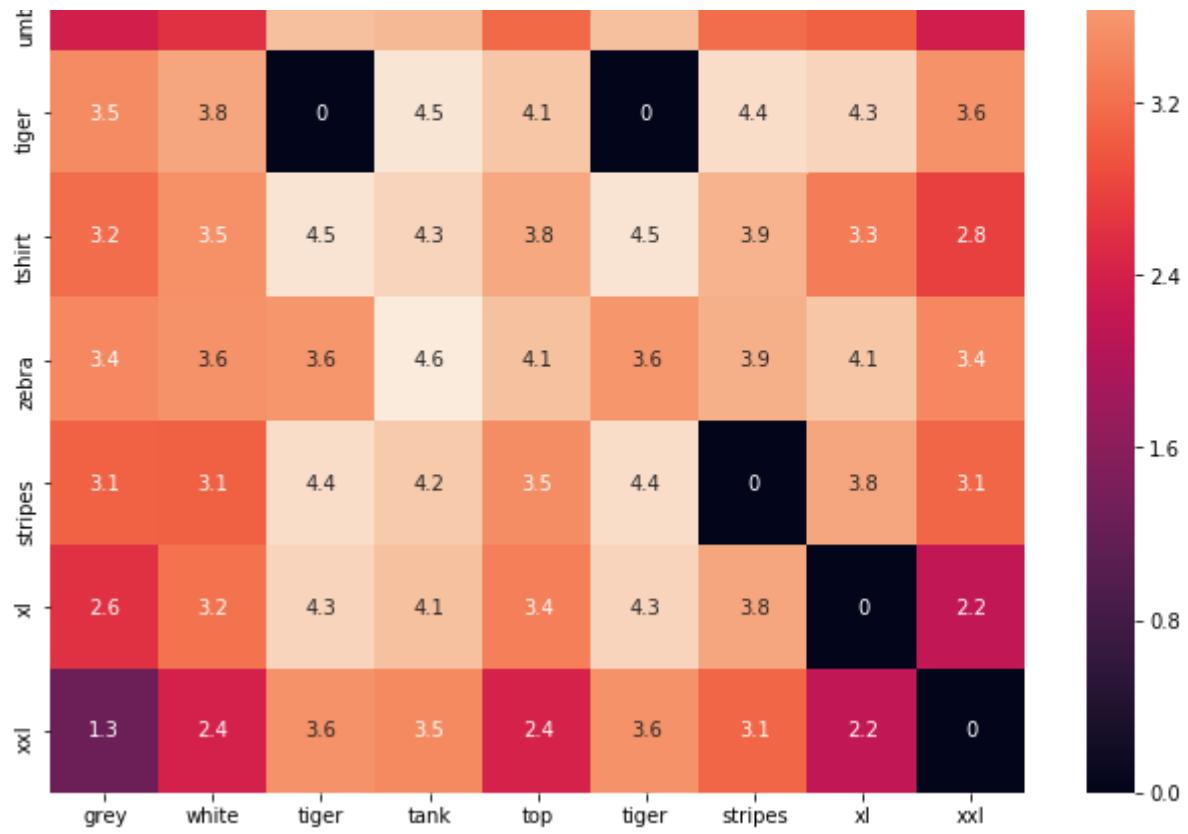


ASIN : B00JXQCWTO

BRAND : Si Row

euclidean distance from given input image : 0.7003436



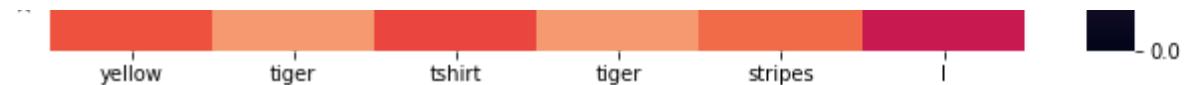


ASIN : B00JXQAFZ2

BRAND : Si Row

euclidean distance from given input image : 0.8928398



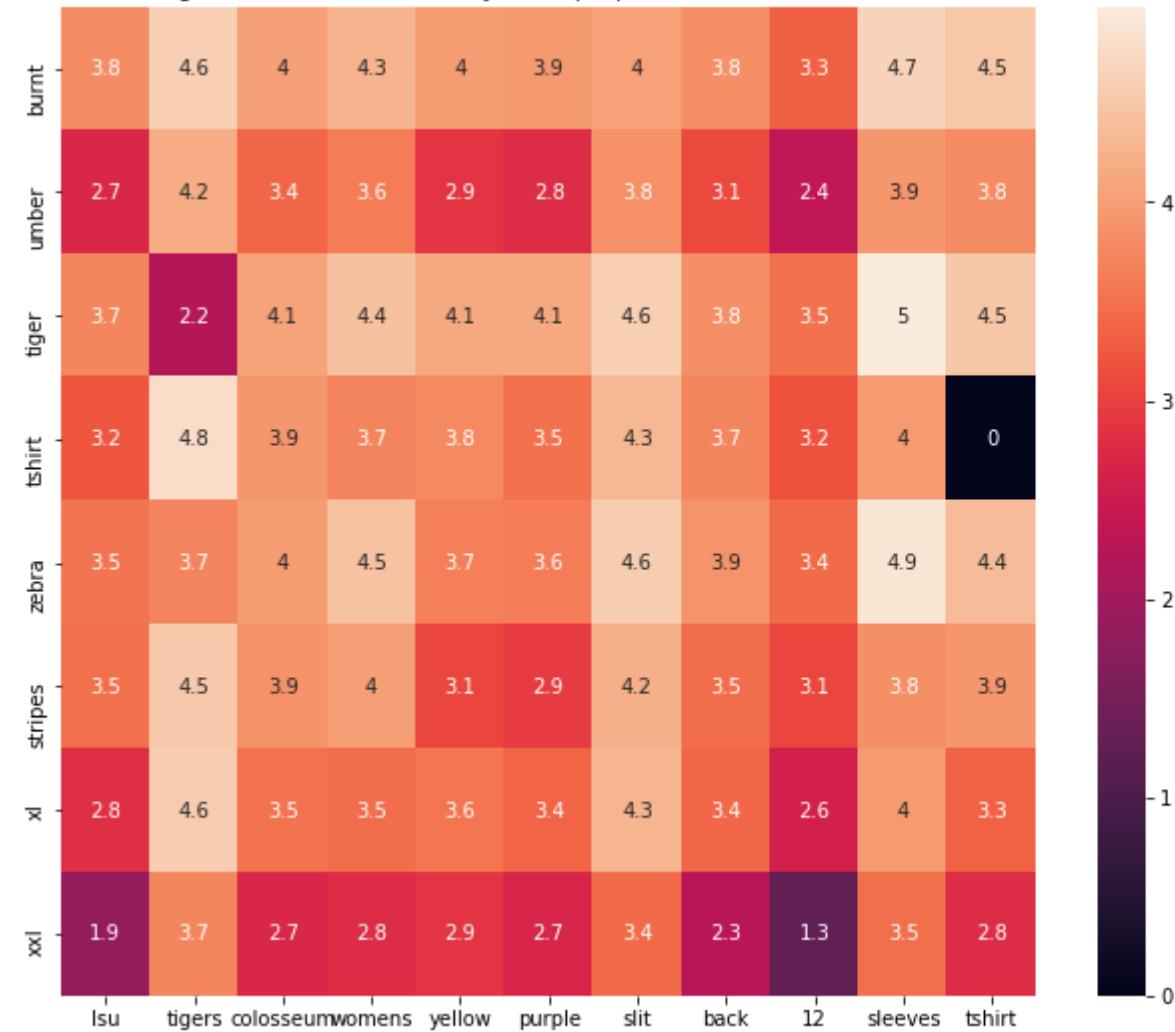


ASIN : B00JXQCUIC

BRAND : Si Row

euclidean distance from given input image : 0.9560129

lsu tigers colosseum womens yellow purple slit back 12 sleeves tshirt

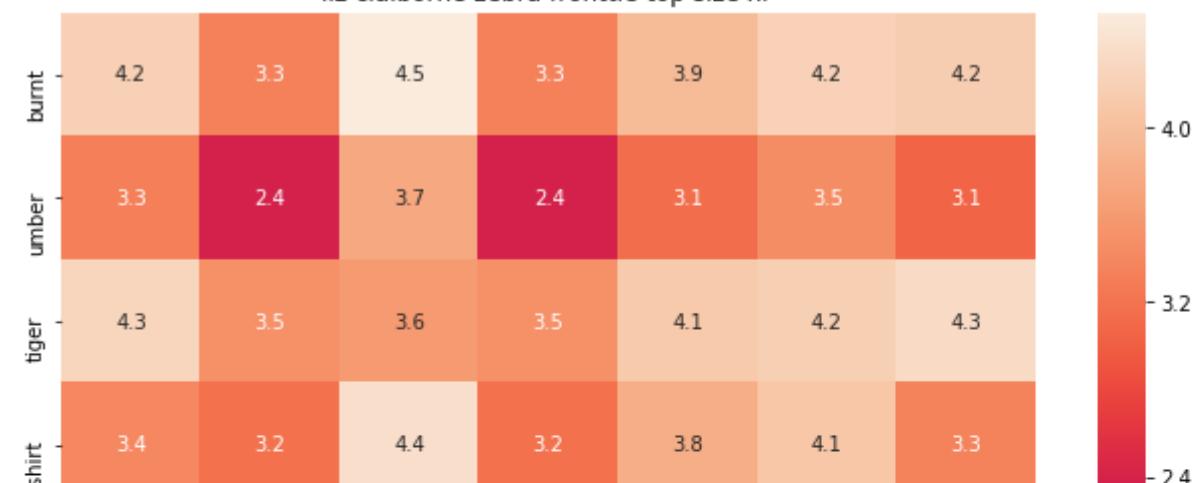


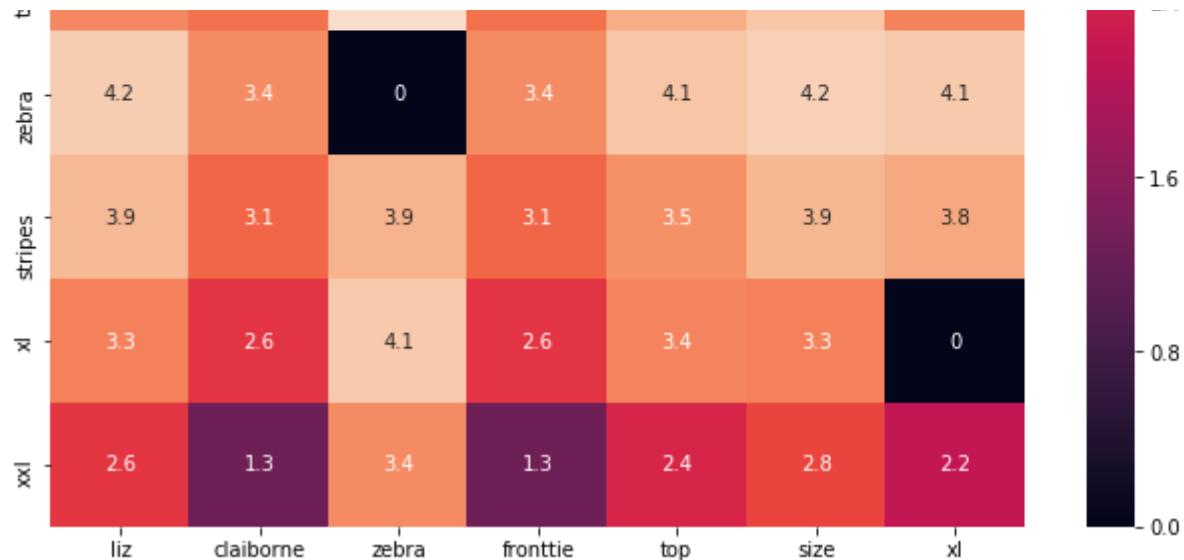
ASIN : B073R5Q8HD

BRAND : Colosseum

euclidean distance from given input image : 1.0229691

liz claiborne zebra fronttie top size xl





ASIN : B06XBY5QXL

BRAND : Liz Claiborne

euclidean distance from given input image : 1.0669324



ASIN : B01L8L73M2

BRAND : Hotgirl4 Raglan Design

euclidean distance from given input image : 1.0731406

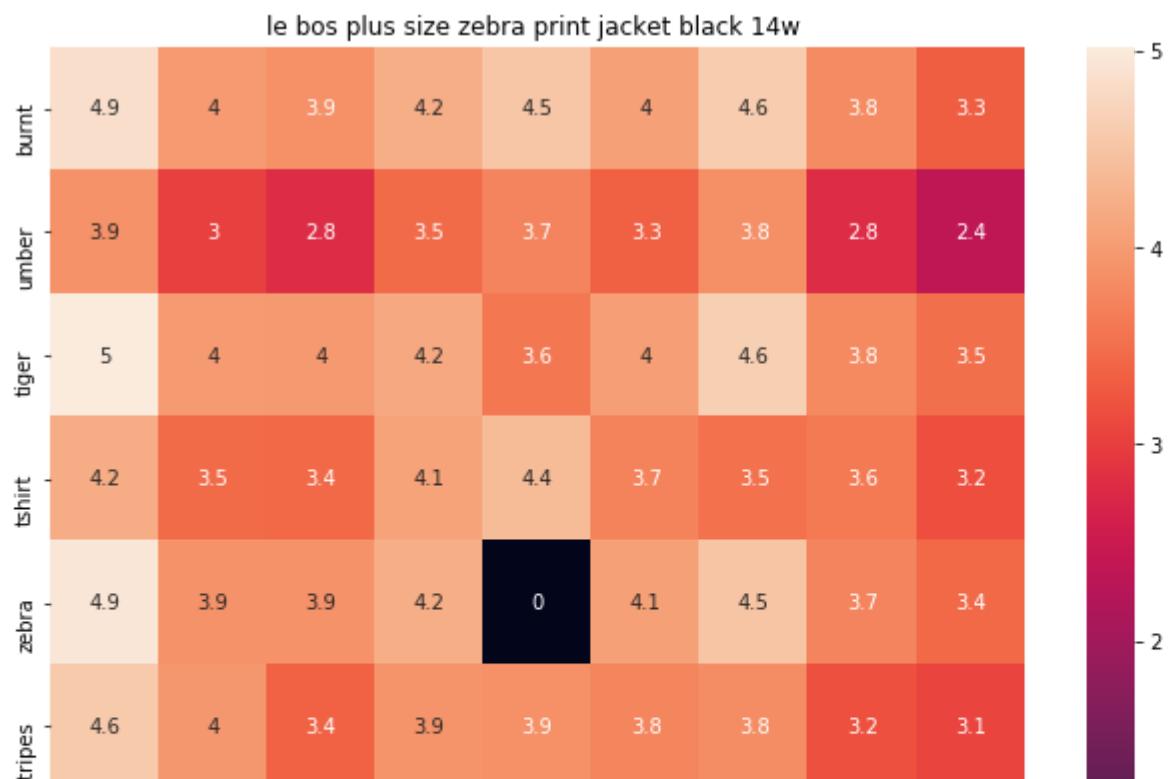
vansty hippo naughty round neck tshirt women yellow size xl

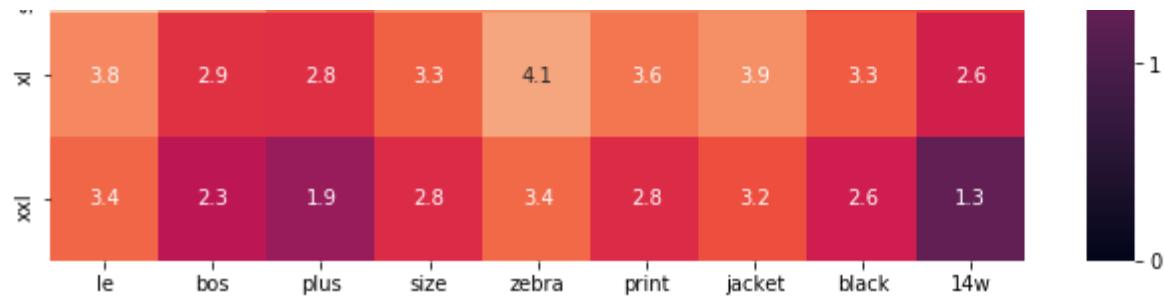


ASIN : B01EJ5H06

BRAND : Vansty

euclidean distance from given input image : 1.0757191

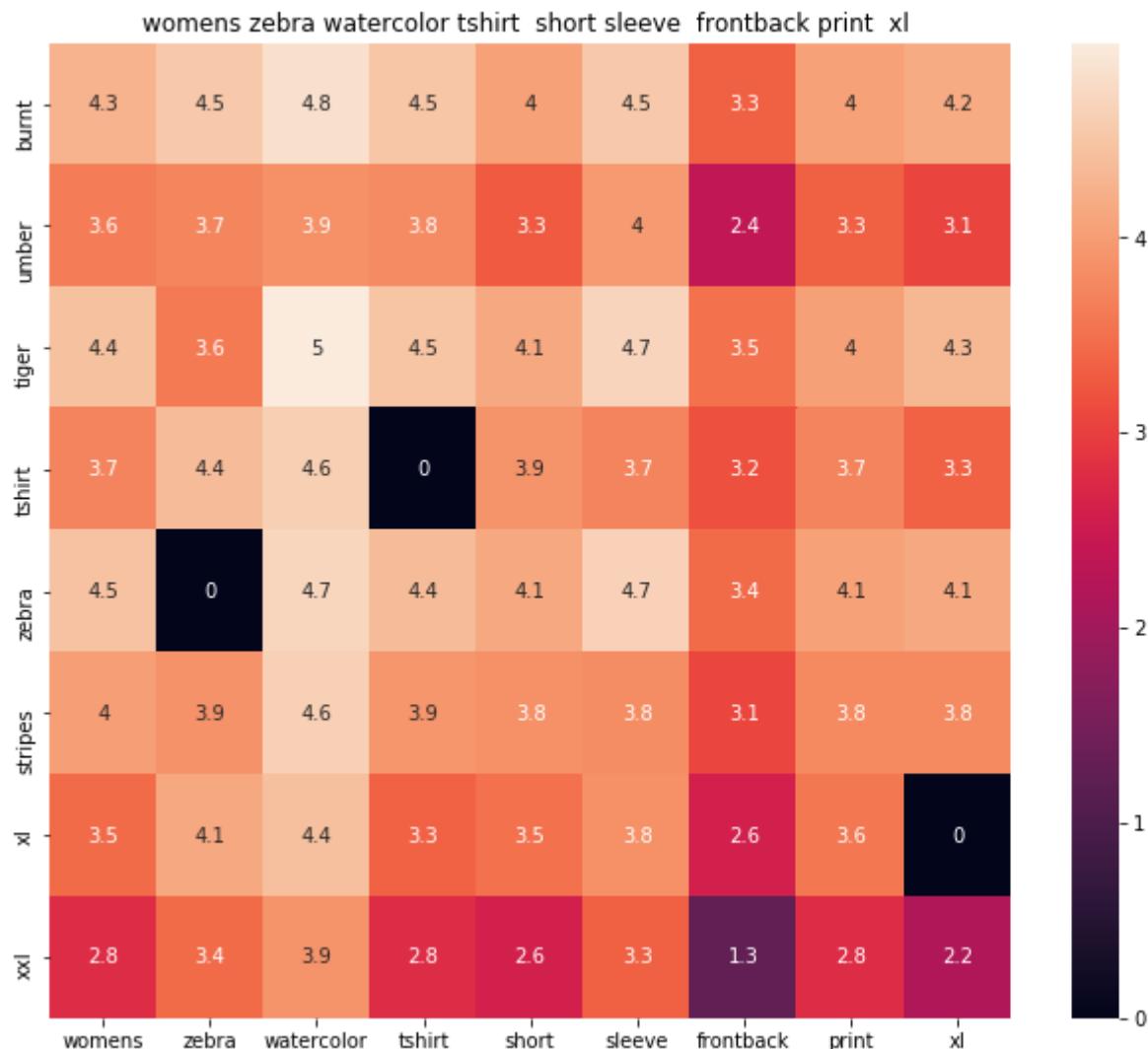




ASIN : B01B01XRK8

BRAND : Le Bos

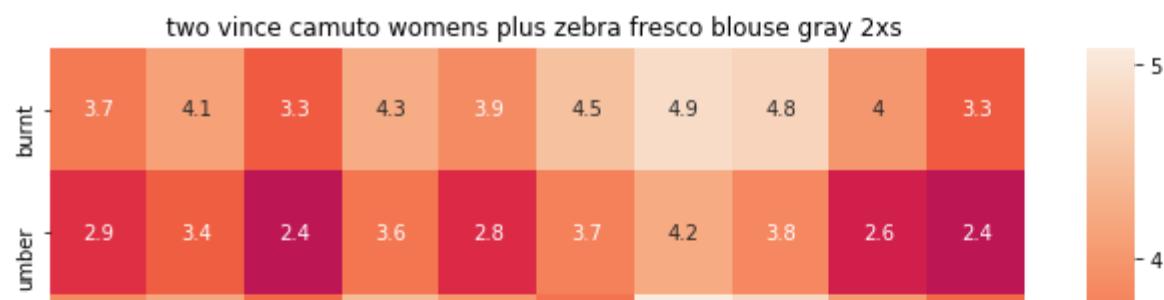
euclidean distance from given input image : 1.0839965

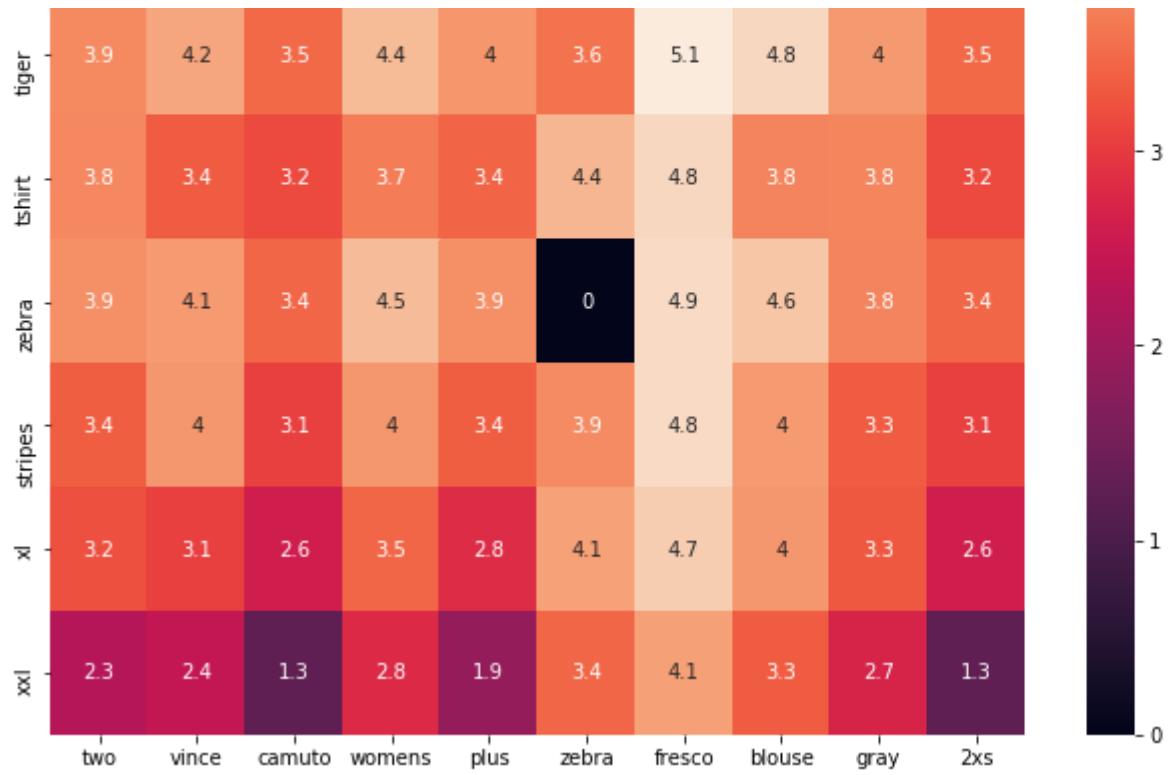


ASIN : B072R2JXKW

BRAND : WHAT ON EARTH

euclidean distance from given input image : 1.0842218

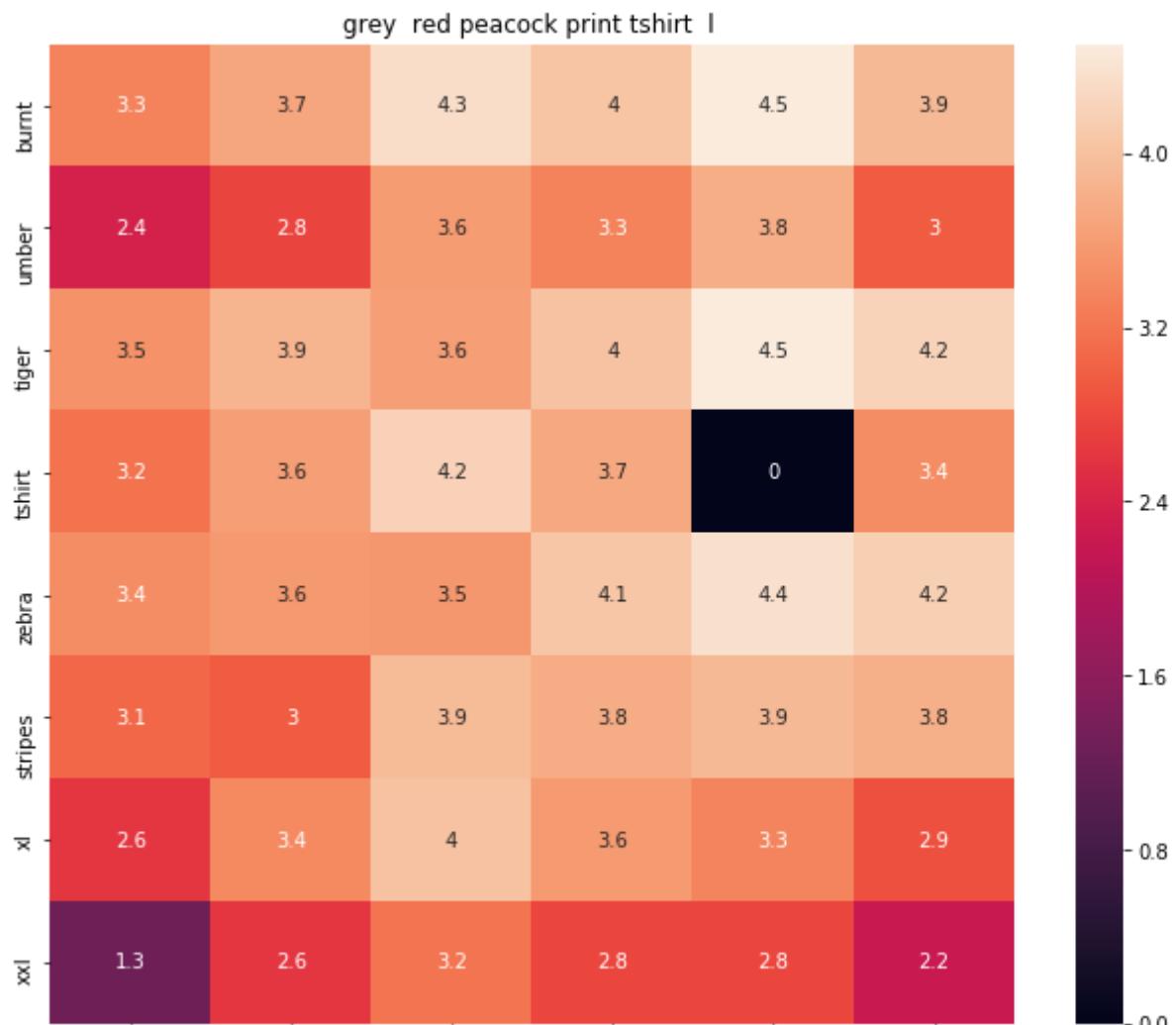




ASIN : B074MJRGW6

BRAND : Two by Vince Camuto

euclidean distance from given input image : 1.0895039

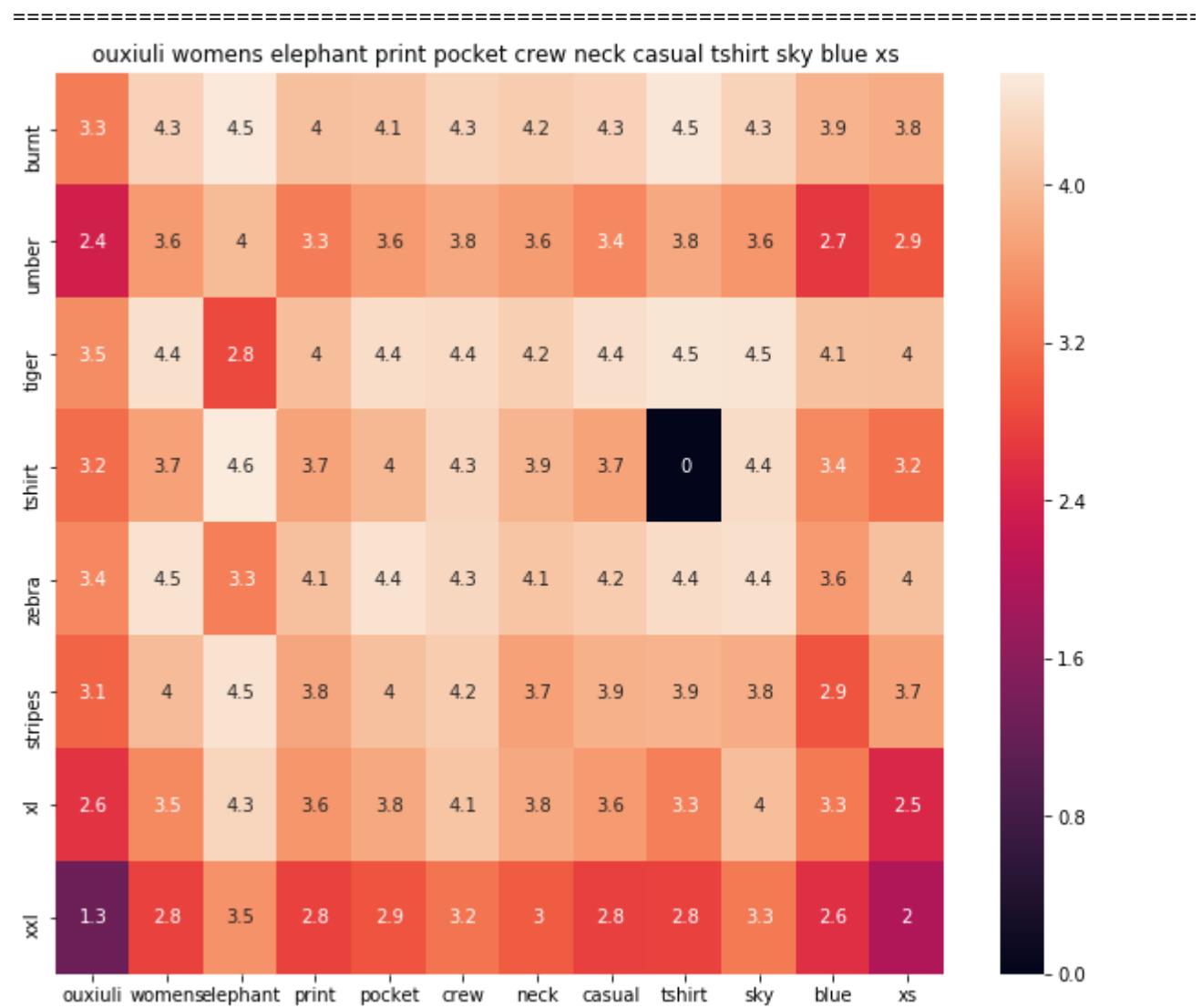


grey red peacock print tshirt |

ASIN : B00JXQCFRS

BRAND : Si Row

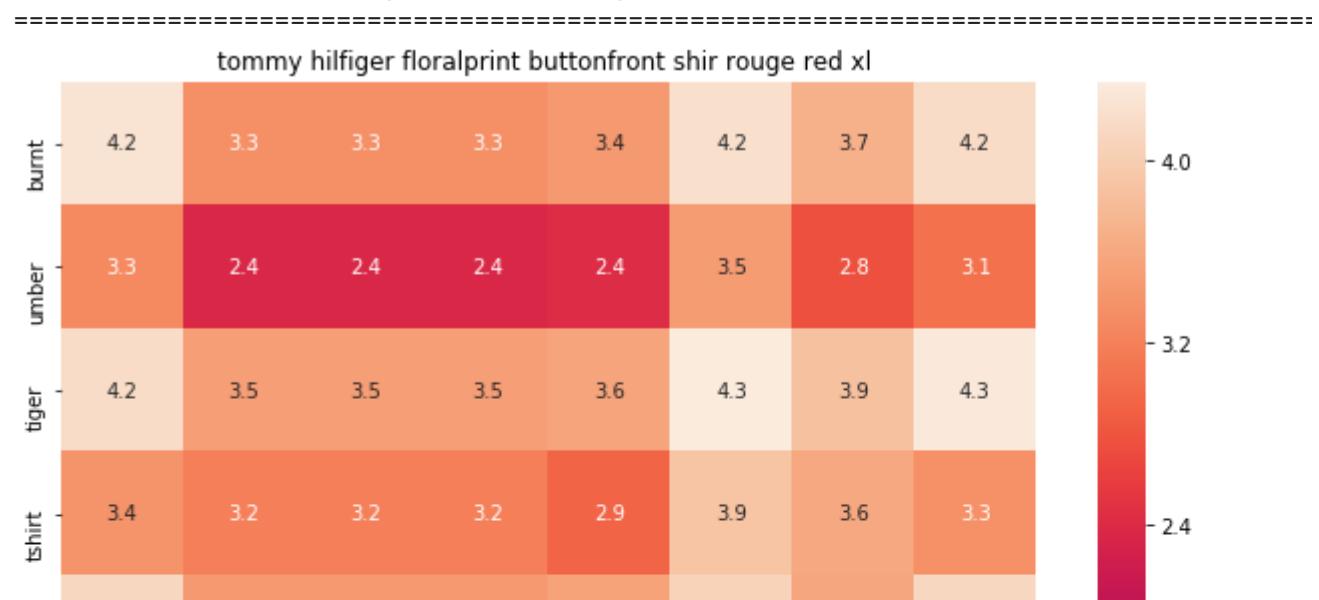
euclidean distance from given input image : 1.0900588

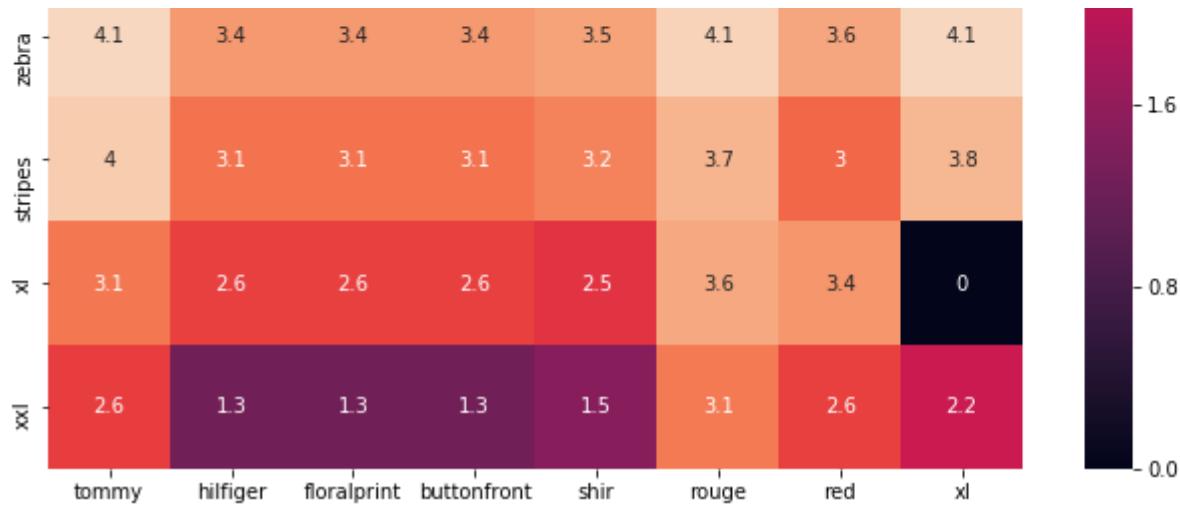


ASIN : B01I53HU6K

BRAND : ouxiuli

euclidean distance from given input image : 1.0920112

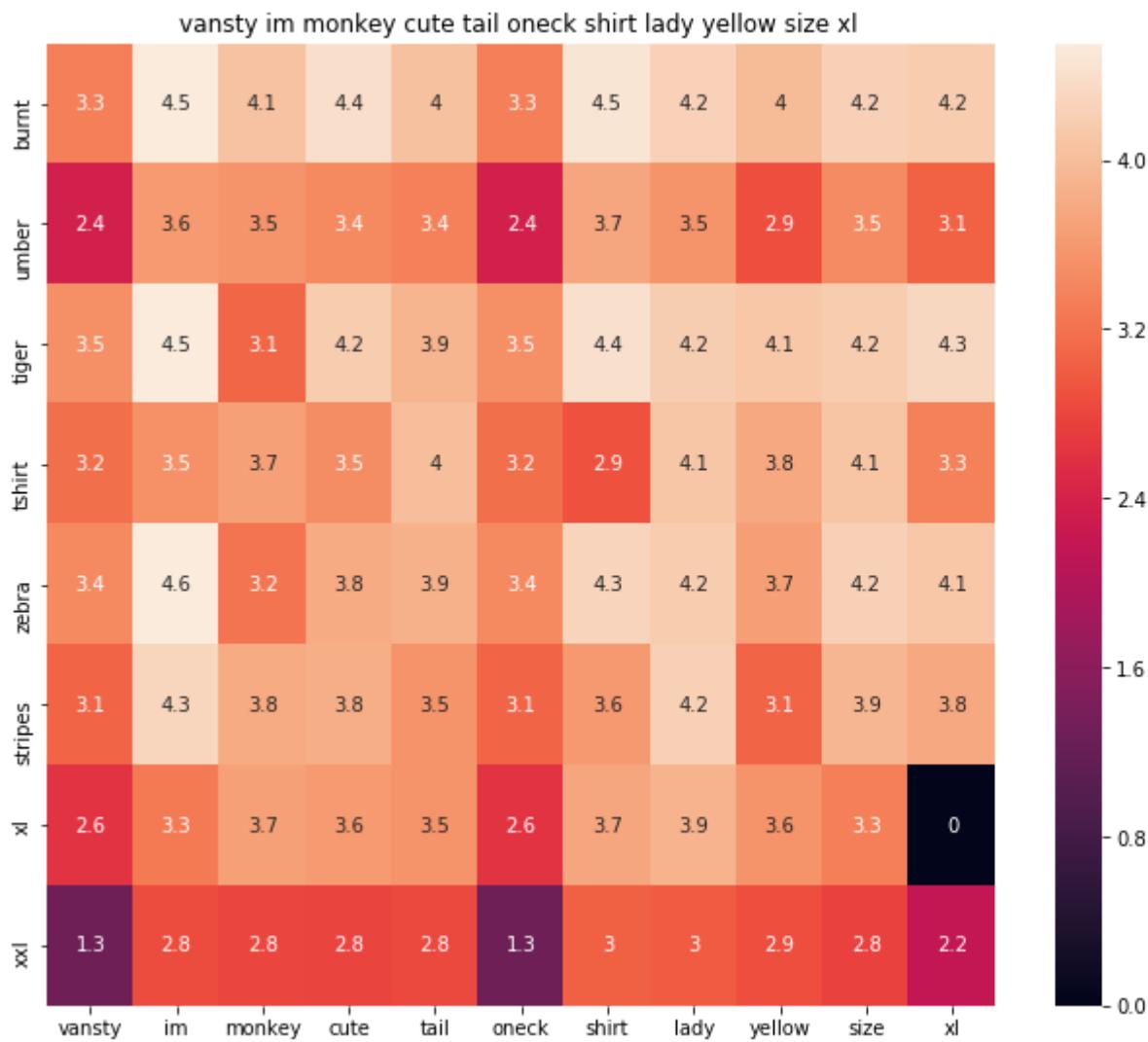




ASIN : B0711NGTQM

BRAND : THILFIGER RTW

euclidean distance from given input image : 1.0923418

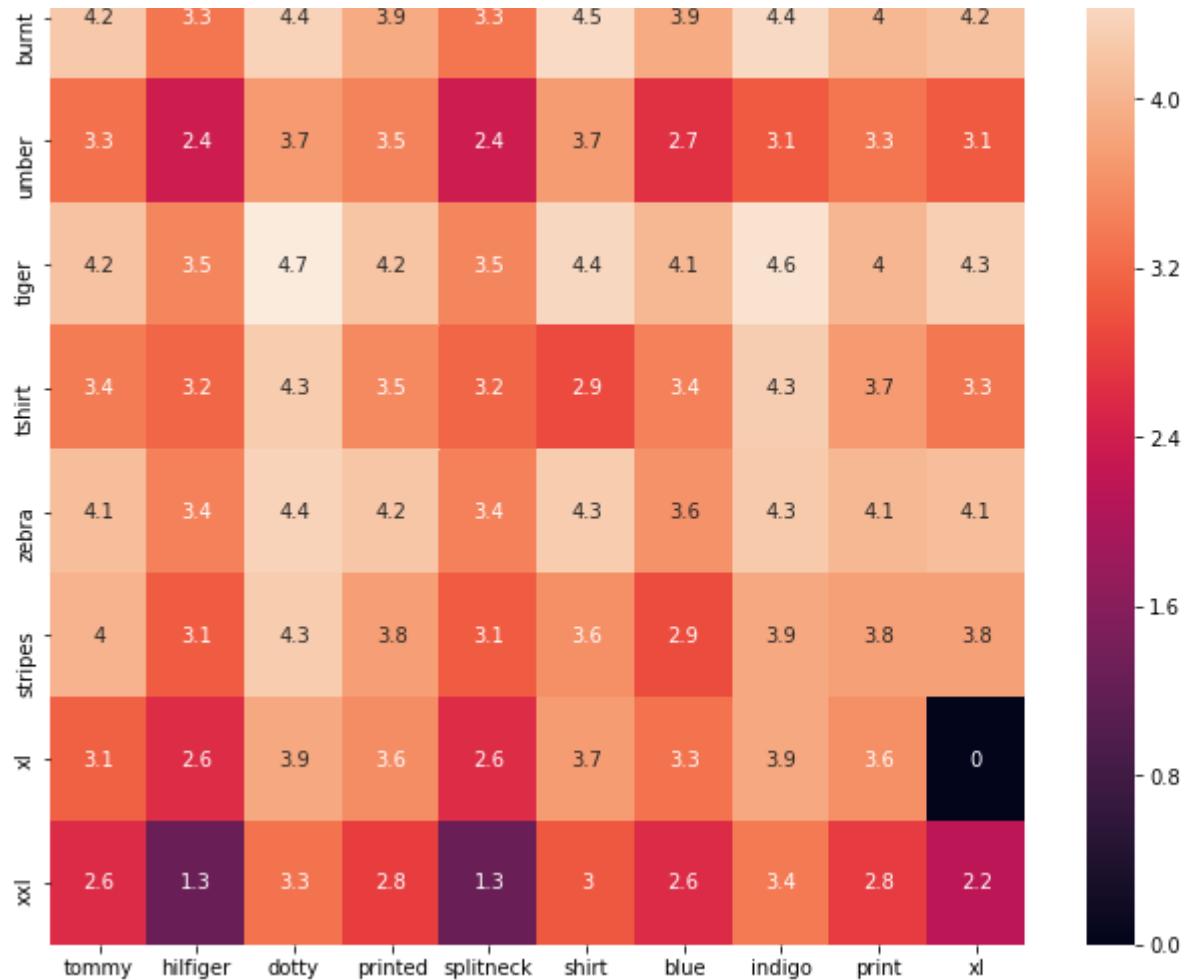


ASIN : B01EFSL08Y

BRAND : Vansty

euclidean distance from given input image : 1.0934006

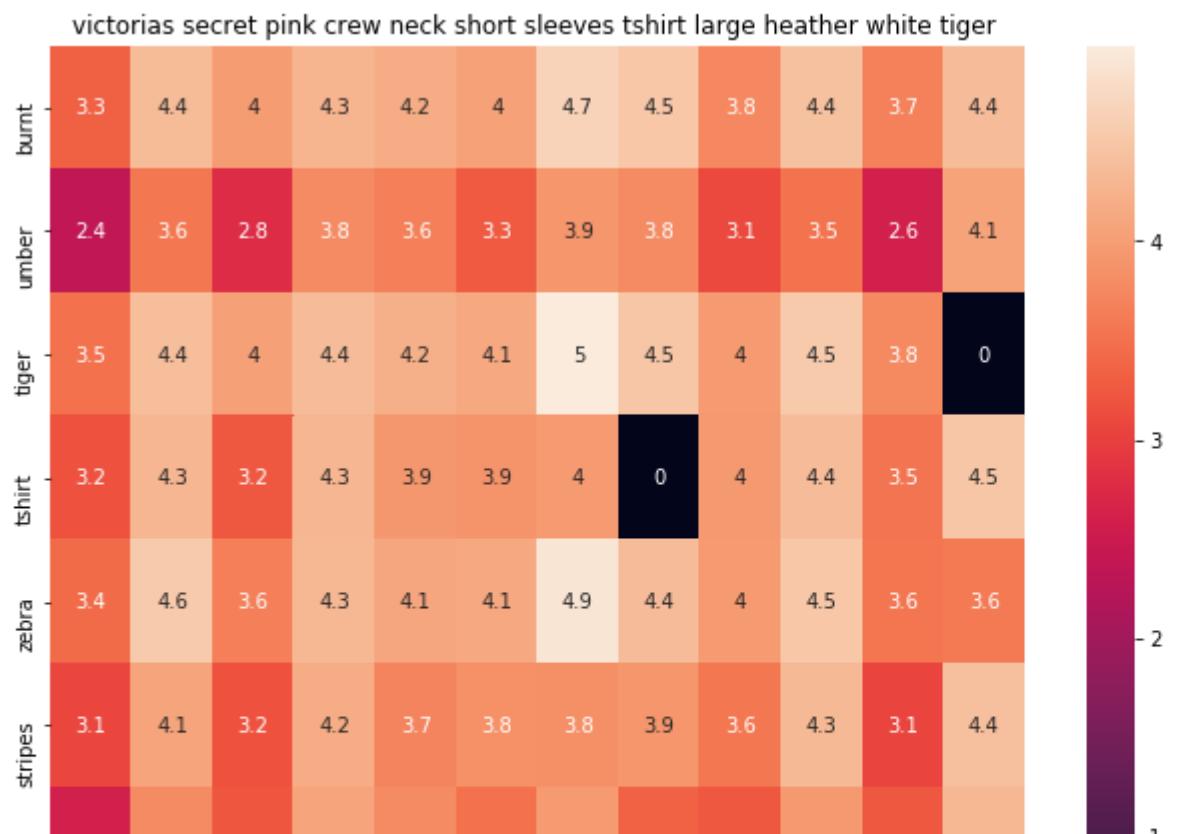


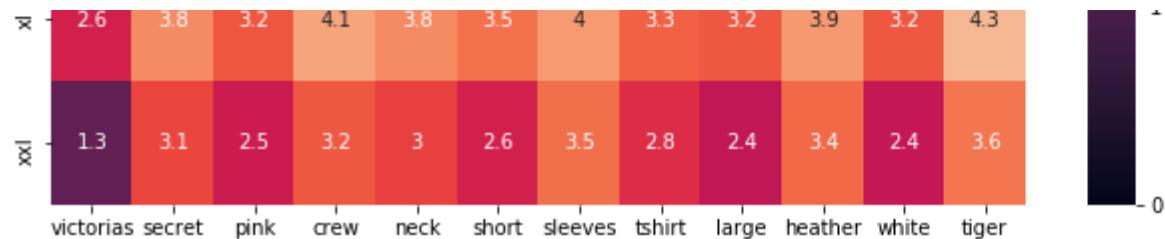


ASIN : B0716TVWQ4

BRAND : THILFIGER RTW

euclidean distance from given input image : 1.0942025

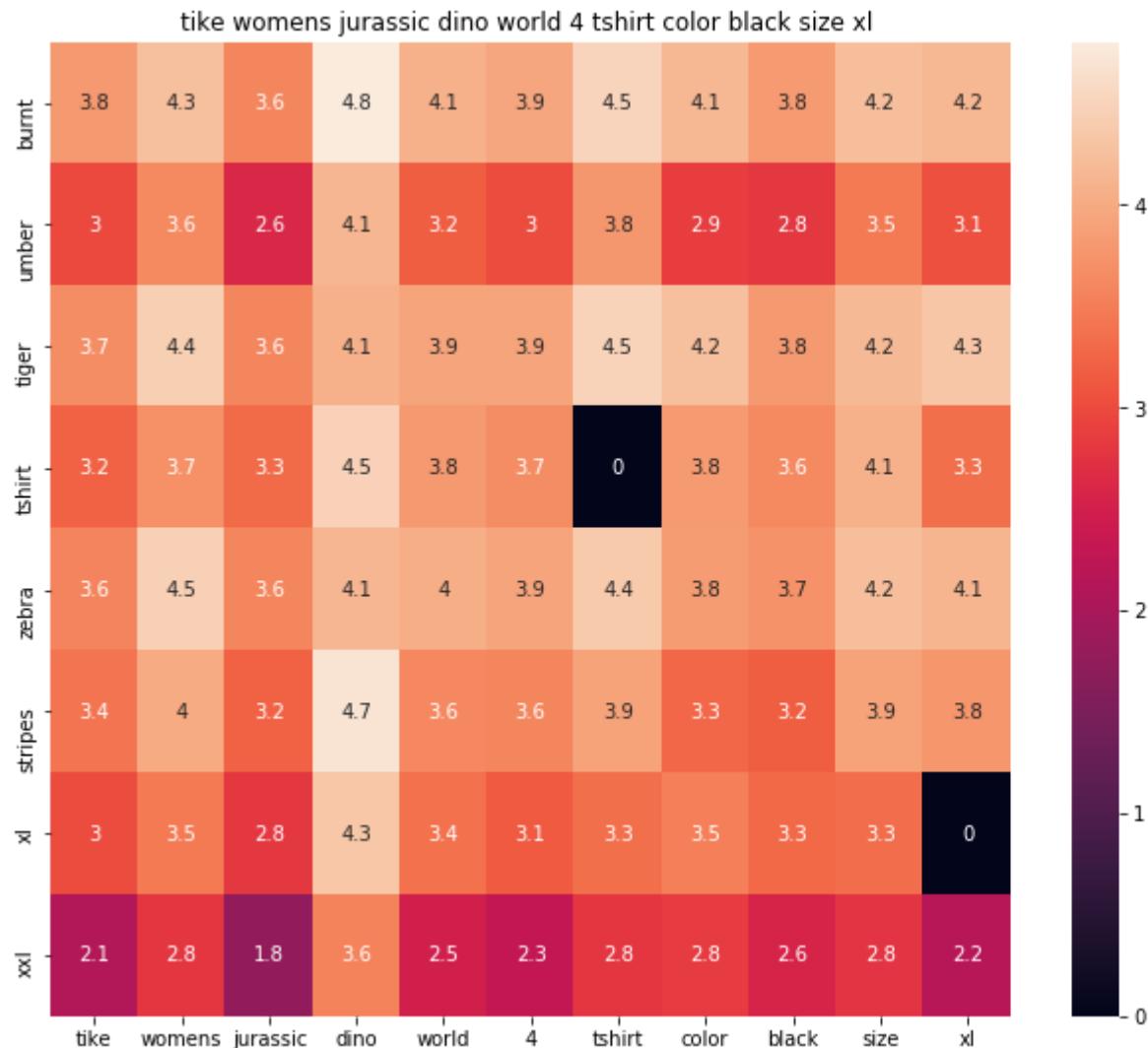




ASIN : B0716MVPGV

BRAND : V.Secret

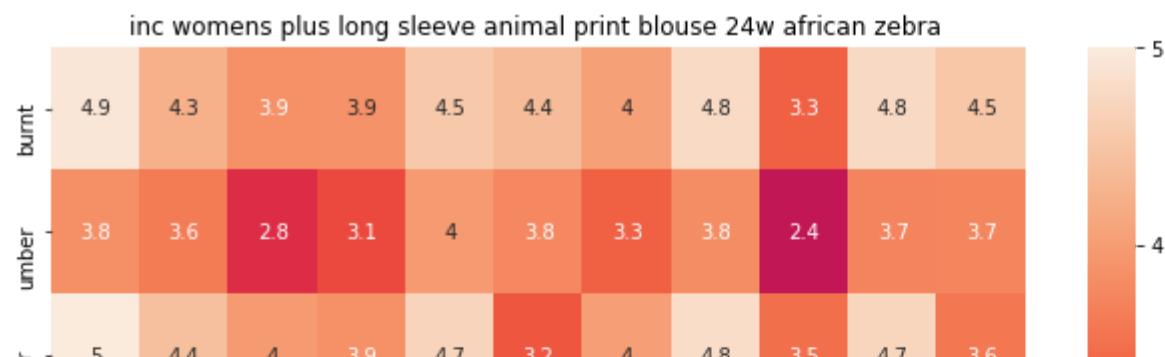
euclidean distance from given input image : 1.0948305

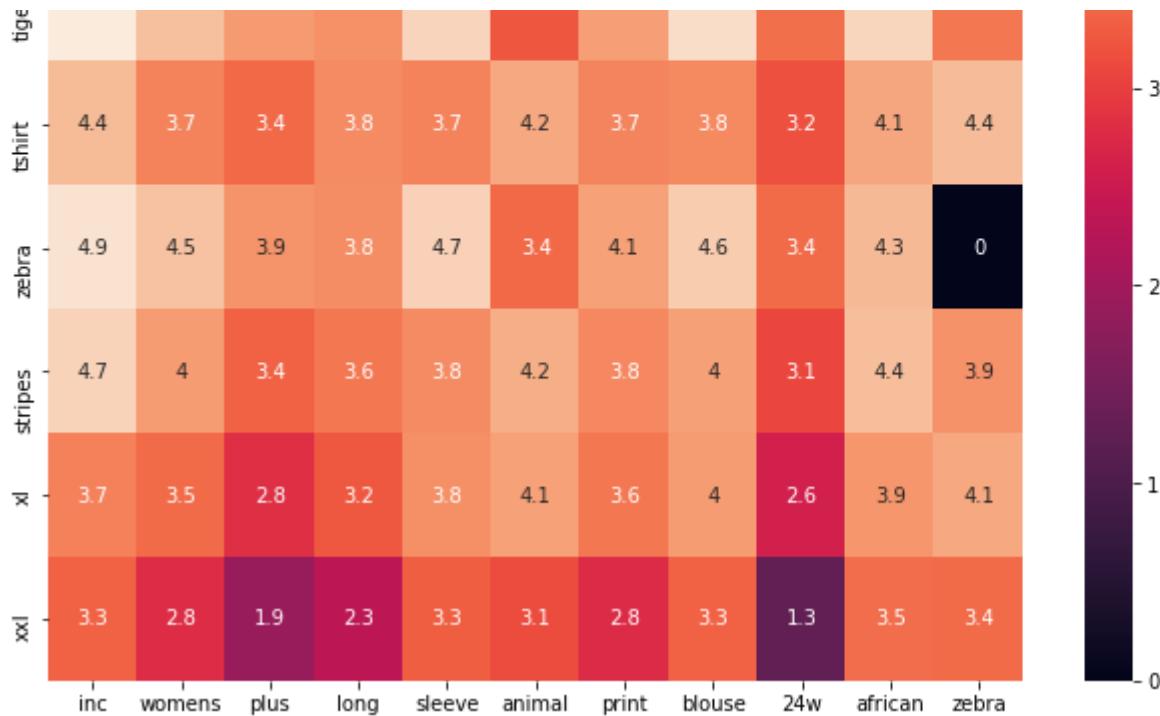


ASIN : B0160PN40I

BRAND : TIKE Fashions

euclidean distance from given input image : 1.0951276





ASIN : B018WDJCUA

BRAND : INC - International Concepts Woman

euclidean distance from given input image : 1.0966893

## ▼ [9.4] IDF weighted Word2Vec for product similarity

```

doc_id = 0
w2v_title_weight = []
# for every title we build a weighted vector representation
for i in data['title']:
    w2v_title_weight.append(build_avg_vec(i, 300, doc_id, 'weighted'))
    doc_id += 1
# w2v_title = np.array(# number of doc in courpus * 300), each row corresponds to a doc
w2v_title_weight = np.array(w2v_title_weight)

def weighted_w2v_model(doc_id, num_results):
    # doc_id: apparel's id in given corpus

    # pairwise_dist will store the distance from given input apparel to all remaining apparels
    # the metric we used here is cosine, the coside distance is mesured as K(X, Y) = <X, Y> / (|X|*|Y|)
    # http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
    pairwise_dist = pairwise_distances(w2v_title_weight, w2v_title_weight[doc_id].reshape(1,-1))

    # np.argsort will return indices of 9 smallest distances
    indices = np.argsort(pairwise_dist.flatten())[0:num_results]
    #pdists will store the 9 smallest distances
    pdists = np.sort(pairwise_dist.flatten())[0:num_results]

    #data frame indices of the 9 smallest distace's
    df_indices = list(data.index[indices])

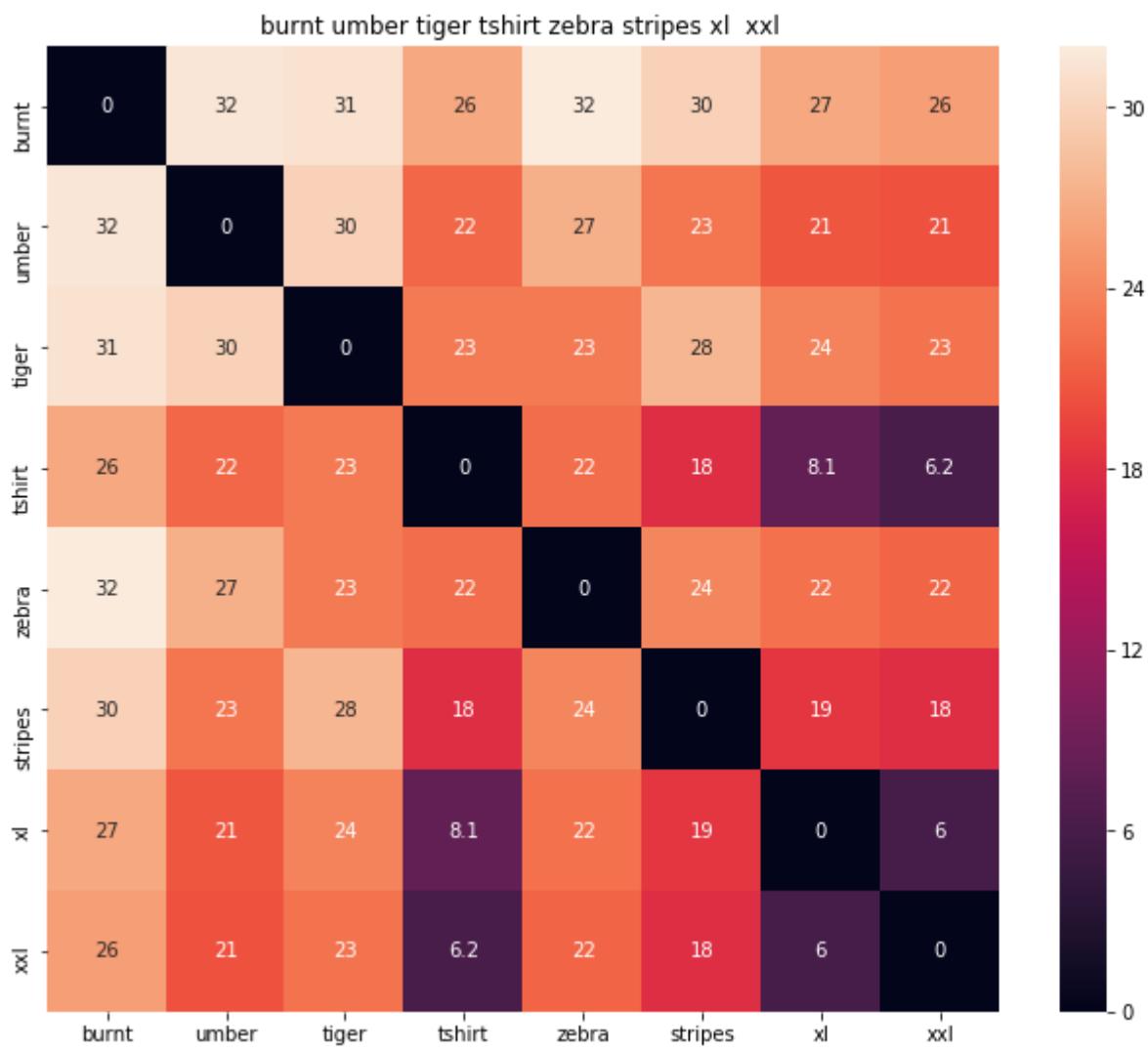
    for i in range(0, len(indices)):
        heat_map_w2v(data['title'].loc[df_indices[0]], data['title'].loc[df_indices[i]], data['me'])
        print('ASIN : ', data['asin'].loc[df_indices[i]])
        print('Brand : ', data['brand'].loc[df_indices[i]])
        print('euclidean distance from input : ', pdists[i])
        print('*125')

weighted_w2v_model(12566, 20)
    
```

```
#931  
#12566
```

```
# in the give heat map, each cell contains the euclidean distance between words i, j
```

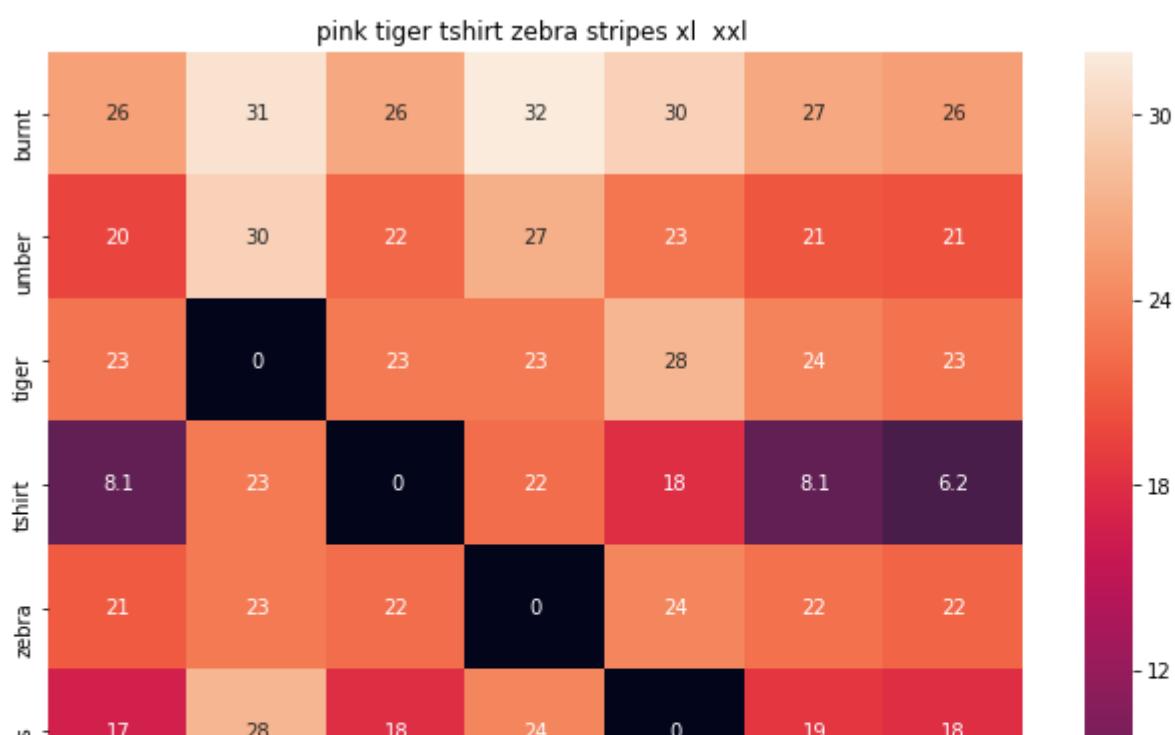


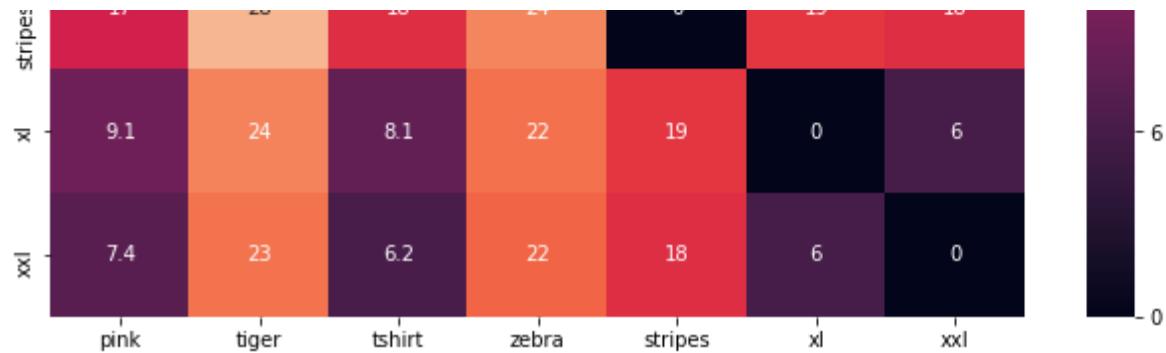


ASIN : B00JXQB5FQ

Brand : Si Row

euclidean distance from input : 0.00390625

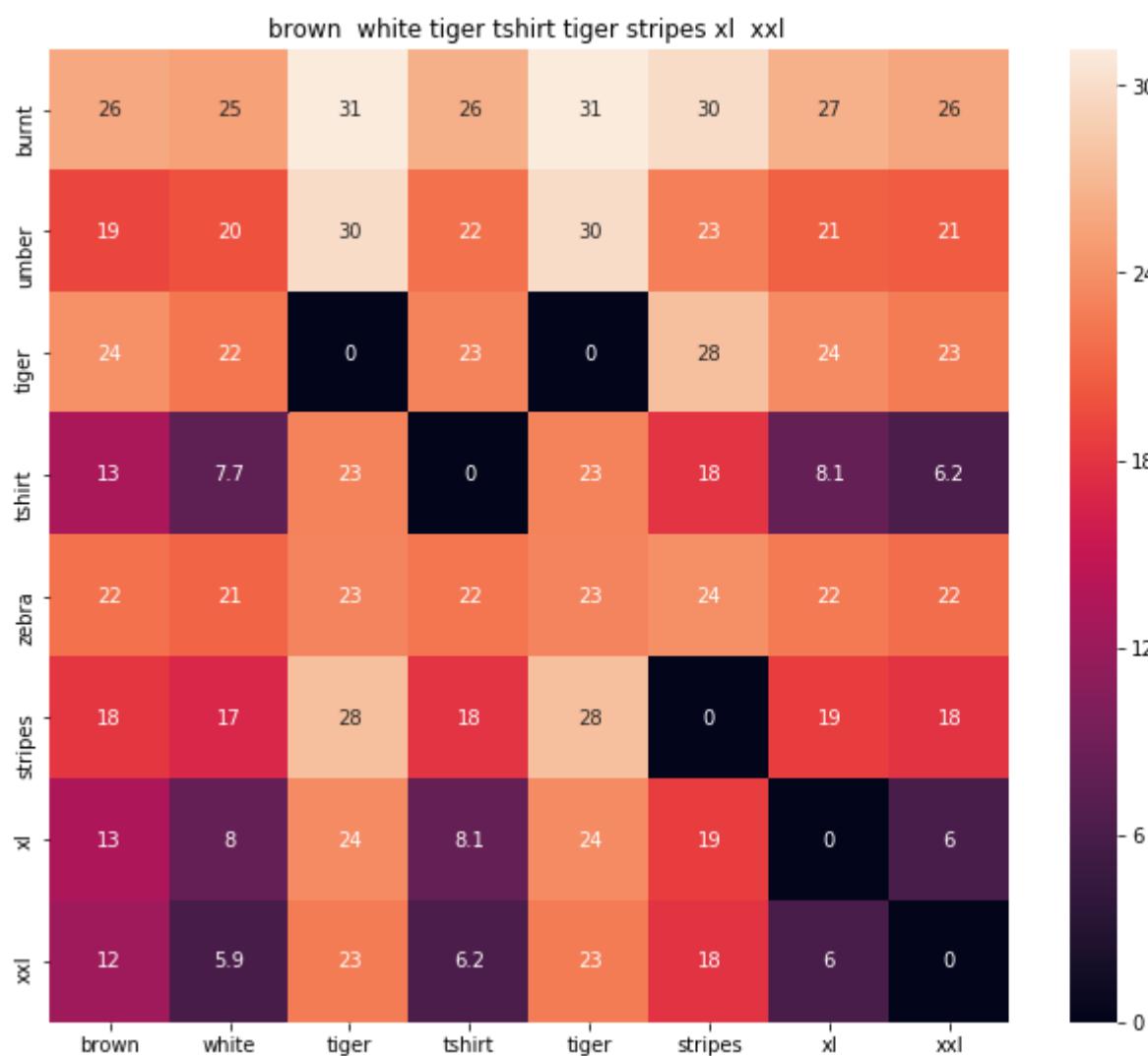




ASIN : B00JXQASS6

Brand : Si Row

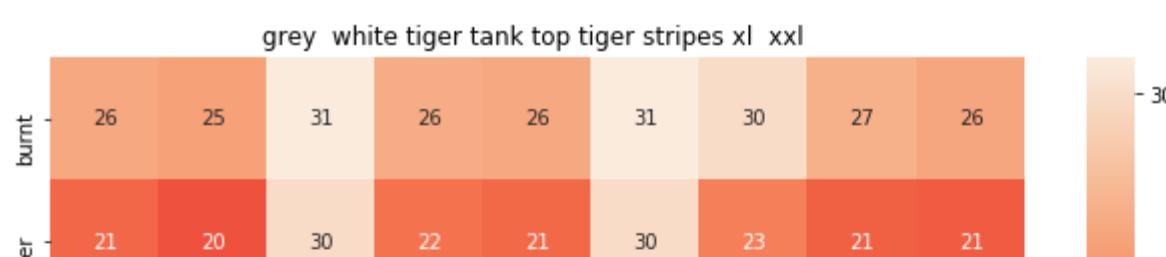
euclidean distance from input : 4.0638876

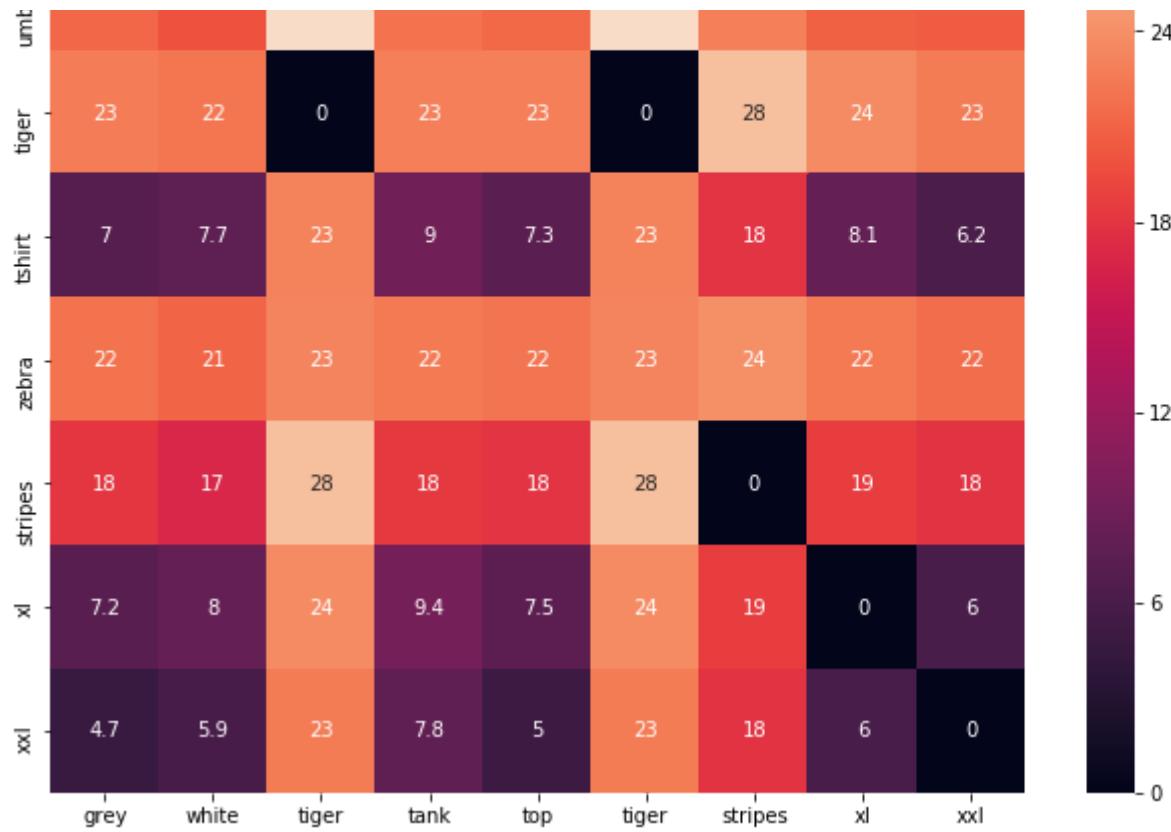


ASIN : B00JXQCWT0

Brand : Si Row

euclidean distance from input : 4.770942

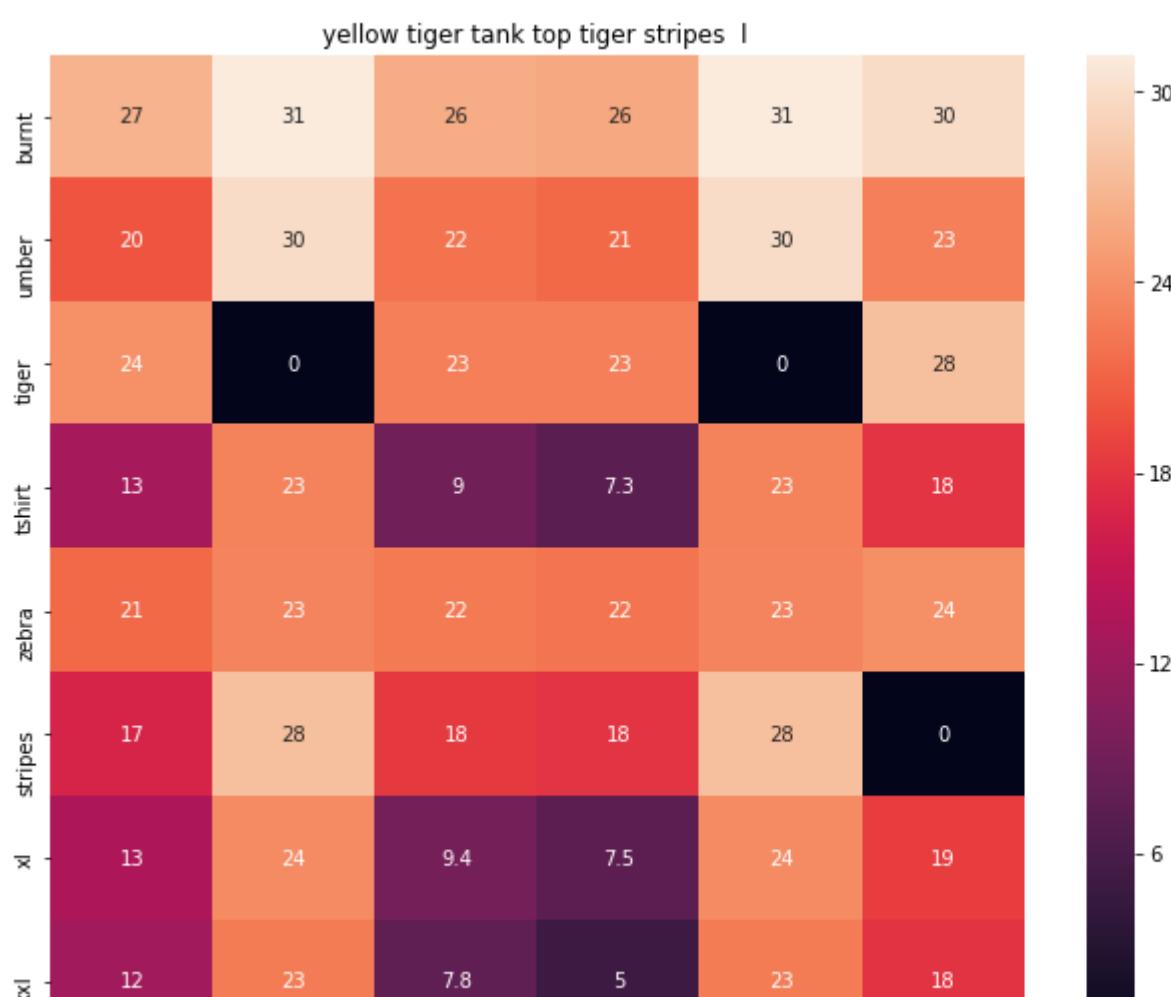




ASIN : B00JXQAFZ2

Brand : Si Row

euclidean distance from input : 5.3601613

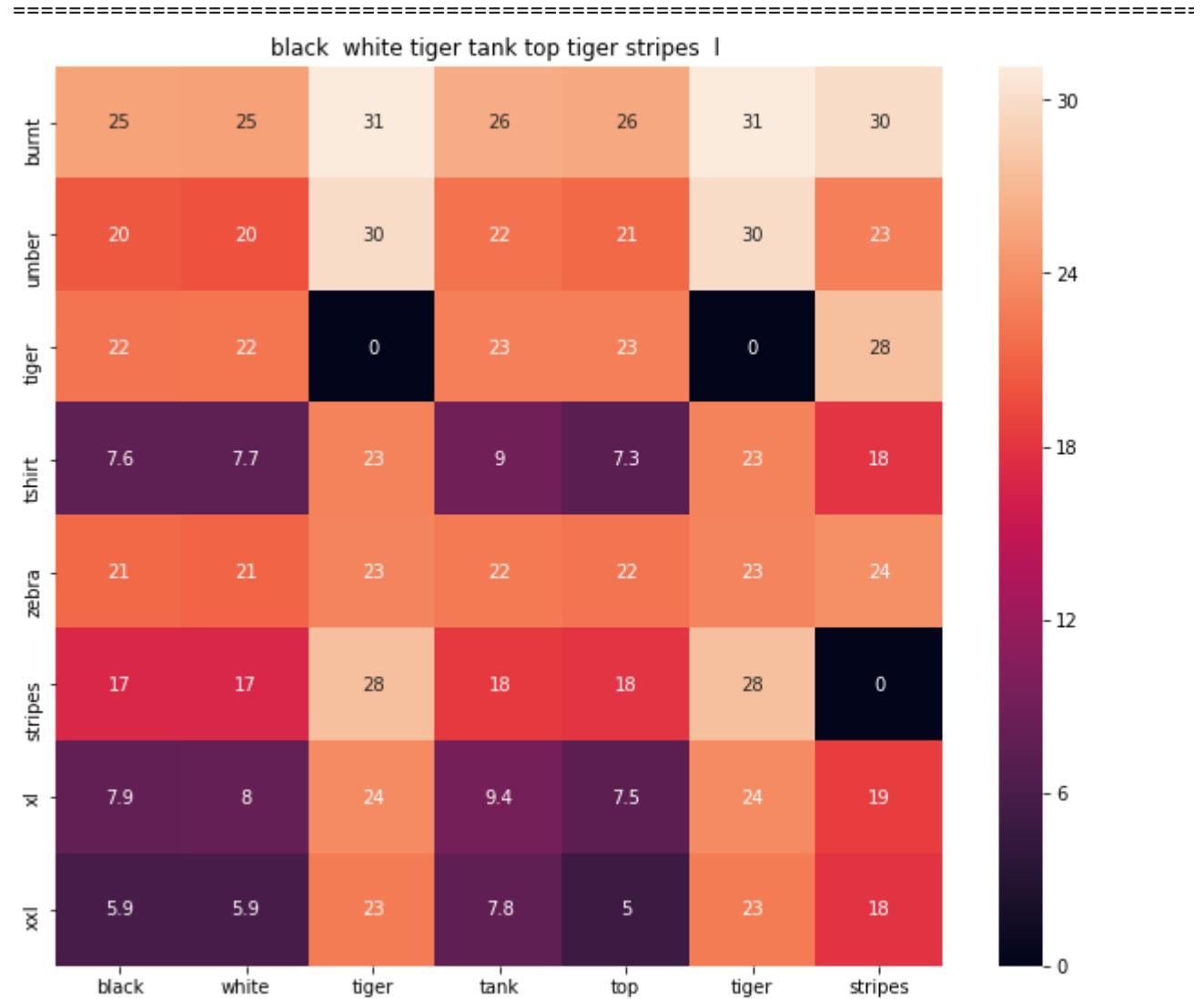




ASIN : B00JXQAUWA

Brand : Si Row

euclidean distance from input : 5.689523

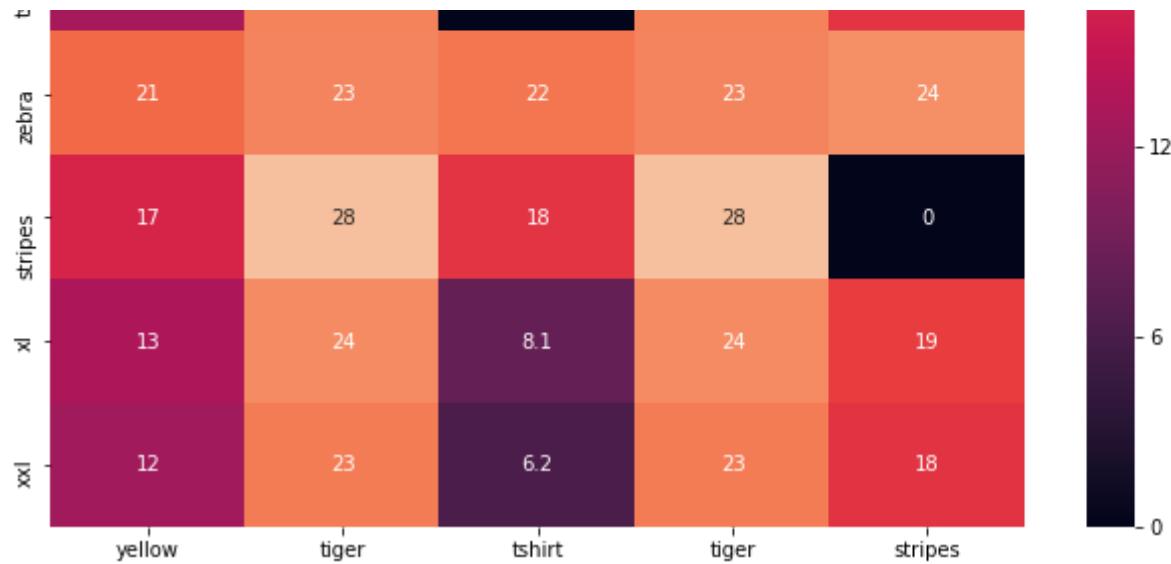


ASIN : B00JXQA094

Brand : Si Row

euclidean distance from input : 5.6930227

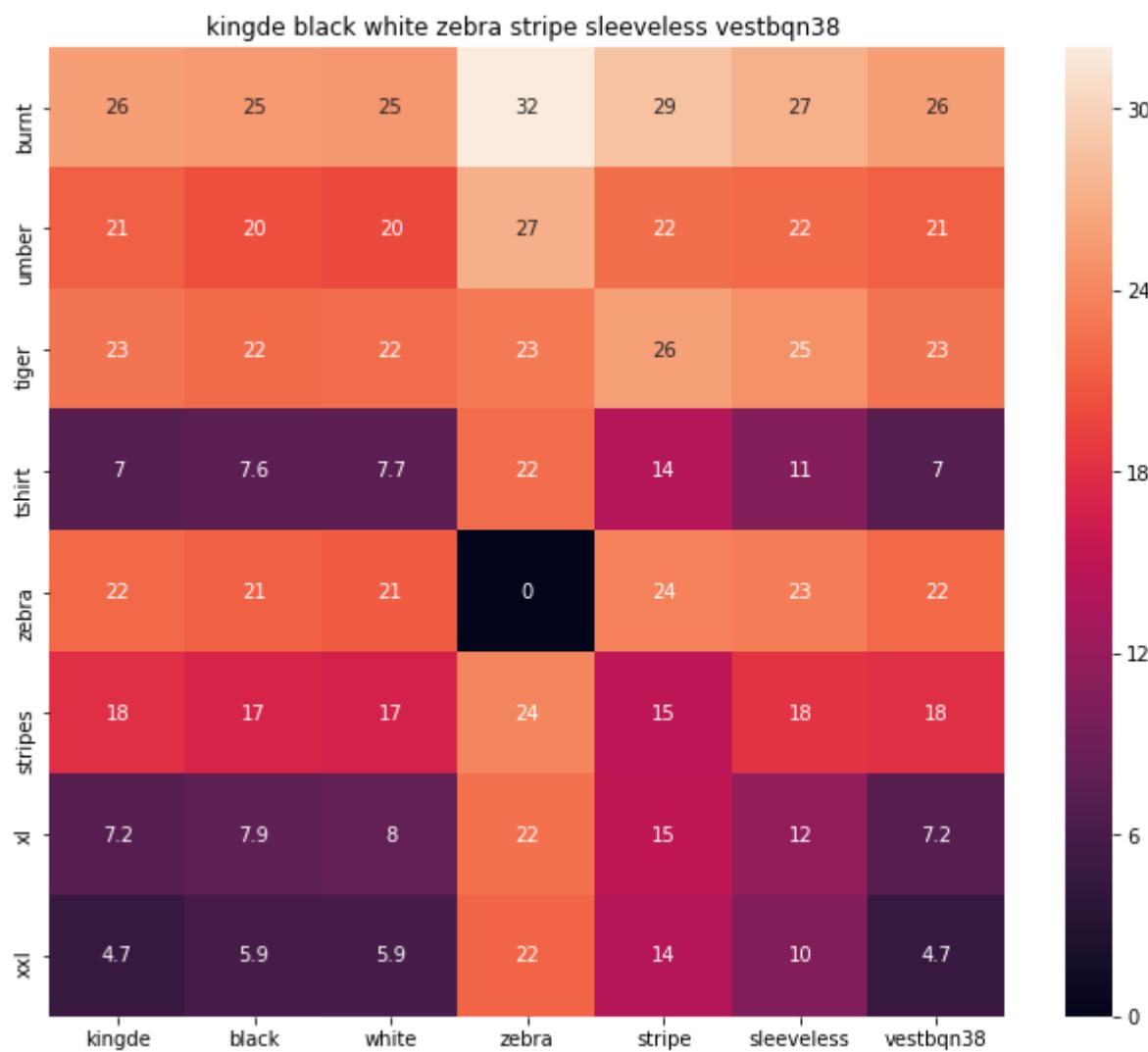




ASIN : B00JXQCUIC

Brand : Si Row

euclidean distance from input : 5.8934426

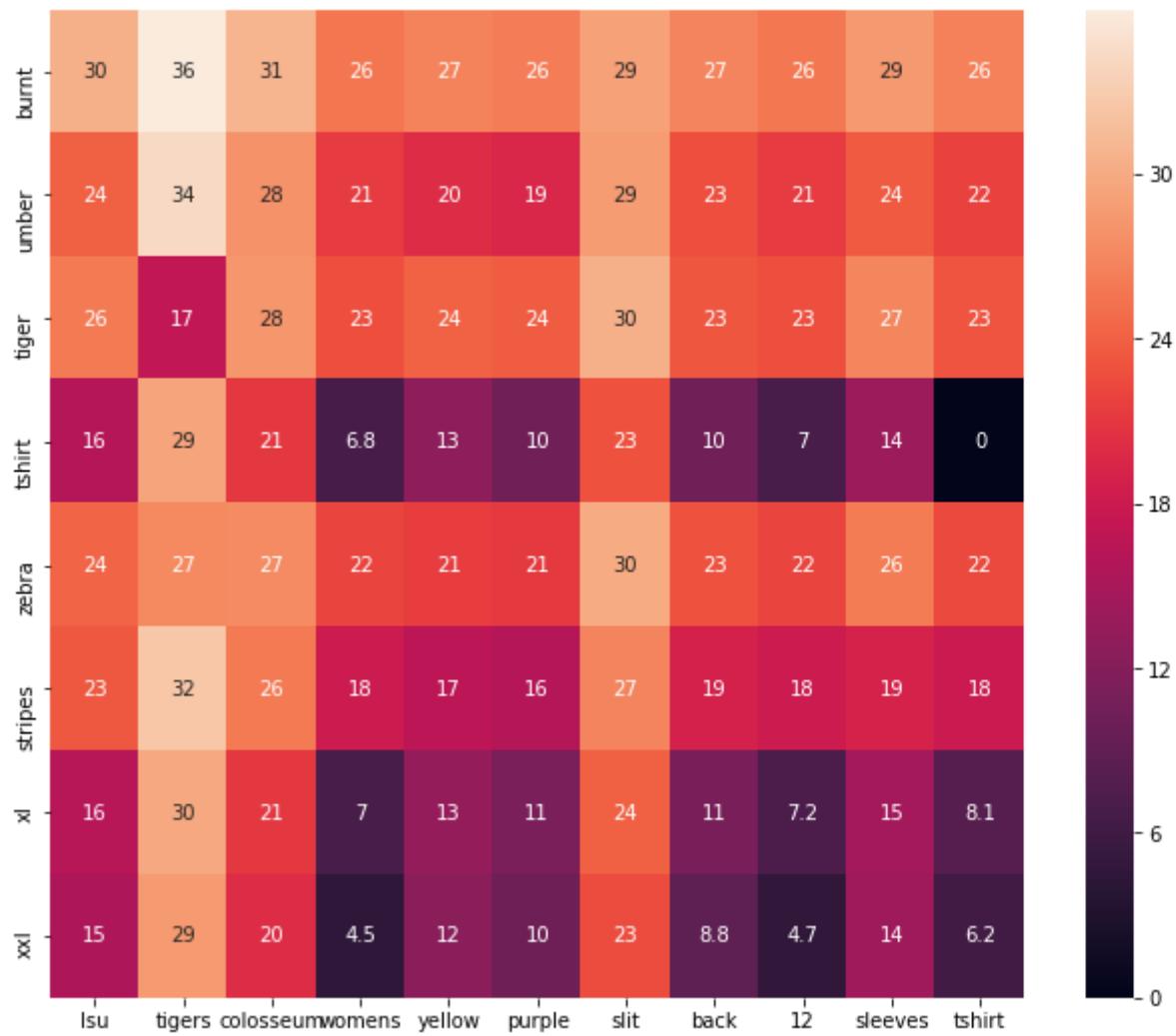


ASIN : B015H41F6G

Brand : KINGDE

euclidean distance from input : 6.13299

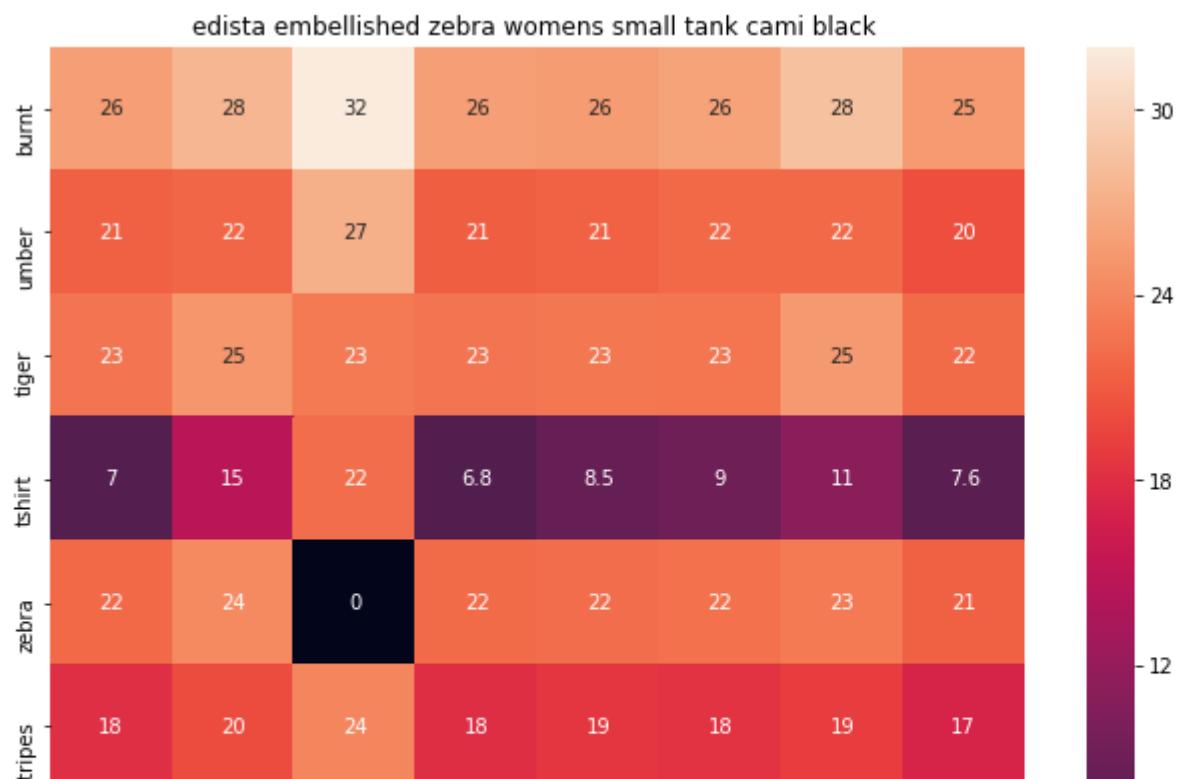
lsu tigers colosseum womens yellow purple slit back 12 sleeves tshirt

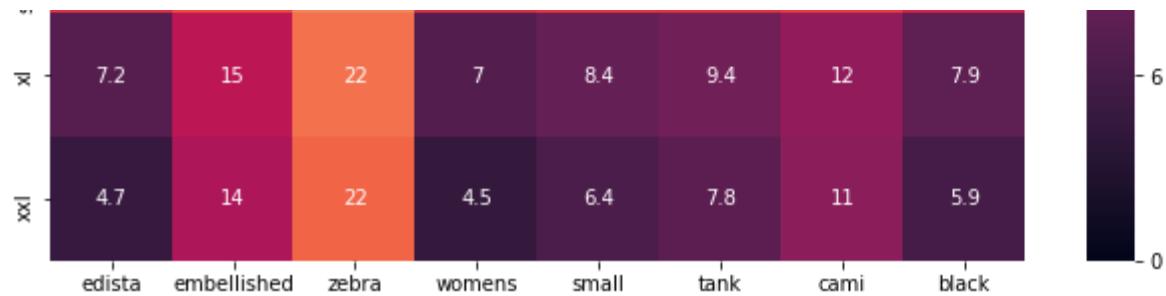


ASIN : B073R5Q8HD

Brand : Colosseum

euclidean distance from input : 6.2567058

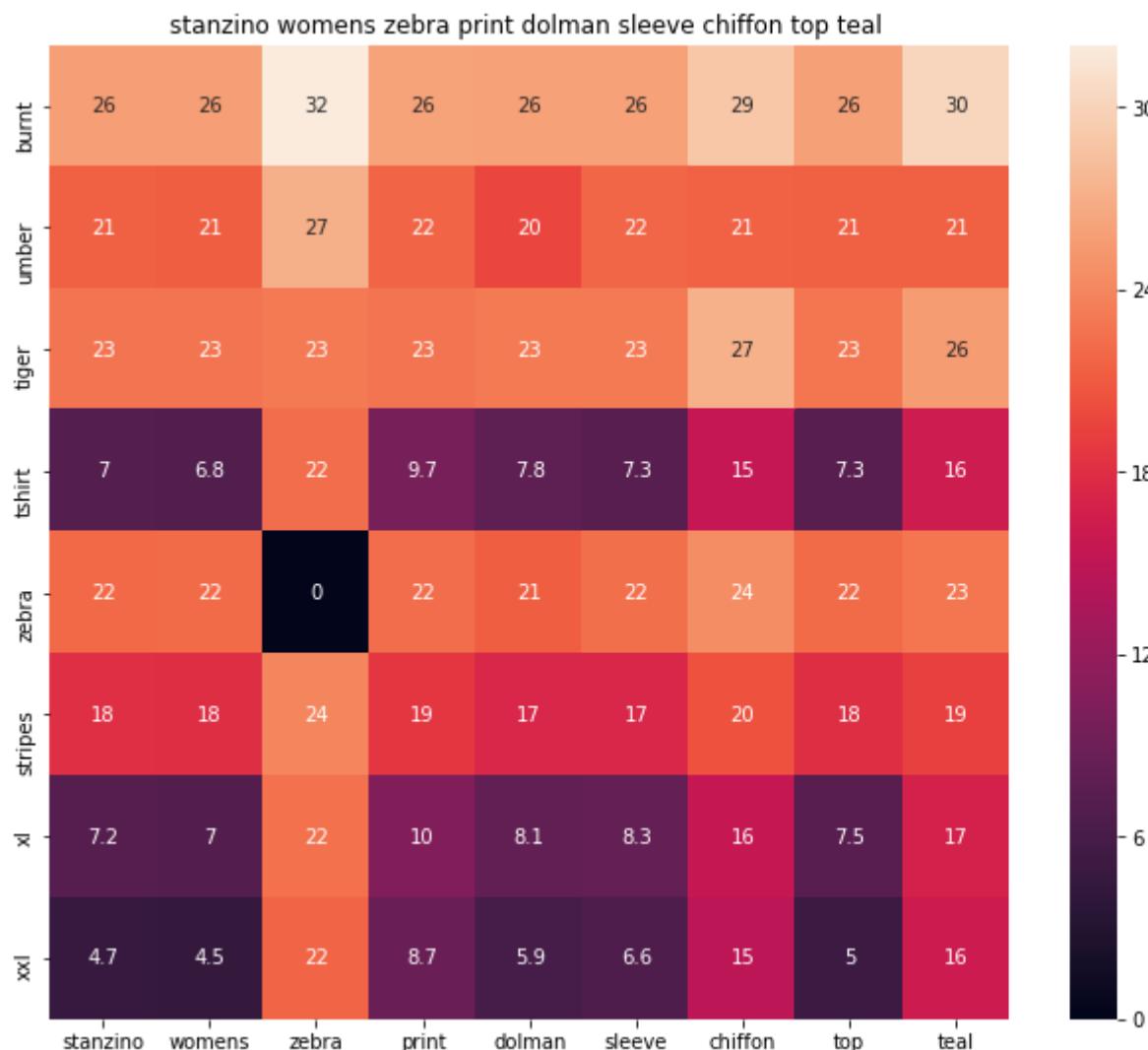




ASIN : B074P8MD22

Brand : Edista

euclidean distance from input : 6.3922043

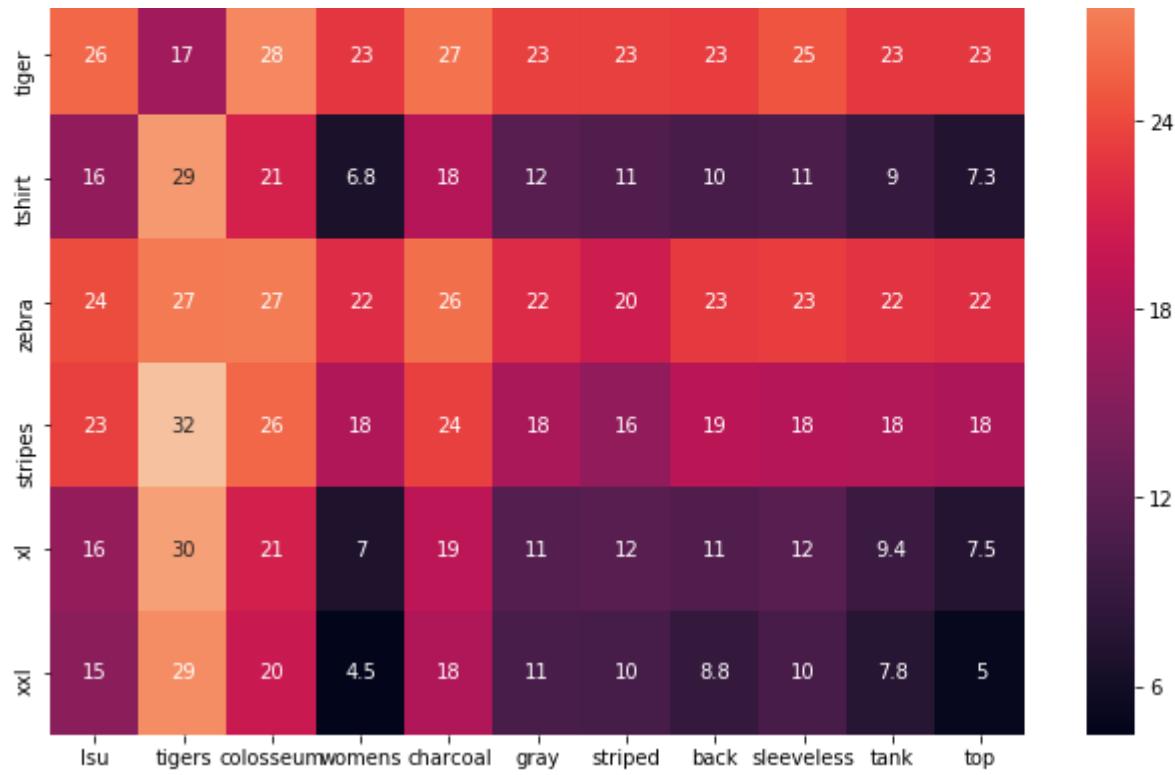


ASIN : B00C0I3U3E

Brand : Stanzino

euclidean distance from input : 6.414901

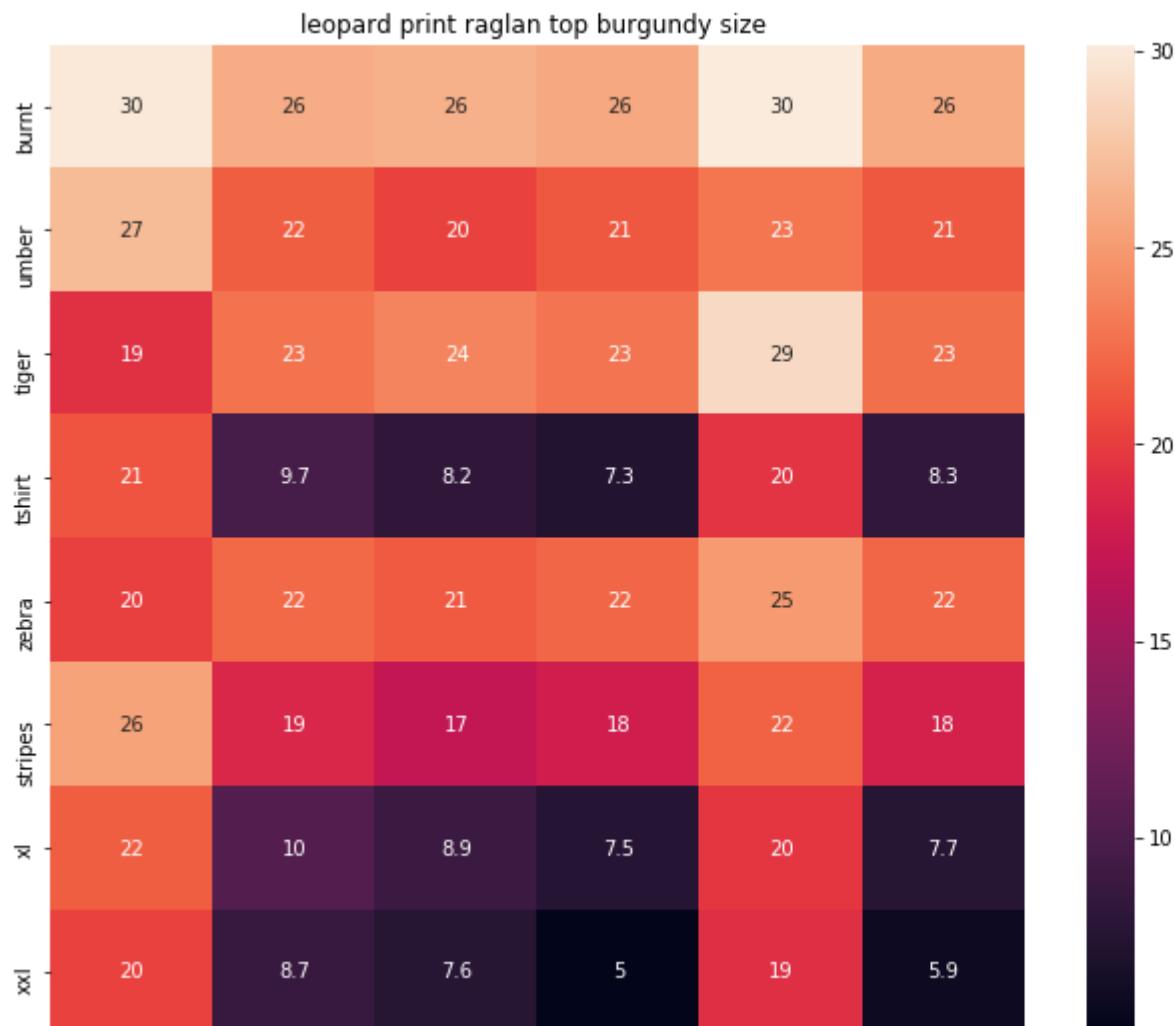




ASIN : B073R4ZM7Y

Brand : Colosseum

euclidean distance from input : 6.4509606

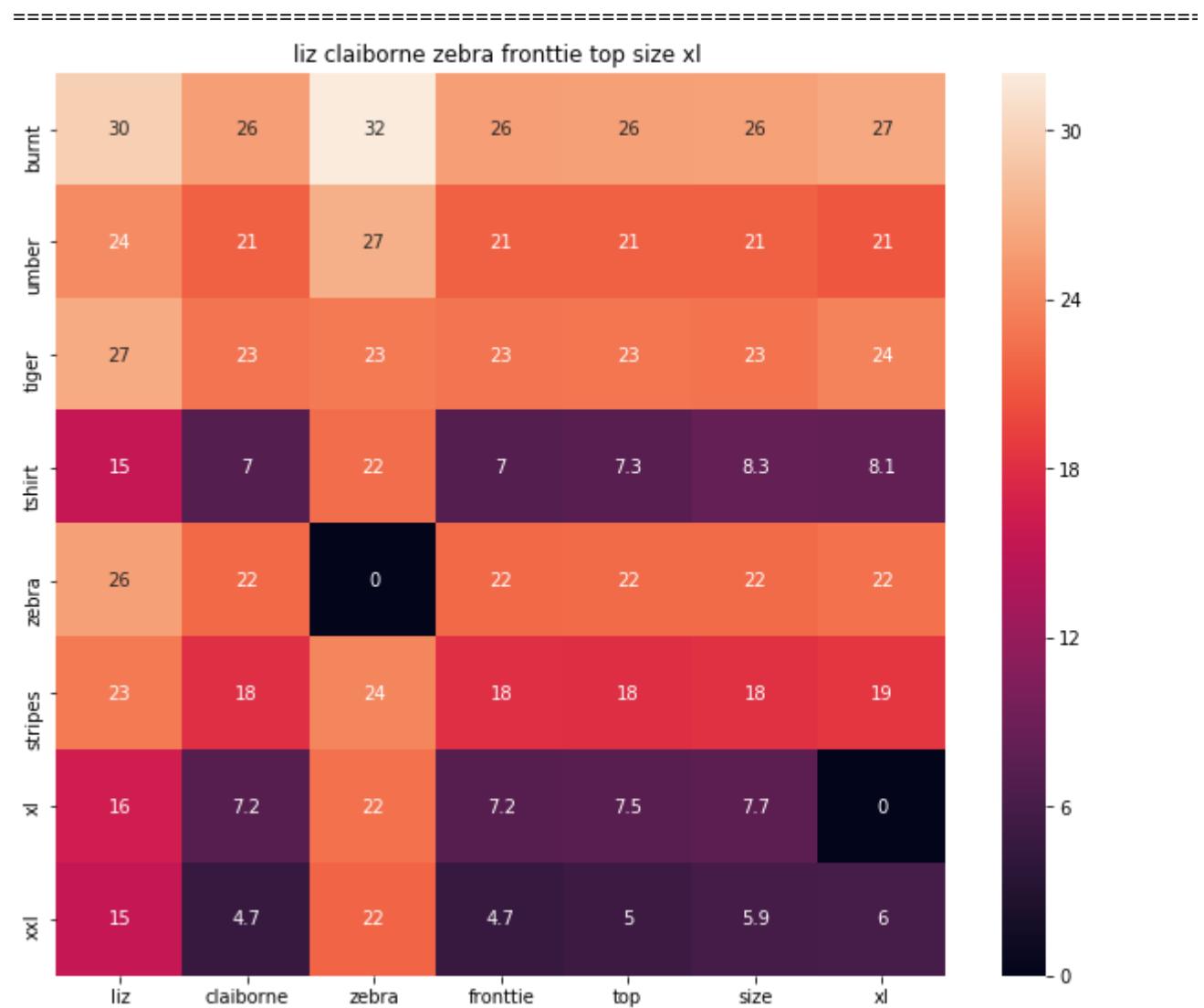


leopard	print	raglan	top	burgundy	size
---------	-------	--------	-----	----------	------

ASIN : B01C60RLDQ

Brand : 1 Mad Fit

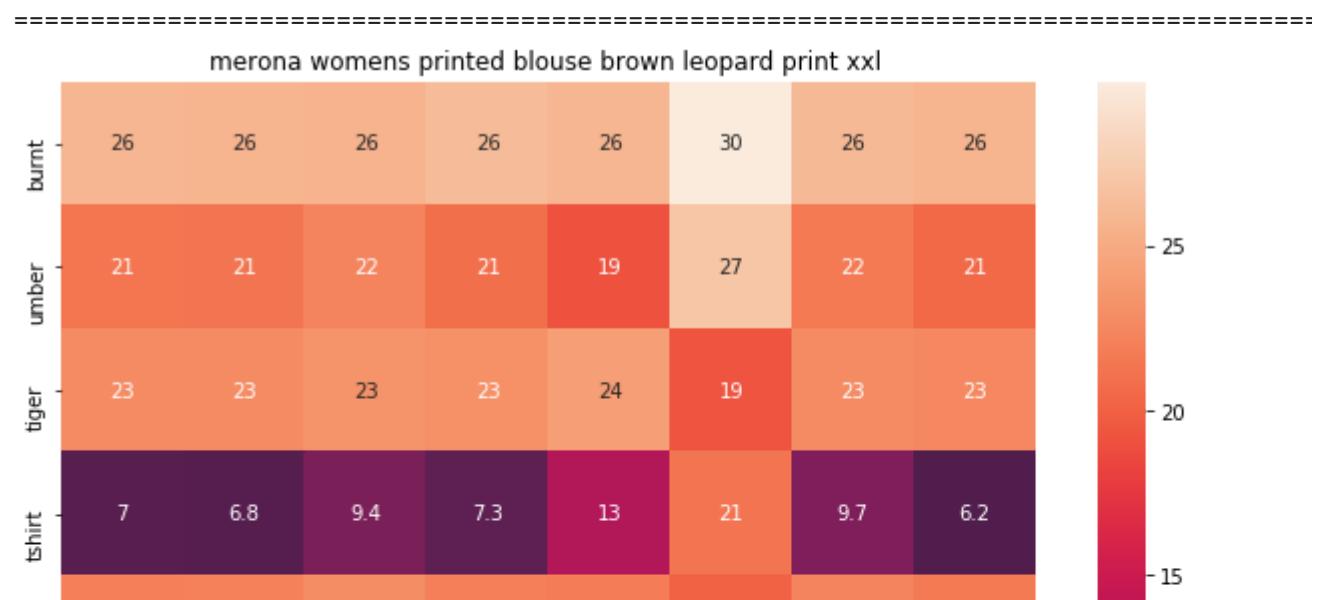
euclidean distance from input : 6.4634094

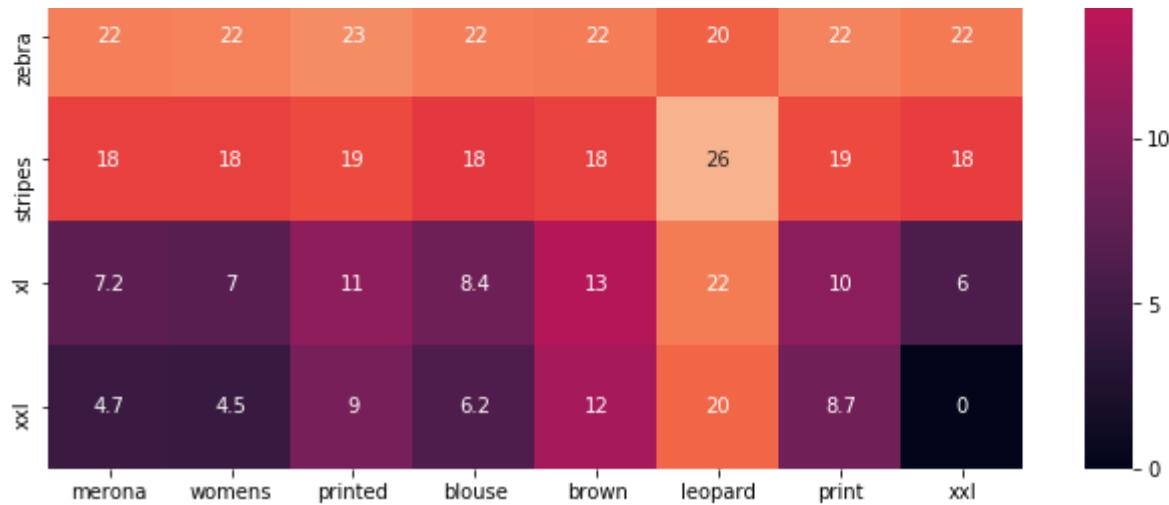


ASIN : B06XBY5QXL

Brand : Liz Claiborne

euclidean distance from input : 6.5392237

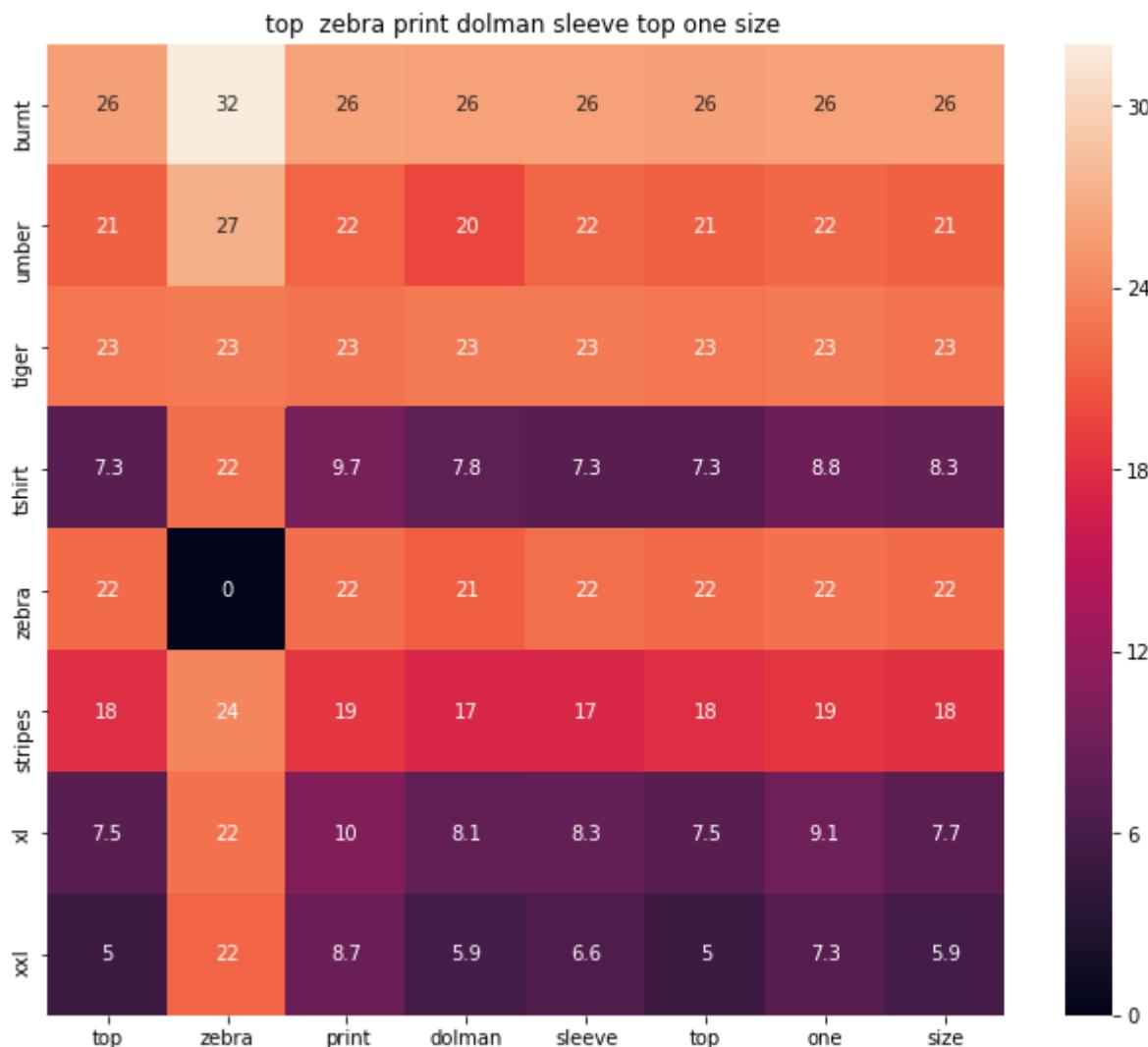




ASIN : B071YF3WDD

Brand : Merona

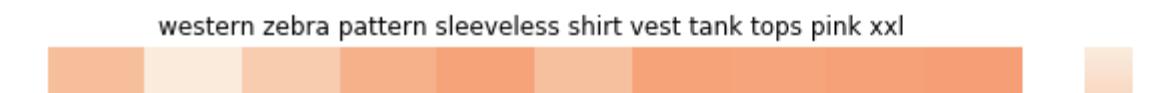
euclidean distance from input : 6.575504

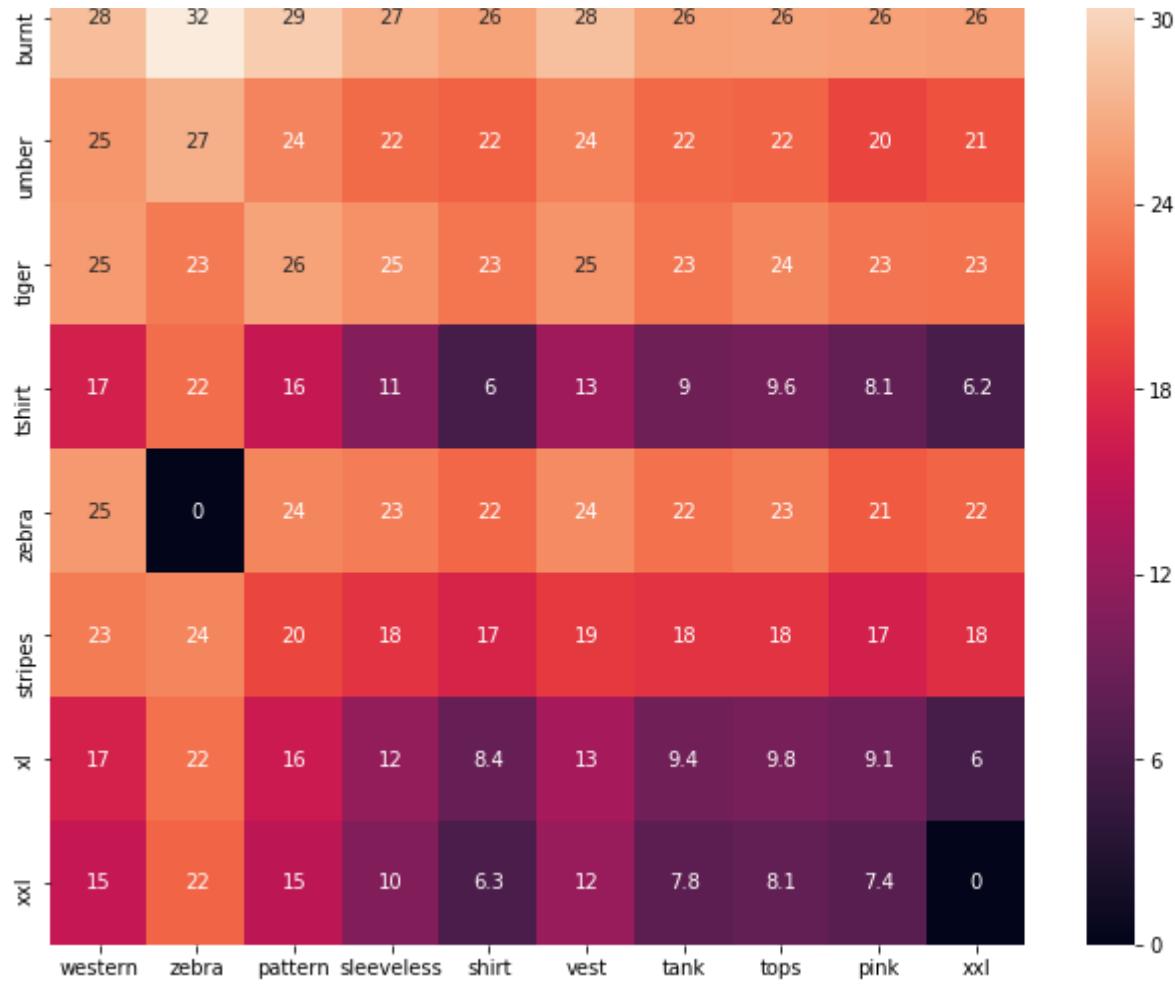


ASIN : B00H8A6ZLI

Brand : Vivian's Fashions

euclidean distance from input : 6.6382155

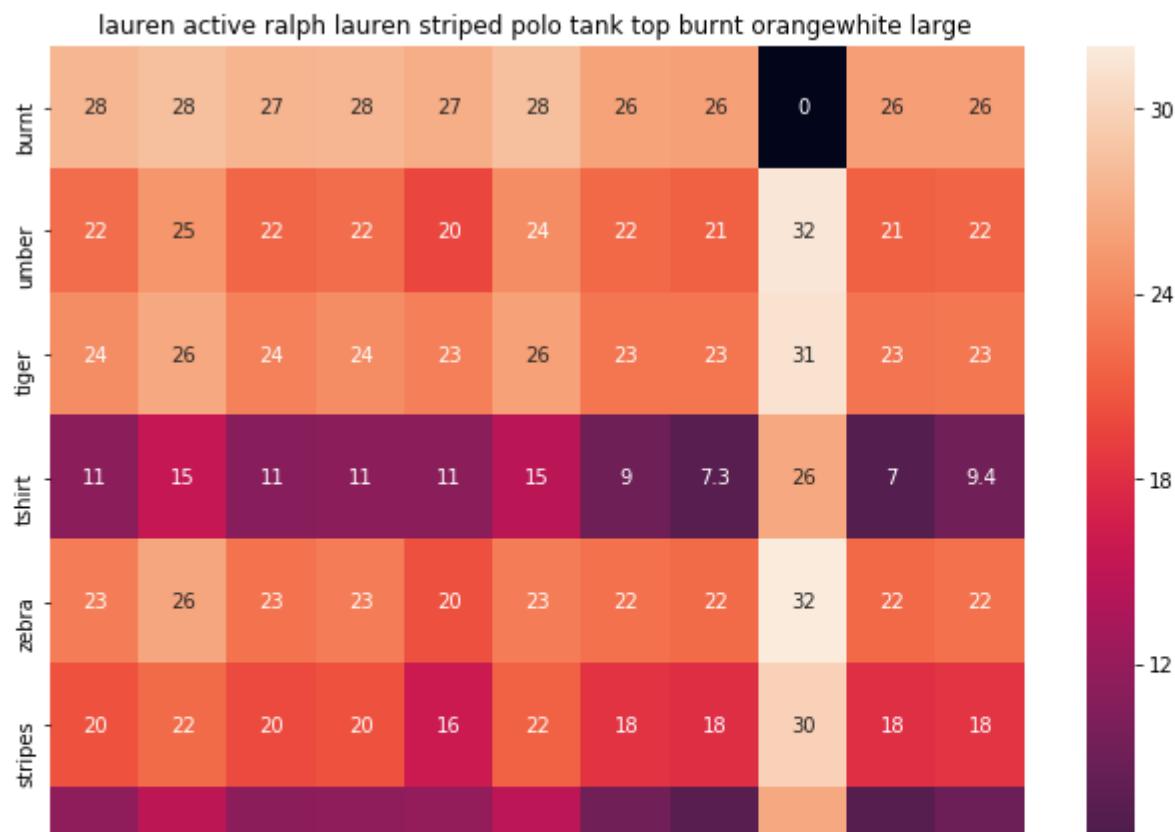


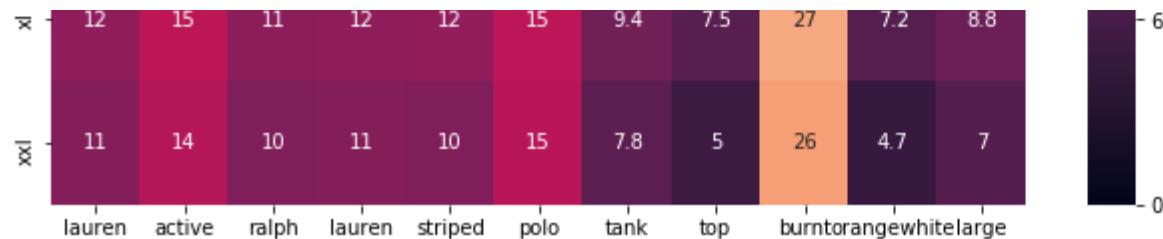


ASIN : B00Z6HEXWI

Brand : Black Temptation

euclidean distance from input : 6.660737





ASIN : B00ILGH50Y

Brand : Ralph Lauren Active

euclidean distance from input : 6.6839056

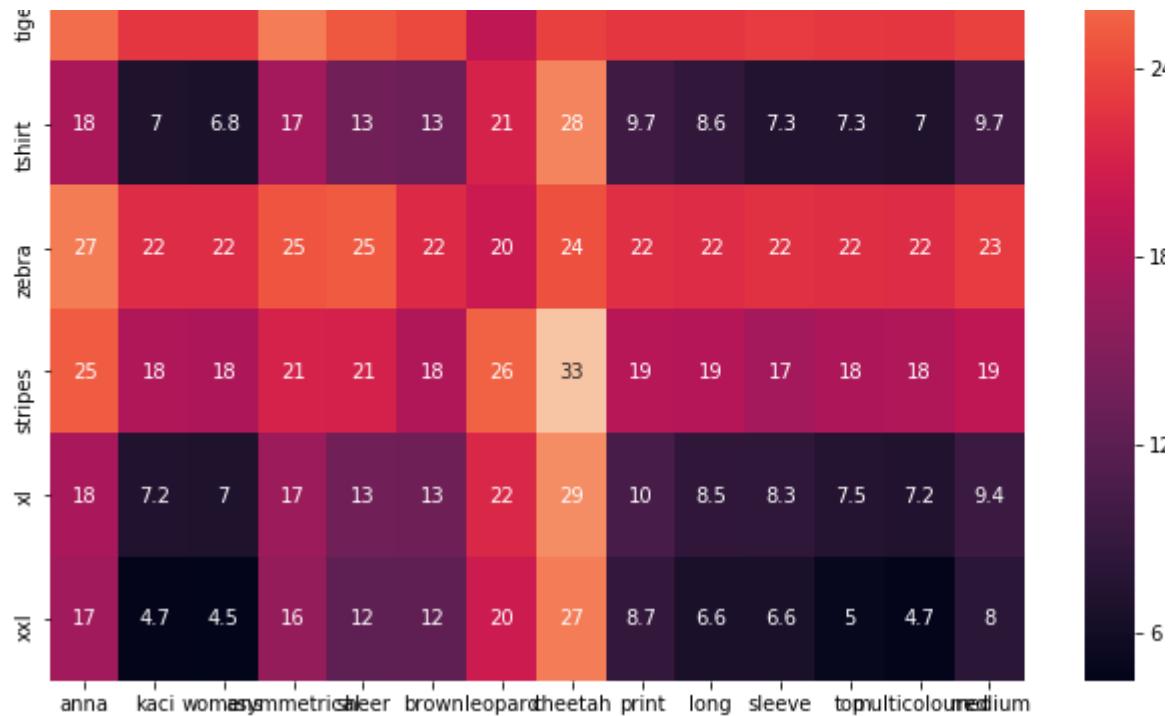


ASIN : B06Y1VN8WQ

Brand : Black Swan

euclidean distance from input : 6.7057643





ASIN : B00KSNTY7Y

Brand : Anna-Kaci

euclidean distance from input : 6.7061253

## ▼ [9.6] Weighted similarity using brand and color

```
# some of the brand values are empty.
# Need to replace Null with string "NULL"
data['brand'].fillna(value="Not given", inplace=True )

# replace spaces with hyphen
brands = [x.replace(" ", "-") for x in data['brand'].values]
types = [x.replace(" ", "-") for x in data['product_type_name'].values]
colors = [x.replace(" ", "-") for x in data['color'].values]

brand_vectorizer = CountVectorizer()
brand_features = brand_vectorizer.fit_transform(brands)

type_vectorizer = CountVectorizer()
type_features = type_vectorizer.fit_transform(types)

color_vectorizer = CountVectorizer()
color_features = color_vectorizer.fit_transform(colors)

extra_features = hstack((brand_features, type_features, color_features)).tocsr()

def heat_map_w2v_brand(sentance1, sentance2, url, doc_id1, doc_id2, df_id1, df_id2, model):

    # sentance1 : title1, input apparel
    # sentance2 : title2, recommended apparel
    # url: apparel image url
    # doc_id1: document id of input apparel
    # doc_id2: document id of recommended apparel
    # df_id1: index of document1 in the data frame
    # df_id2: index of document2 in the data frame
    # model: it can have two values, 1. avg 2. weighted

    #s1_vec = np.array(#number_of_words_title1 * 300), each row is a vector(weighted/avg) of len
    s1_vec = get_word_vec(sentance1, doc_id1, model)
```

```
#s2_vec = np.array(#number_of_words_title2 * 300), each row is a vector(weighted/avg) of len
s2_vec = get_word_vec(sentance2, doc_id2, model)

# s1_s2_dist = np.array(#number of words in title1 * #number of words in title2)
# s1_s2_dist[i,j] = euclidean distance between words i, j
s1_s2_dist = get_distance(s1_vec, s2_vec)

data_matrix = [['Asin', 'Brand', 'Color', 'Product type'],
               [data['asin'].loc[df_id1], brands[doc_id1], colors[doc_id1], types[doc_id1]], # in
               [data['asin'].loc[df_id2], brands[doc_id2], colors[doc_id2], types[doc_id2]]] # re

colorscale = [[0, '#1d004d'], [.5, '#f2e5ff'], [1, '#f2e5d1']] # to color the headings of each

# we create a table with the data_matrix
table = ff.create_table(data_matrix, index=True, colorscale=colorscale)
# plot it with plotly
plotly.offline.iplot(table, filename='simple_table')

# devide whole figure space into 25 * 1:10 grids
gs = gridspec.GridSpec(25, 15)
fig = plt.figure(figsize=(25,5))

# in first 25*10 grids we plot heatmap
ax1 = plt.subplot(gs[:, :-5])
# plotting the heatmap based on the pairwise distances
ax1 = sns.heatmap(np.round(s1_s2_dist,6), annot=True)
# set the x axis labels as recommended apparels title
ax1.set_xticklabels(sentance2.split())
# set the y axis labels as input apparels title
ax1.set_yticklabels(sentance1.split())
# set title as recommended apparels title
ax1.set_title(sentance2)

# in last 25 * 10:15 grids we display image
ax2 = plt.subplot(gs[:, 10:16])
# we dont display grid lines and axis labels to images
ax2.grid(False)
ax2.set_xticks([])
ax2.set_yticks([])

# pass the url it display it
display_img(url, ax2, fig)

plt.show()

def idf_w2v_brand(doc_id, w1, w2, num_results):
    # doc_id: apparel's id in given corpus
    # w1: weight for w2v features
    # w2: weight for brand and color features

    # pairwise_dist will store the distance from given input apparel to all remaining apparels
    # the metric we used here is cosine, the cosine distance is measured as K(X, Y) = <X, Y> / (|X| * |Y|)
    # http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
    idf_w2v_dist = pairwise_distances(w2v_title_weight, w2v_title_weight[doc_id].reshape(1,-1))
    ex_feat_dist = pairwise_distances(extra_features, extra_features[doc_id])
    pairwise_dist = (w1 * idf_w2v_dist + w2 * ex_feat_dist)/float(w1 + w2)

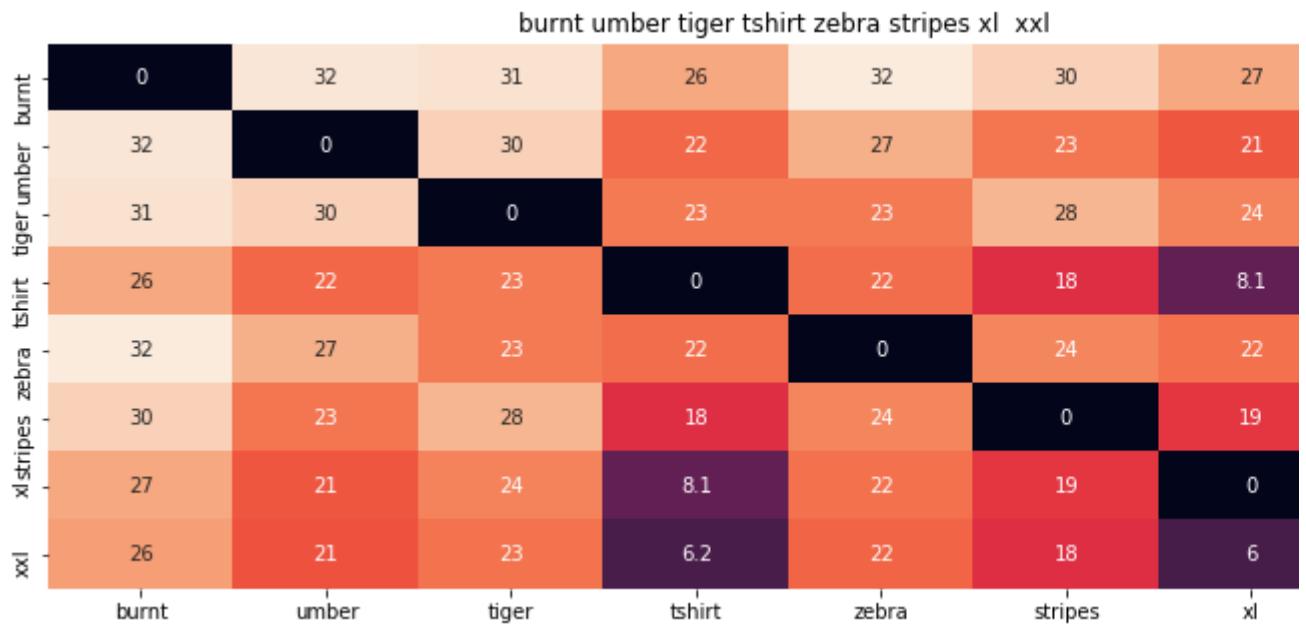
    # np.argsort will return indices of 9 smallest distances
    indices = np.argsort(pairwise_dist.flatten())[0:num_results]
    # pdists will store the 9 smallest distances
    pdists = np.sort(pairwise_dist.flatten())[0:num_results]

    # data frame indices of the 9 smallest distance's
    df_indices = list(data.index[indices])

    for i in range(0, len(indices)):
```

```
heat_map_w2v_brand(data['title'].loc[df_indices[0]],data['title'].loc[df_indices[i]], da  
print('ASIN :',data['asin'].loc[df_indices[i]])  
print('Brand :',data['brand'].loc[df_indices[i]])  
print('euclidean distance from input :', pdists[i])  
print('*125)  
  
idf_w2v_brand(12566, 5, 5, 20)  
# in the give heat map, each cell contains the euclidean distance between words i, j
```

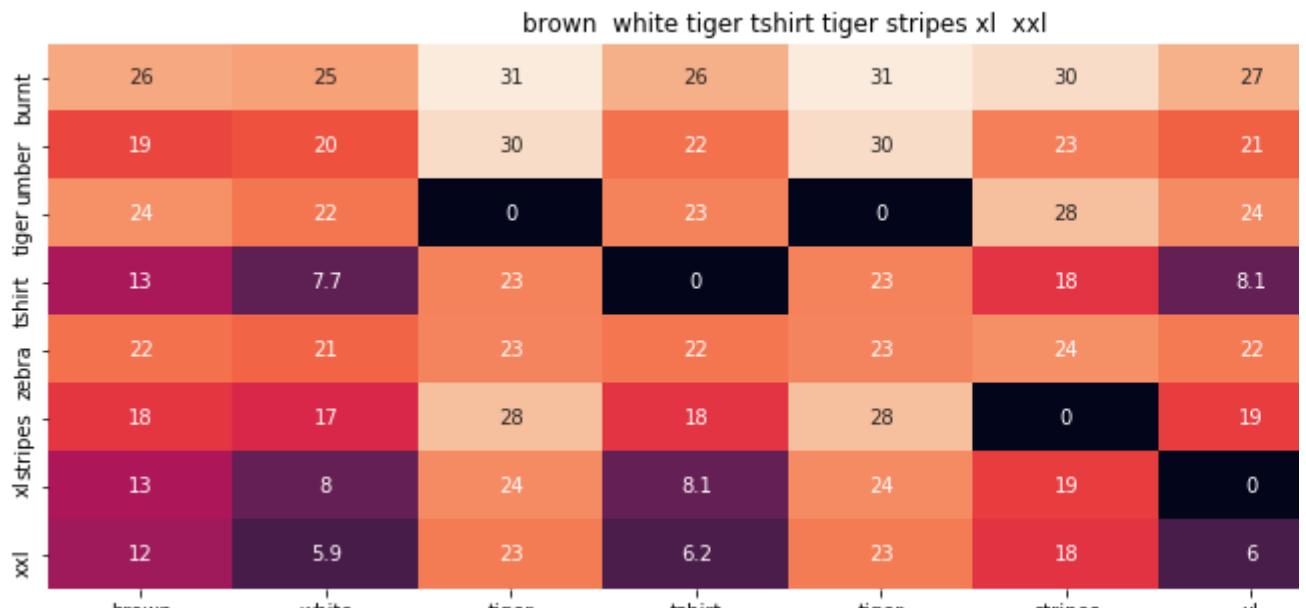




ASIN : B00JXQB5FQ

Brand : Si Row

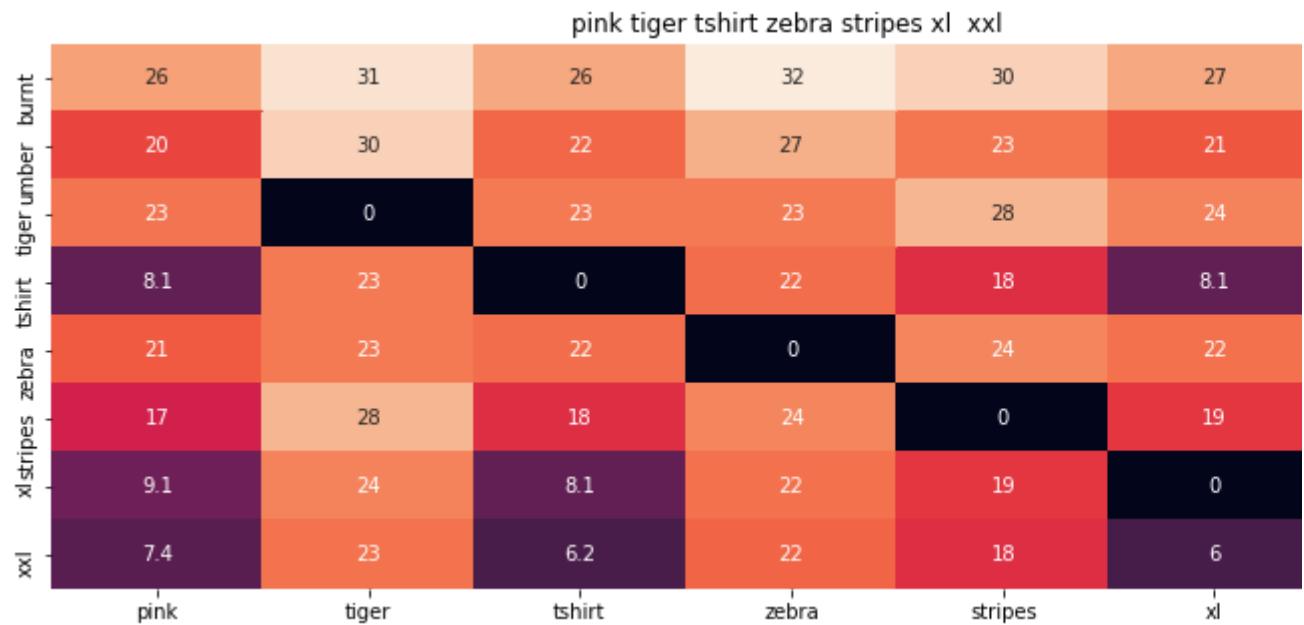
euclidean distance from input : 0.001953125



ASIN : B00JXQCWT0

Brand : Si Row

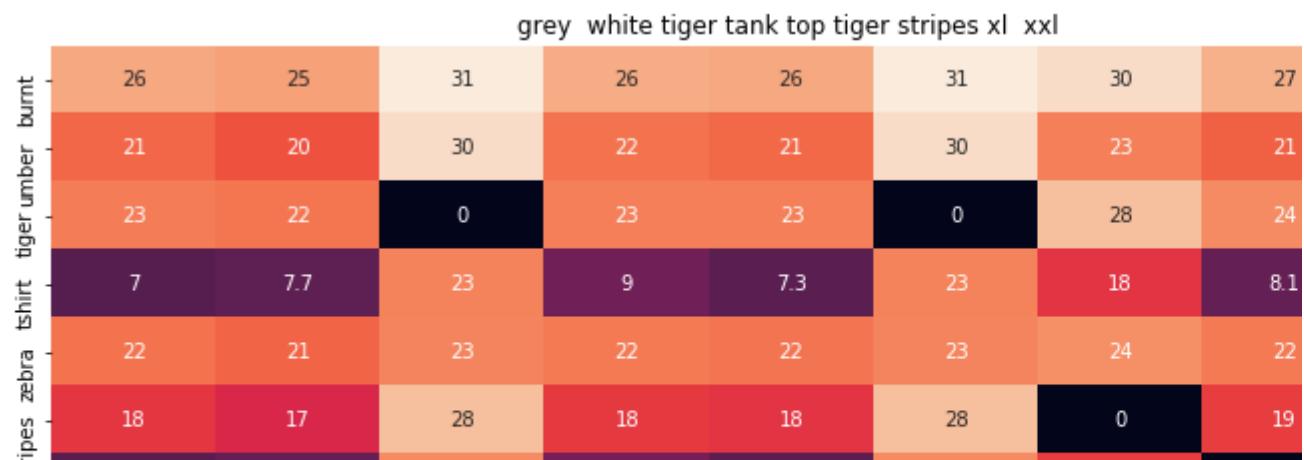
euclidean distance from input : 2.385471153259277



ASIN : B00JXQASS6

Brand : Si Row

euclidean distance from input : 2.7390506746191647



xxl	7.2	8	24	9.4	7.5	24	19	0
xxl	4.7	5.9	23	7.8	5	23	18	6
	grey	white	tiger	tank	top	tiger	stripes	xl

ASIN : B00JXQAFZ2

Brand : Si Row

euclidean distance from input : 3.3871873857397703

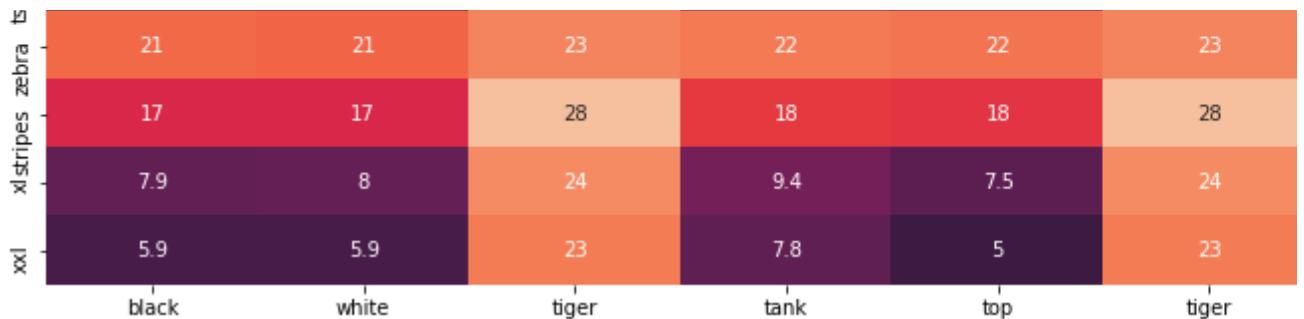
	yellow	tiger	tank	top	stripes	l
burnt	27	31	26	26	31	
umber	20	30	22	21	30	
tiger	24	0	23	23	0	
tshirt	13	23	9	7.3	23	
zebra	21	23	22	22	23	
stripes	17	28	18	18	28	
xl	13	24	9.4	7.5	24	
xxl	12	23	7.8	5	23	
	yellow	tiger	tank	top	tiger	

ASIN : B00JXQAUWA

Brand : Si Row

euclidean distance from input : 3.5518684389013915

	black	white	tiger	tank	top	tiger	stripes	l
burnt	25	25	31	26	26	31		
umber	20	20	30	22	21	30		
tiger	22	22	0	23	23	0		
tshirt	7.6	7.7	23	9	7.3	23		
	black	white	tiger	tank	top	tiger	stripes	

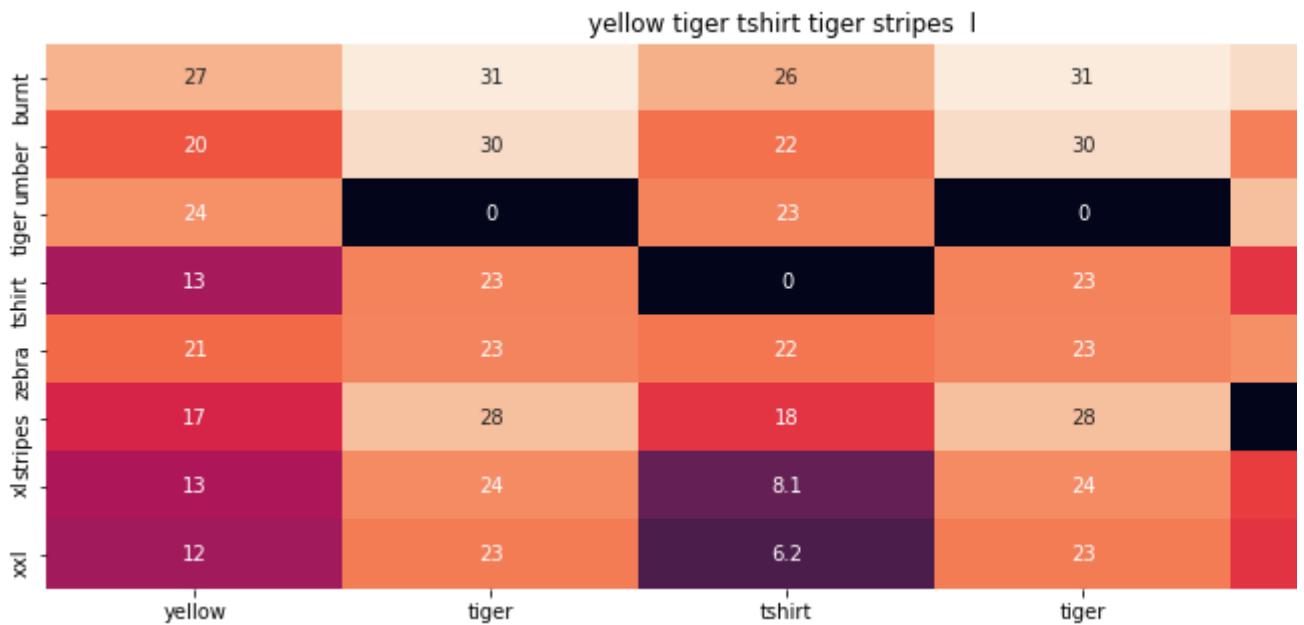


ASIN : B00JXQA094

Brand : Si Row

euclidean distance from input : 3.5536182405371335

=====

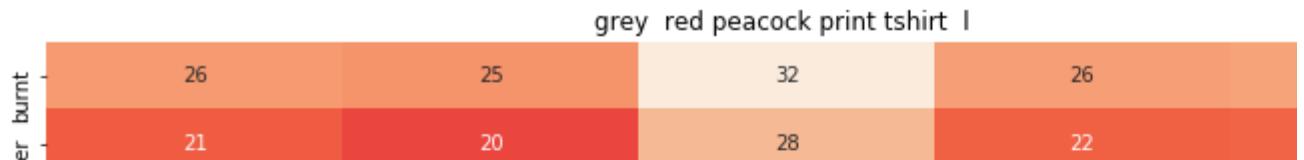


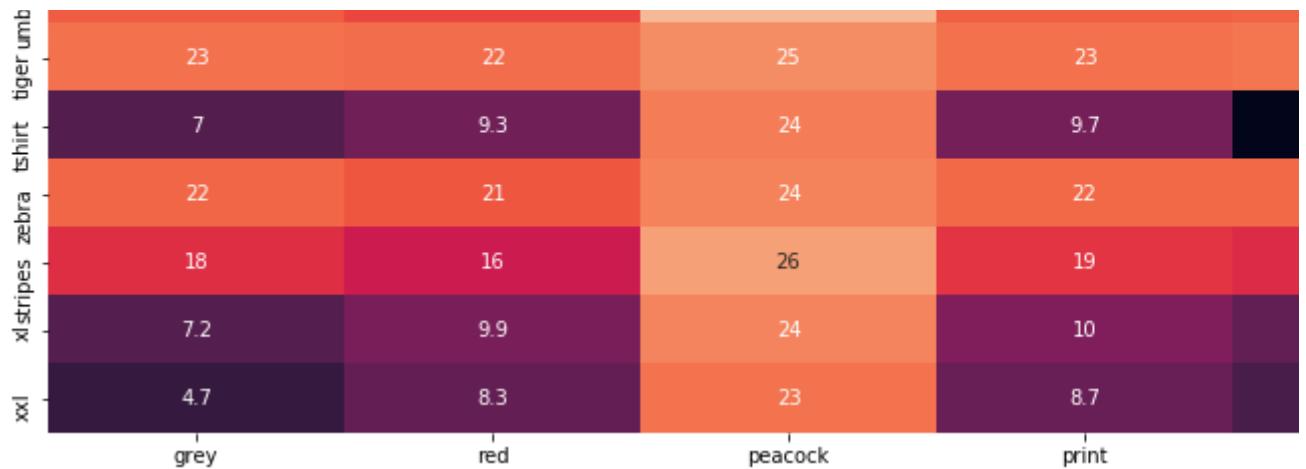
ASIN : B00JXQCUIC

Brand : Si Row

euclidean distance from input : 3.653828048886743

=====

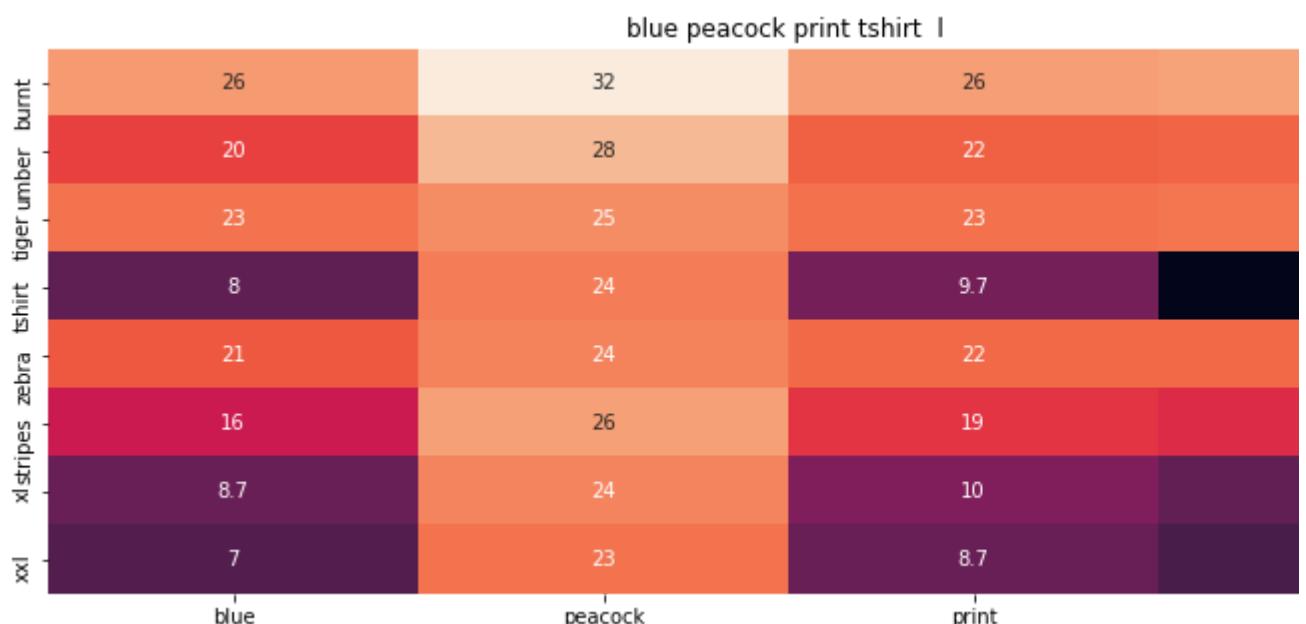




ASIN : B00JXQCFRS

Brand : Si Row

euclidean distance from input : 4.128811645688501



ASIN : B00JXQC8L6

Brand : Si Row

euclidean distance from input : 4.203900146665063

	red	butterfly	black	white	tank	top	xl
xxl	8.3	21	5.9	5.9	7.8	5	6
xl	9.9	22	7.9	8	9.4	7.5	0
l	16	26	17	17	18	18	19
m	21	26	21	21	22	22	22
s	9.3	21	7.6	7.7	9	7.3	8.1
xsm	22	27	22	22	23	23	24
sm	20	27	20	20	22	21	21
md	25	33	25	25	26	26	27
lg							
xxl							

ASIN : B00JV63CW2

Brand : Si Row

euclidean distance from input : 4.286586761655298

=====

kingde black white zebra stripe sleeveless vestbqn38

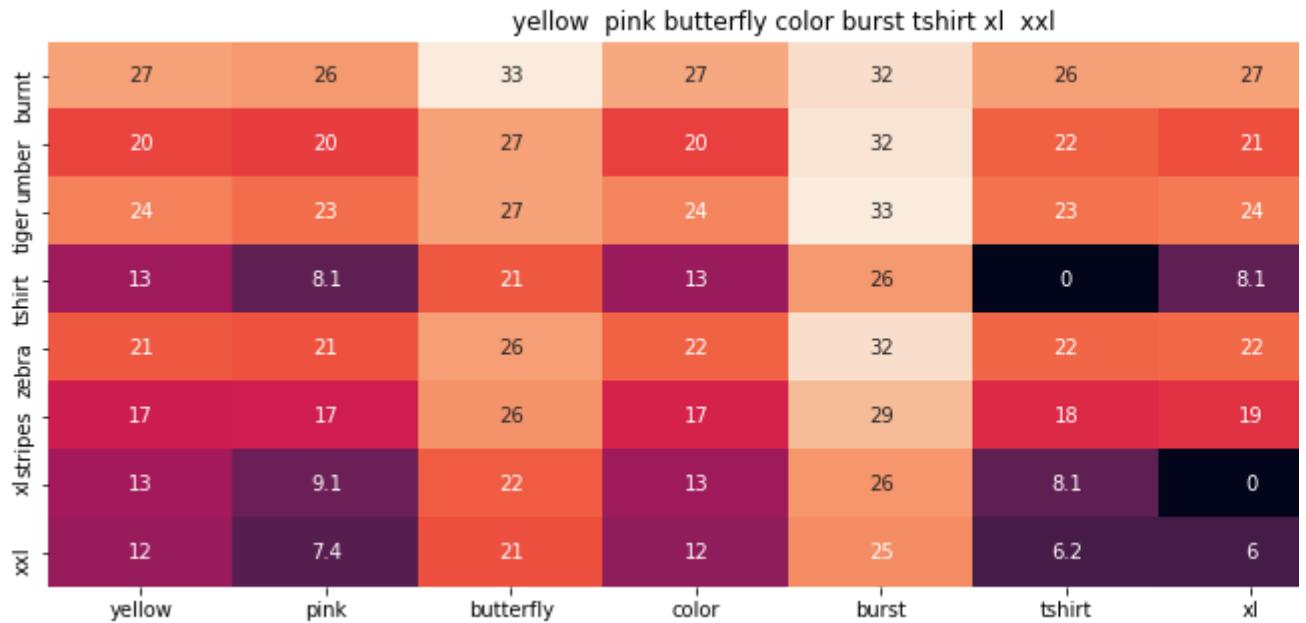
	kingde	black	white	zebra	stripe	sleeveless
xxl	4.7	5.9	5.9	22	14	10
xl	7.2	7.9	8	22	15	12
l	18	17	17	24	15	18
m	22	21	21	0	24	23
s	7	7.6	7.7	22	14	11
xsm	23	22	22	23	26	25
sm	21	20	20	27	22	22
md	26	25	25	32	29	27
lg						
xxl						

ASIN : B015H41F6G

Brand : KINGDE

euclidean distance from input : 4.389370597243721

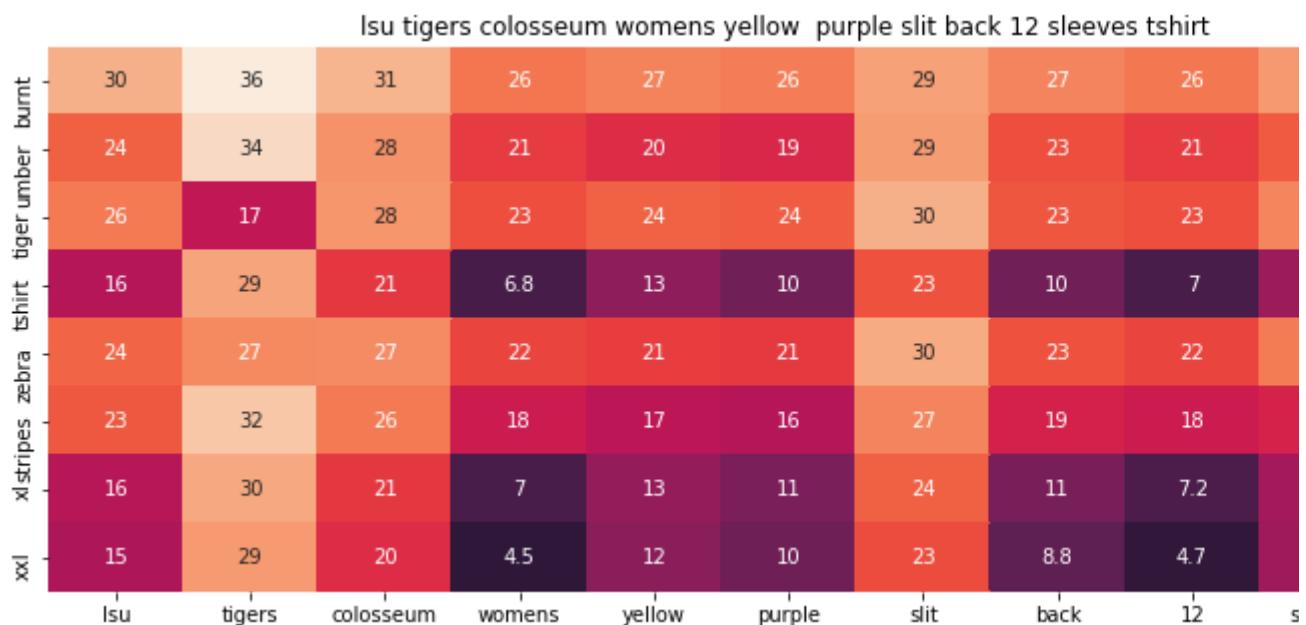
=====



ASIN : B00JXQBBMI

Brand : Si Row

euclidean distance from input : 4.397909927548852



ASIN : B073R5Q8HD

Brand : Colosseum

euclidean distance from input : 4.451228583693917

	edista	embellished	zebra	womens	small	tank	cami
edista	26	28	32	26	26	26	28
tiger umber	21	22	27	21	21	22	22
tiger	23	25	23	23	23	23	25
tshirt	7	15	22	6.8	8.5	9	11
zebra	22	24	0	22	22	22	23
xl stripes	18	20	24	18	19	18	19
xxl	7.2	15	22	7	8.4	9.4	12
xxl	4.7	14	22	4.5	6.4	7.8	11

ASIN : B074P8MD22

Brand : Edista

euclidean distance from input : 4.518977797866279

	red	pink	floral	heel	sleeveless	shirt	xxl
red	25	26	28	35	27	26	27
tiger umber	20	20	21	31	22	22	21
tiger	22	23	24	32	25	23	24
tshirt	9.3	8.1	12	24	11	6	8.1
zebra	21	21	23	29	23	22	22
xl stripes	16	17	19	27	18	17	19
xxl	9.9	9.1	12	25	12	8.4	0
xxl	8.3	7.4	11	24	10	6.3	6

ASIN : B00JV63QOE

Brand : Si Row

euclidean distance from input : 4.52937545794436

=====

stanzino womens zebra print dolman sleeve chiffon top teal

	stanzino	womens	zebra	print	dolman	sleeve	chiffon	top
burnt	26	26	32	26	26	26	29	26
tiger	21	21	27	22	20	22	21	21
umbre	23	23	23	23	23	23	27	23
tshirt	7	6.8	22	9.7	7.8	7.3	15	7.3
zebra	22	22	0	22	21	22	24	22
stripes	18	18	24	19	17	17	20	18
xl	7.2	7	22	10	8.1	8.3	16	7.5
xxl	4.7	4.5	22	8.7	5.9	6.6	15	5

ASIN : B00C0I3U3E

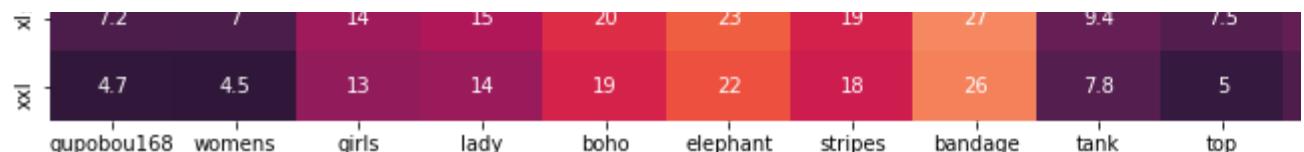
Brand : Stanzino

euclidean distance from input : 4.530326140761788

=====

gupobou168 womens girls lady boho elephant stripes bandage tank top one size

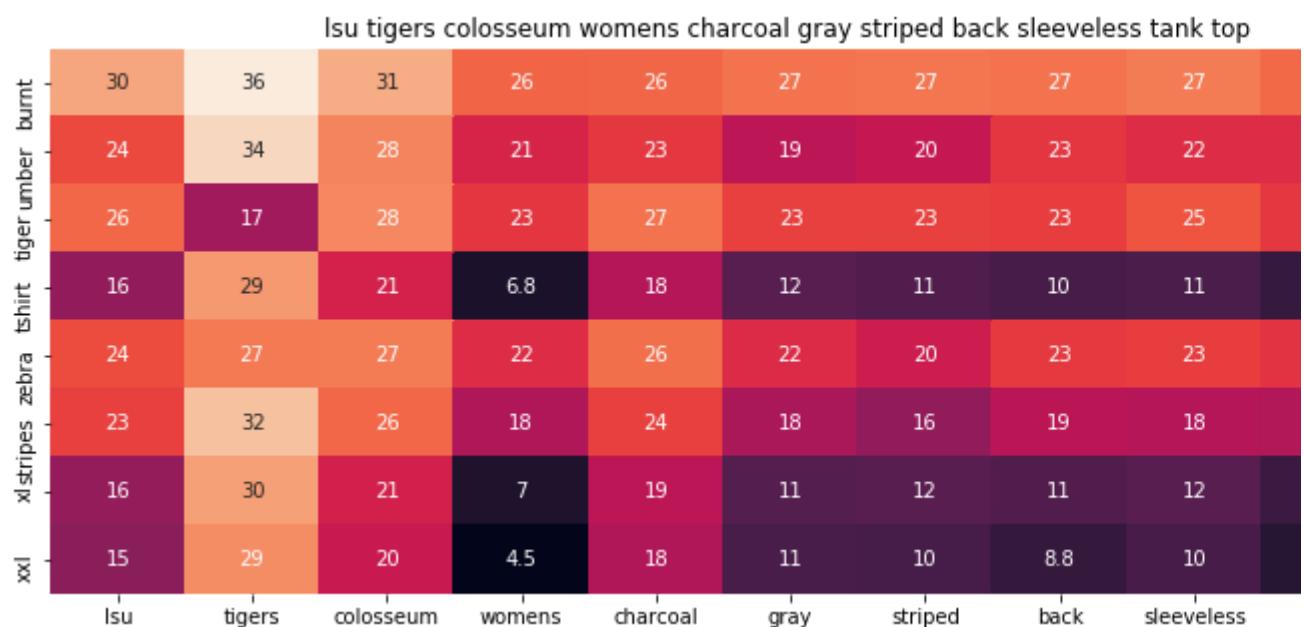
	26	26	28	28	31	32	30	35	26	26
burnt	26	26	28	28	31	32	30	35	26	26
tiger	21	21	24	24	25	29	23	32	22	21
umber	23	23	25	25	29	18	28	33	23	23
tshirt	7	6.8	14	15	19	23	18	25	9	7.3
zebra	22	22	25	24	27	22	24	32	22	22
stripes	18	18	22	23	24	28	0	28	18	18
xxl	7.2	7	14	15	20	22	18	27	8.4	7.5



ASIN : B01ER18406

Brand : GuPoBoU168

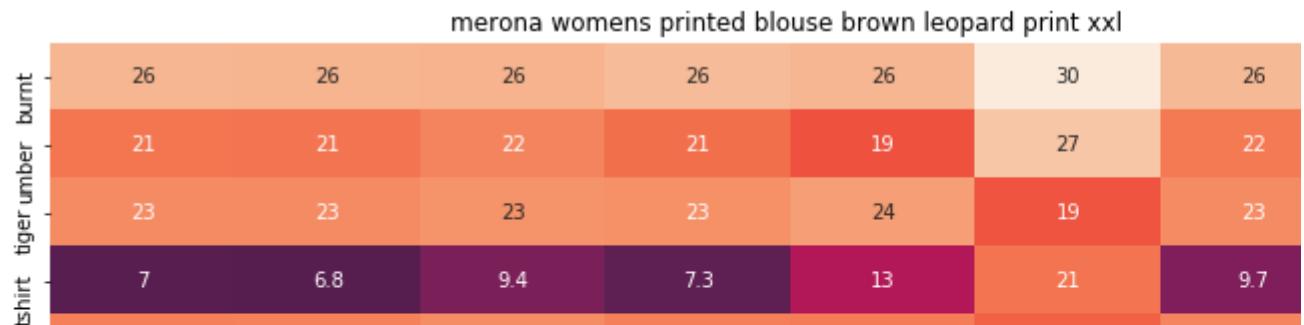
euclidean distance from input : 4.546817024028215



ASIN : B073R4ZM7Y

Brand : Colosseum

euclidean distance from input : 4.548355925918037



xxl	zebra	22	22	23	22	22	20	22
xl	stripes	18	18	19	18	18	26	19
l	merona	7.2	7	11	8.4	13	22	10
s	womens	4.7	4.5	9	6.2	12	20	8.7
	printed							
	blouse							
	brown							
	leopard							
	print							

ASIN : B071YF3WDD

Brand : Merona

euclidean distance from input : 4.610627425551827

=====

		leopard	print	raglan	top	burgundy	size
xxl	burnt	30	26	26	26	30	
xl	tiger	27	22	20	21	23	
l	umber	19	23	24	23	29	
s	tshirt	21	9.7	8.2	7.3	20	
	zebra	20	22	21	22	25	
	stripes	26	19	17	18	22	
	xl	22	10	8.9	7.5	20	
	xxl	20	8.7	7.6	5	19	
	leopard						
	print						
	raglan						
	top						
	burgundy						

ASIN : B01C60RLDQ

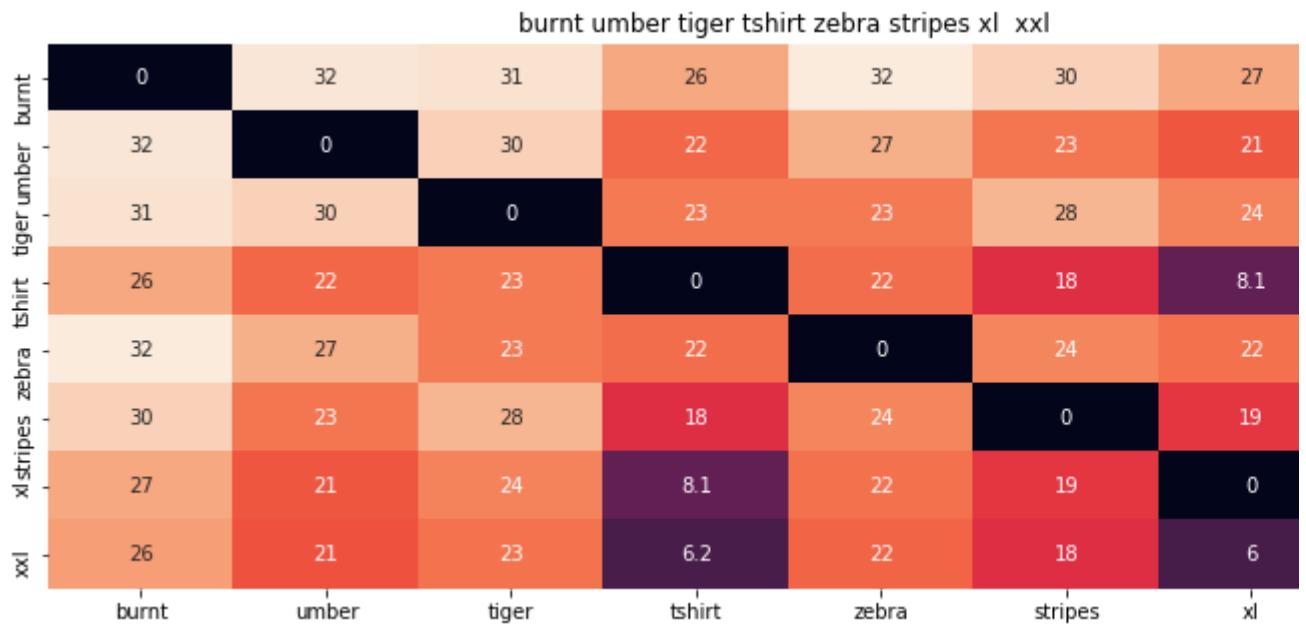
Brand : 1 Mad Fit

euclidean distance from input : 4.645918274287157

```
# brand and color weight =50
# title vector weight = 5
```

```
idf_w2v_brand(12566, 5, 50, 20)
```

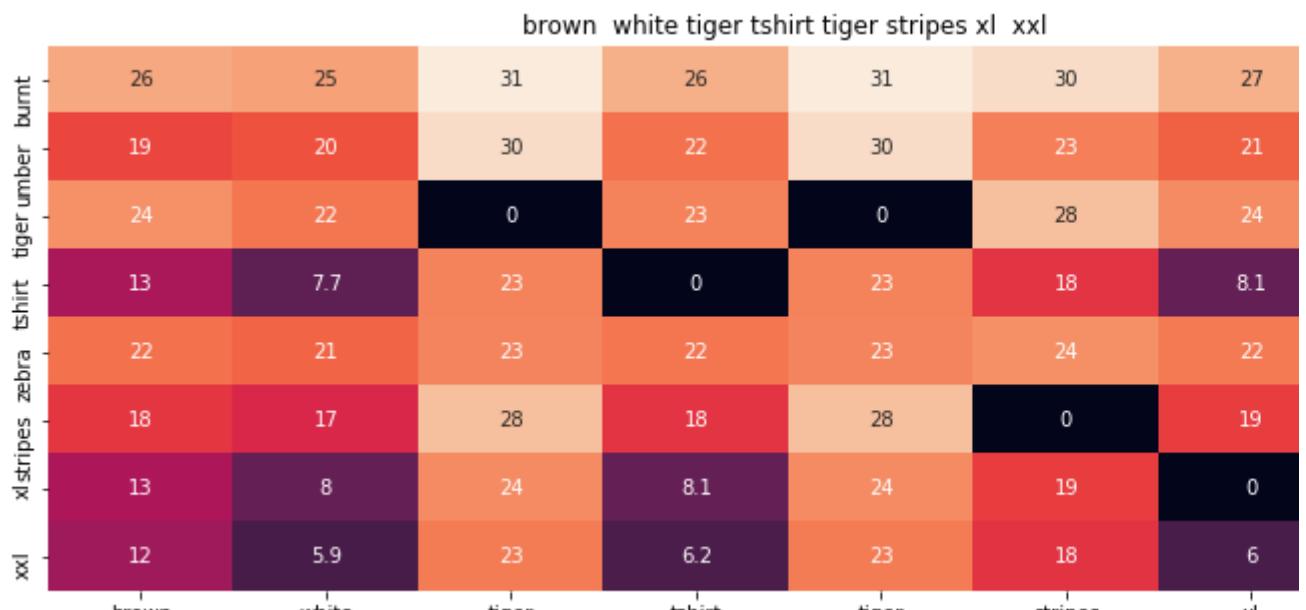




ASIN : B00JXQB5FQ

Brand : Si Row

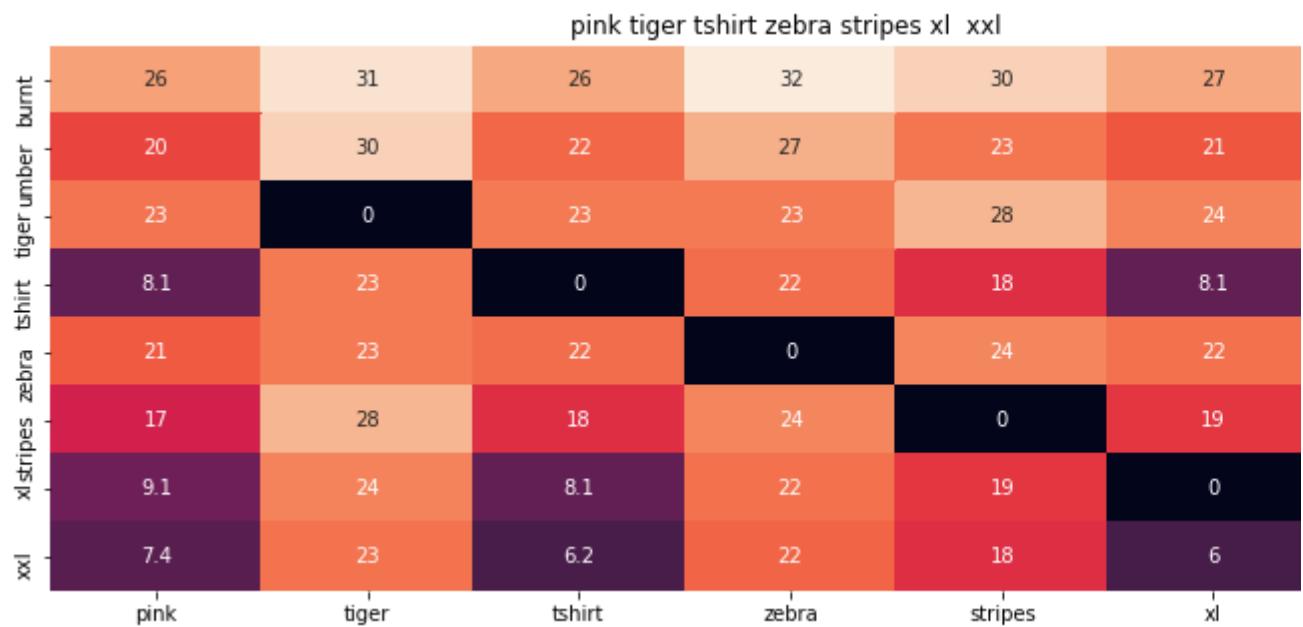
euclidean distance from input : 0.0003551136363636364



ASIN : B00JXQCWT0

Brand : Si Row

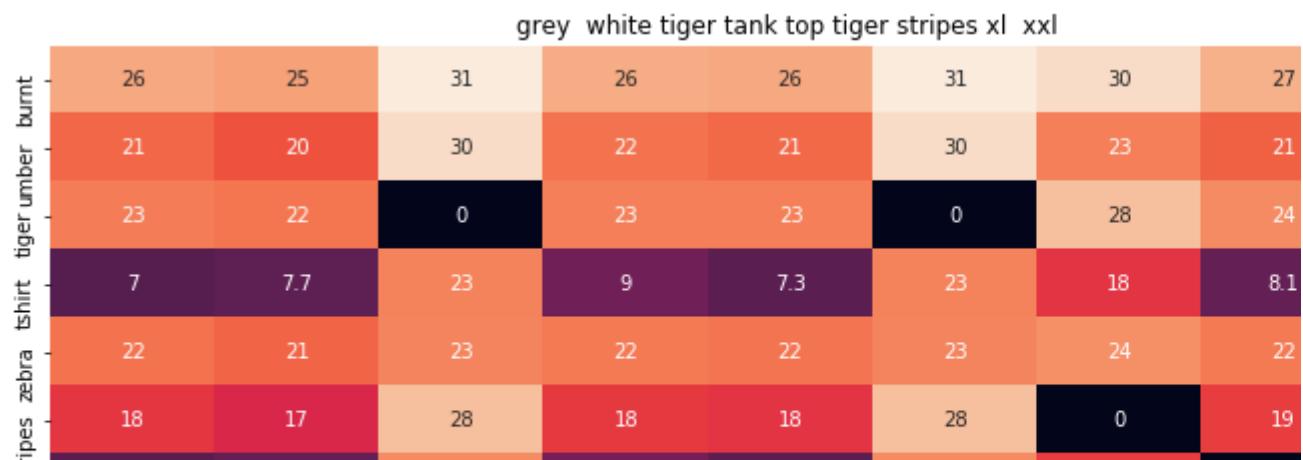
euclidean distance from input : 0.43372202786532316



ASIN : B00JXQASS6

Brand : Si Row

euclidean distance from input : 1.655093037326926



xxl	7.2	8	24	9.4	7.5	24	19	0
xxl	4.7	5.9	23	7.8	5	23	18	6
	grey	white	tiger	tank	top	tiger	stripes	xl

ASIN : B00JXQAFZ2

Brand : Si Row

euclidean distance from input : 1.7729360757124906

=====

	yellow	tiger	tank	top	stripes	l
burnt	27	31	26	26	31	
umber	20	30	22	21	30	
tiger	24	0	23	23	0	
tshirt	13	23	9	7.3	23	
zebra	21	23	22	22	23	
stripes	17	28	18	18	28	
xl	13	24	9.4	7.5	24	
xxl	12	23	7.8	5	23	
	yellow	tiger	tank	top	tiger	

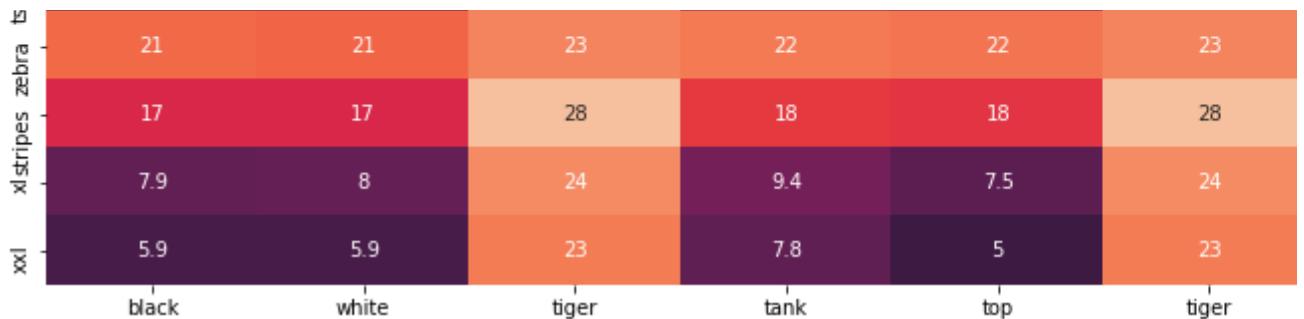
ASIN : B00JXQUWA

Brand : Si Row

euclidean distance from input : 1.8028780853782398

=====

	black	white	tiger	tank	top	tiger	stripes	l
burnt	25	25	31	26	26	31		
umber	20	20	30	22	21	30		
tiger	22	22	0	23	23	0		
tshirt	7.6	7.7	23	9	7.3	23		

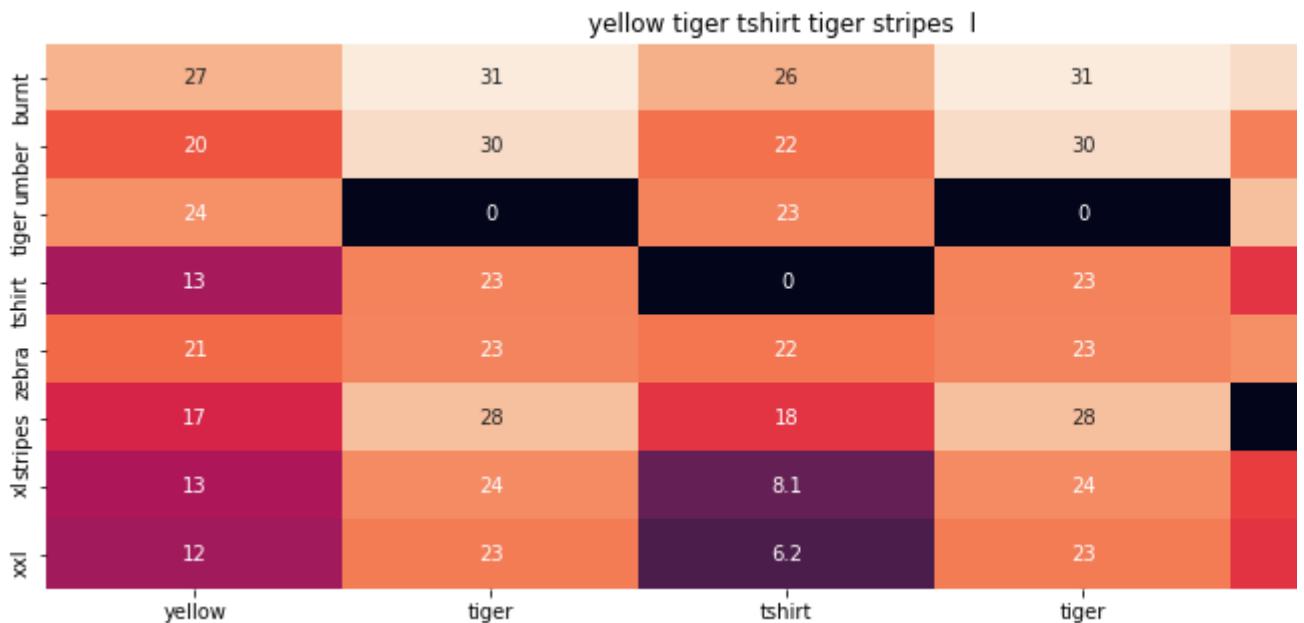


ASIN : B00JXQA094

Brand : Si Row

euclidean distance from input : 1.803196231130193

=====

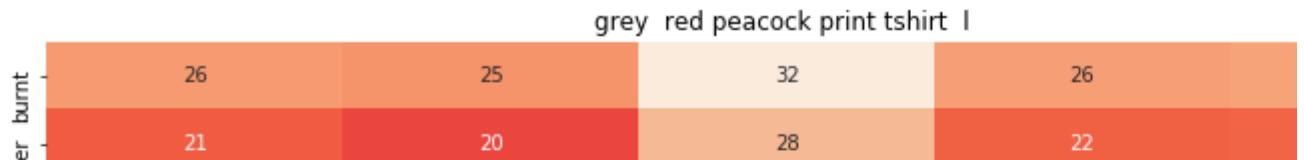


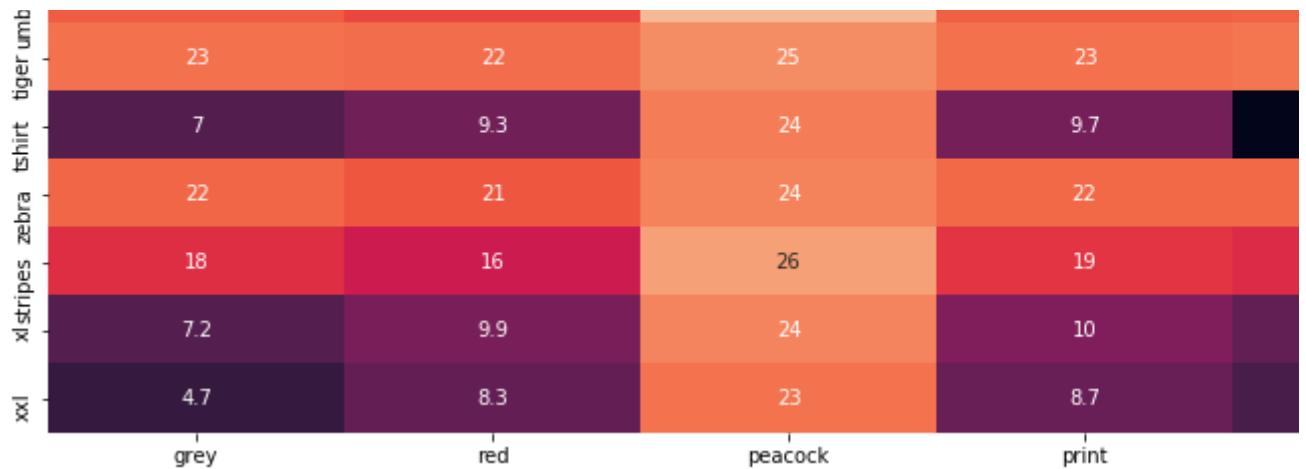
ASIN : B00JXQCUIC

Brand : Si Row

euclidean distance from input : 1.8214161962846673

=====

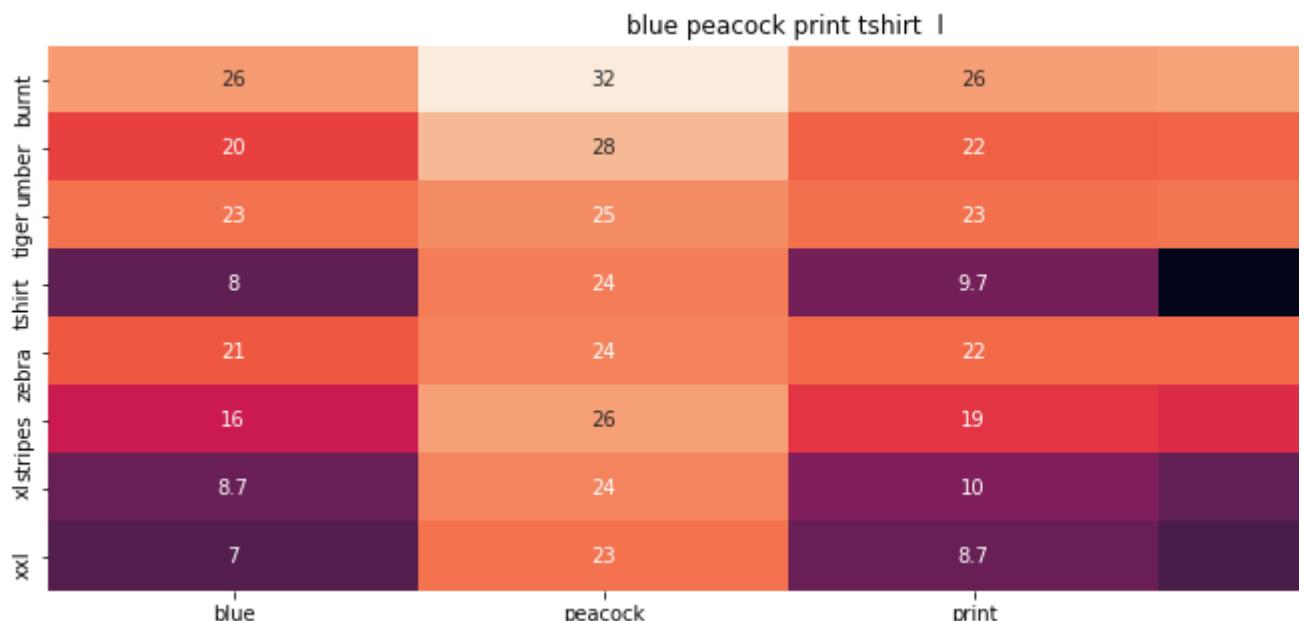




ASIN : B00JXQCFRS

Brand : Si Row

euclidean distance from input : 1.9077768502486234



ASIN : B00JXQC8L6

Brand : Si Row

euclidean distance from input : 1.9214293049716347

red butterfly black white tank top xl xxl

	red	butterfly	black	white	tank	top	xl
burnt	25	33	25	25	26	26	27
tiger umber	20	27	20	20	22	21	21
zebra	22	27	22	22	23	23	24
tshirt	9.3	21	7.6	7.7	9	7.3	8.1
xxl	21	26	21	21	22	22	22
stripes	16	26	17	17	18	18	19
xl	9.9	22	7.9	8	9.4	7.5	0
xl	8.3	21	5.9	5.9	7.8	5	6

ASIN : B00JV63CW2

Brand : Si Row

euclidean distance from input : 1.9364632349698592

=====

yellow pink butterfly color burst tshirt xl xxl

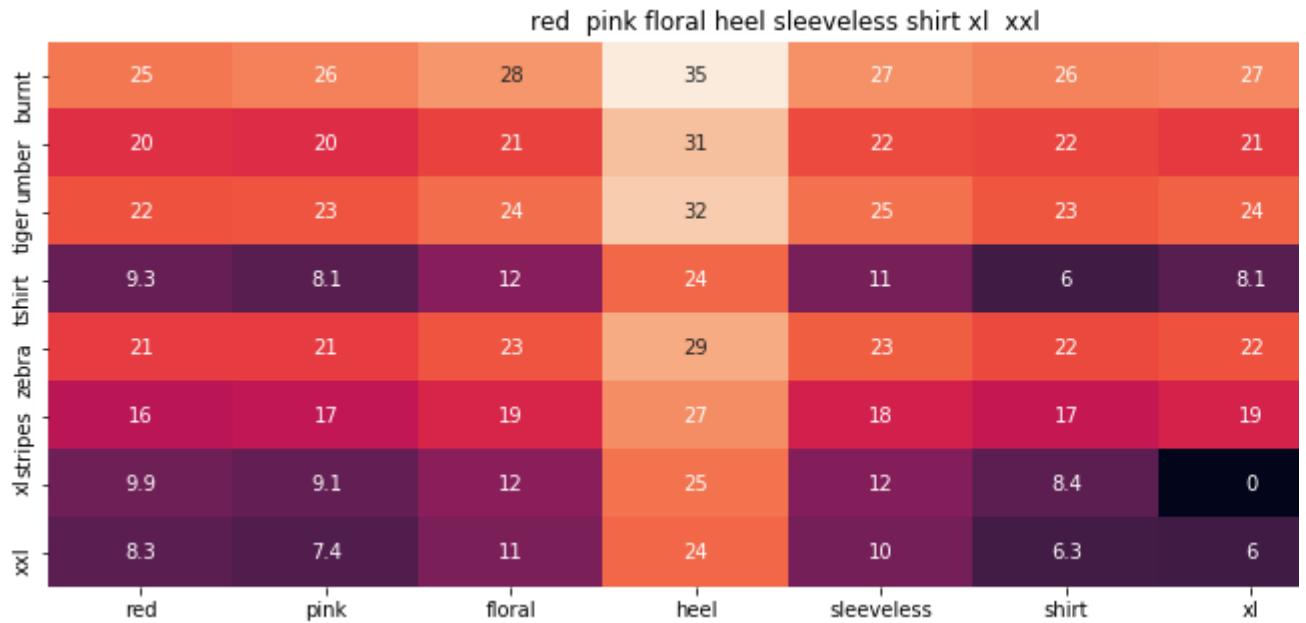
	yellow	pink	butterfly	color	burst	tshirt	xl
burnt	27	26	33	27	32	26	27
tiger umber	20	20	27	20	32	22	21
zebra	24	23	27	24	33	23	24
tshirt	13	8.1	21	13	26	0	8.1
xxl	21	21	26	22	32	22	22
stripes	17	17	26	17	29	18	19
xl	13	9.1	22	13	26	8.1	0
xl	12	7.4	21	12	25	6.2	6

ASIN : B00JXQBBMI

Brand : Si Row

euclidean distance from input : 1.956703810586869

=====



ASIN : B00JV63QQE

Brand : Si Row

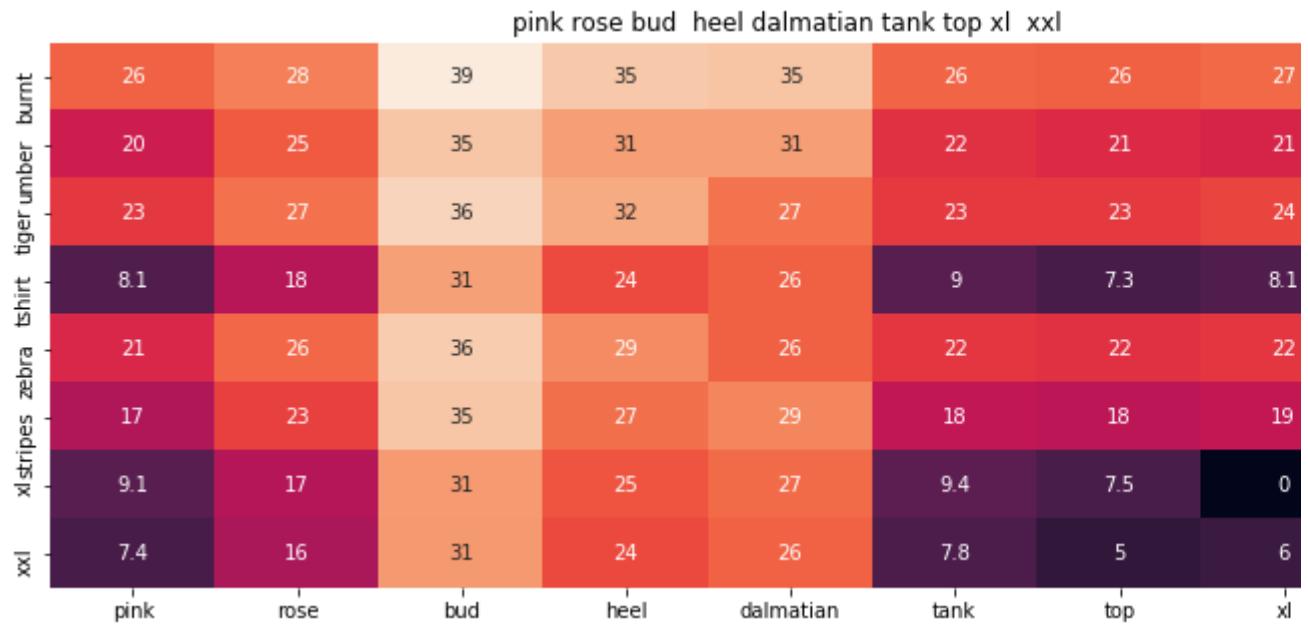
euclidean distance from input : 1.9806066342951432



ASIN : B00JV63VC8

Brand : Si Row

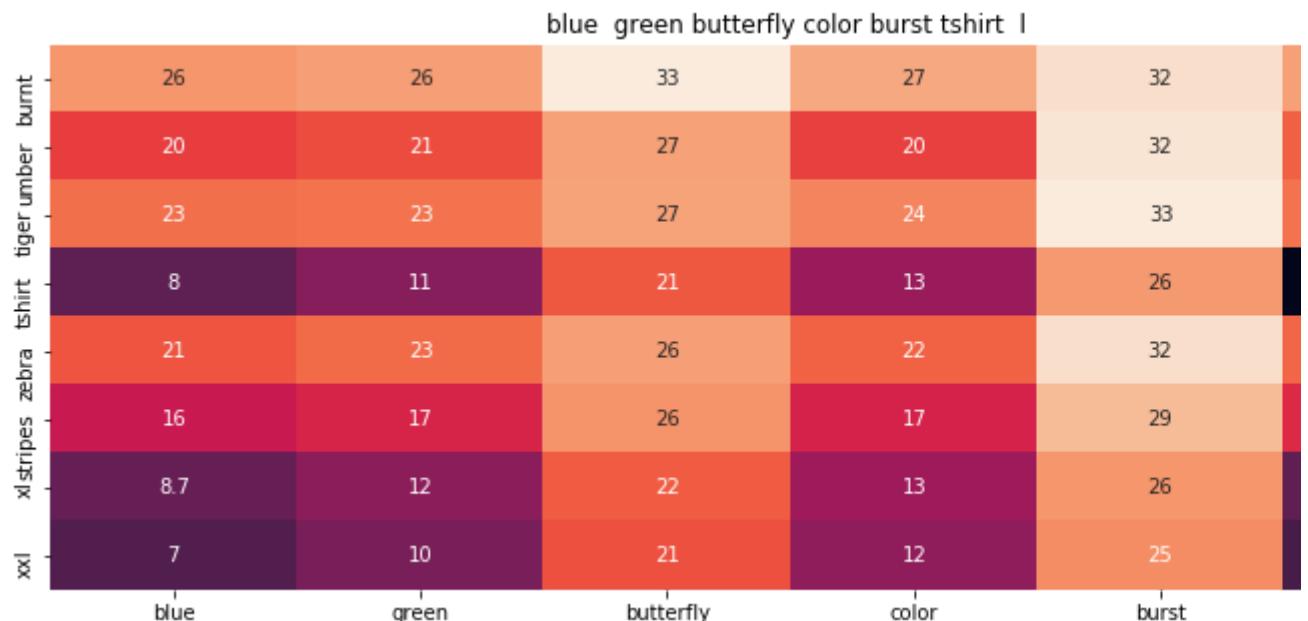
euclidean distance from input : 2.0121855999157043



ASIN : B00JXQAX2C

Brand : Si Row

euclidean distance from input : 2.013351787548872



ASIN : B00JXQC0C8

Brand : Si Row

euclidean distance from input : 2.013883348273304

=====

	red	rose	bud	heel	dalmatian	tshirt	xl	xxl
burnt	25	28	39	35	35	26	27	
tiger	20	25	35	31	31	22	21	
umbre	22	27	36	32	27	23	24	
tshirt	9.3	18	31	24	26	0	8.1	
zebra	21	26	36	29	26	22	22	
stripes	16	23	35	27	29	18	19	
xl	9.9	17	31	25	27	8.1	0	
xxl	8.3	16	31	24	26	6.2	6	
	red	rose	bud	heel	dalmatian	tshirt	xl	xxl

ASIN : B00JXQABB0

Brand : Si Row

euclidean distance from input : 2.0367257554998663

=====

	gupobou	168	womens	girls	lady	boho	elephant	stripes	bandage	tank	top	one size
burnt	26	26	28	28	31	32	30	35	26	26		
tiger	21	21	24	24	25	29	23	32	22	21		
umber	23	23	25	25	29	18	28	33	23	23		
tshirt	7	6.8	14	15	19	23	18	25	9	7.3		
zebra	22	22	25	24	27	22	24	32	22	22		
stripes	18	18	22	23	24	28	0	28	18	18		
xxl	7.2	7	11	15	20	22	20	27	8.4	7.5		
	gupobou	168	womens	girls	lady	boho	elephant	stripes	bandage	tank	top	one size

xl	7.2	7	14	15	20	23	19	27	9.4	7.5
xxl	4.7	4.5	13	14	19	22	18	26	7.8	5

ASIN : B01ER18406

Brand : GuPoBoU168

euclidean distance from input : 2.6562041677776853

=====

wayf womens button cold shoulder leopard blouse brown xs

	wayf	womens	button	cold	shoulder	leopard	blouse	brow
burnt	26	26	29	28	28	30	26	26
umber	21	21	24	25	25	27	21	19
tiger	23	23	25	26	25	19	23	24
tshirt	7	6.8	13	15	13	21	7.3	13
zebra	22	22	25	25	24	20	22	22
stripes	18	18	20	22	19	26	18	18
xs	7.2	7	14	15	14	22	8.4	13
xxl	4.7	4.5	12	14	12	20	6.2	12

ASIN : B01LZ7BQ4H

Brand : WAYF

euclidean distance from input : 2.684906782301833

=====

yabina womens pullover leopard print long sleeve hoodie us10 brown

burnt	26	26	31	30	26	26	26	32
umber	21	21	25	27	22	22	22	28
tiger	23	23	26	19	23	23	23	28
tshirt	7	6.8	14	21	9.7	8.6	7.3	17

	22	22	25	20	22	22	22	27	22
xxl stripes zebra	18	18	21	26	19	19	17	23	22
xl stripes	7.2	7	16	22	10	8.5	8.3	19	7.2
xxl	4.7	4.5	15	20	8.7	6.6	6.6	18	4.2
	yabina	womens	pullover	leopard	print	long	sleeve	hoodie	xxl

ASIN : B01KJUM6JI

Brand : YABINA

euclidean distance from input : 2.6858381926578345

=====

	wayf	womens	small	button	leopard	cold	shoulder	blouse	brown
	26	26	26	29	30	28	28	26	26
burnt	21	21	21	24	27	25	25	21	21
tiger	23	23	23	25	19	26	25	23	23
umber	7	6.8	8.5	13	21	15	13	7.3	7.3
shirt	22	22	22	25	20	25	24	22	22
zebra	18	18	19	20	26	22	19	18	18
xxl stripes	7.2	7	8.4	14	22	15	14	8.4	8.4
xxl	4.7	4.5	6.4	12	20	14	12	6.2	6.2
	wayf	womens	small	button	leopard	cold	shoulder	blouse	brown

ASIN : B01M06V4X1

Brand : WAYF

euclidean distance from input : 2.694761948650377

## ▼ [10.2] Keras and Tensorflow to extract features

```
# https://gist.github.com/fchollet/f35fbc80e066a49d65f1688a7e99f069
# Code reference: https://blog.keras.io/building-powerful-image-classification-models-using-very
```

```
# This code takes 40 minutes to run on a modern GPU (graphics card)
# like Nvidia 1050.
```

```
# GPU (Nvidia 1050): 0.175 seconds per image
```

```
# This codse takes 160 minutes to run on a high end i7 CPU
# CPU (i7): 0.615 seconds per image.
```

```
#DO NOT run this code unless you want to wait a few hours for it to generate output

# each image is converted into 25088 length dense-vector

...
# dimensions of our images.
img_width, img_height = 224, 224

top_model_weights_path = 'bottleneck_fc_model.h5'
train_data_dir = 'images2/'
nb_train_samples = 16042
epochs = 50
batch_size = 1

def save_bottlebeck_features():

    #Function to compute VGG-16 CNN for image feature extraction.

    asins = []
    datagen = ImageDataGenerator(rescale=1. / 255)

    # build the VGG16 network
    model = applications.VGG16(include_top=False, weights='imagenet')
    generator = datagen.flow_from_directory(
        train_data_dir,
        target_size=(img_width, img_height),
        batch_size=batch_size,
        class_mode=None,
        shuffle=False)

    for i in generator.filenames:
        asins.append(i[2:-5])

    bottleneck_features_train = model.predict_generator(generator, nb_train_samples // batch_size)
    bottleneck_features_train = bottleneck_features_train.reshape((16042,25088))

    np.save(open('16k_data_cnn_features.npy', 'wb'), bottleneck_features_train)
    np.save(open('16k_data_cnn_feature_asins.npy', 'wb'), np.array(asins))

save_bottlebeck_features()
```

### ▼ [10.3] Visual features based product similarity

```
#load the features and corresponding ASINS info.
bottleneck_features_train = np.load('16k_data_cnn_features.npy')
asins = np.load('16k_data_cnn_feature_asins.npy')
asins = list(asins)

# load the original 16K dataset
data = pd.read_pickle('pickels/16k_apperal_data_preprocessed')
df_asins = list(data['asin'])

from IPython.display import display, Image, SVG, Math, YouTubeVideo

#get similar products using CNN features (VGG-16)
def get_similar_products_cnn(doc_id, num_results):
    doc_id = asins.index(df_asins[doc_id])
    pairwise_dist = pairwise_distances(bottleneck_features_train, bottleneck_features_train[doc_
```

```
indices = np.argsort(pairwise_dist.flatten())[0:num_results]
pdists = np.sort(pairwise_dist.flatten())[0:num_results]

for i in range(len(indices)):
    rows = data[['medium_image_url','title']].loc[data['asin']==asins[indices[i]]]
    for idx, row in rows.iterrows():
        display(Image(url=row['medium_image_url'], embed=True))
        print('Product Title: ', row['title'])
        print('Euclidean Distance from input image:', pdists[i])
        print('Amazon Url: www.amazon.com/dp/' + asins[indices[i]])
```

get\_similar\_products\_cnn(12566, 20)





Product Title: burnt umber tiger tshirt zebra stripes xl xxl  
Euclidean Distance from input image: 0.044194173  
Amazon Url: [www.amazon.com/dp/B00JXQB5FQ](https://www.amazon.com/dp/B00JXQB5FQ)



Product Title: pink tiger tshirt zebra stripes xl xxl  
Euclidean Distance from input image: 30.050056  
Amazon Url: [www.amazon.com/dp/B00JXQASS6](https://www.amazon.com/dp/B00JXQASS6)



Product Title: yellow tiger tshirt tiger stripes l  
Euclidean Distance from input image: 41.261112  
Amazon Url: [www.amazon.com/dp/B00JXQCUIC](https://www.amazon.com/dp/B00JXQCUIC)



Product Title: brown white tiger tshirt tiger stripes xl xxl  
Euclidean Distance from input image: 44.0002  
Amazon Url: [www.amazon.com/dp/B00JXQCWTO](https://www.amazon.com/dp/B00JXQCWTO)



Product Title: kawaii pastel tops tees pink flower design  
Euclidean Distance from input image: 47.38251  
Amazon Url: [www.amazon.com/dp/B071FCWD97](https://www.amazon.com/dp/B071FCWD97)



Product Title: womens thin style tops tees pastel watermelon print  
Euclidean Distance from input image: 47.71839  
Amazon Url: [www.amazon.com/dp/B01JUNHBRM](https://www.amazon.com/dp/B01JUNHBRM)



Product Title: kawaii pastel tops tees baby blue flower design  
Euclidean Distance from input image: 47.9021  
Amazon Url: [www.amazon.com/dp/B071SBCY9W](https://www.amazon.com/dp/B071SBCY9W)



Product Title: edv cheetah run purple multi xl  
Euclidean Distance from input image: 48.046467  
Amazon Url: [www.amazon.com/dp/B01CUPYBM0](https://www.amazon.com/dp/B01CUPYBM0)



Product Title: danskin womens vneck loose performance tee xsmall pink ombre  
Euclidean Distance from input image: 48.101875  
Amazon Url: [www.amazon.com/dp/B01F7PHXY8](https://www.amazon.com/dp/B01F7PHXY8)





Product Title: summer alpaca 3d pastel casual loose tops tee design  
Euclidean Distance from input image: 48.118896  
Amazon Url: [www.amazon.com/dp/B01I80A93G](https://www.amazon.com/dp/B01I80A93G)



Product Title: miss chievous juniors striped peplum tank top medium shadowpeach  
Euclidean Distance from input image: 48.13128  
Amazon Url: [www.amazon.com/dp/B0177DM70S](https://www.amazon.com/dp/B0177DM70S)



Product Title: red pink floral heel sleeveless shirt xl xxl  
Euclidean Distance from input image: 48.16945  
Amazon Url: [www.amazon.com/dp/B00JV63QQE](https://www.amazon.com/dp/B00JV63QQE)



Product Title: moana logo adults hot v neck shirt black xxl  
Euclidean Distance from input image: 48.25678  
Amazon Url: [www.amazon.com/dp/B01LX6H43D](https://www.amazon.com/dp/B01LX6H43D)



Product Title: abaday multicolor cartoon cat print short sleeve longline shirt large  
Euclidean Distance from input image: 48.265644

Amazon Url: [www.amazon.com/dp/B01CR57YY0](http://www.amazon.com/dp/B01CR57YY0)



Product Title: kawaii cotton pastel tops tees peach pink cactus design

Euclidean Distance from input image: 48.362583

Amazon Url: [www.amazon.com/dp/B071WYLBZS](http://www.amazon.com/dp/B071WYLBZS)



Product Title: chicago chicago 18 shirt women pink

Euclidean Distance from input image: 48.383648

Amazon Url: [www.amazon.com/dp/B01GXAZTRY](http://www.amazon.com/dp/B01GXAZTRY)



Product Title: yichun womens tiger printed summer tshirts tops

Euclidean Distance from input image: 48.449345

Amazon Url: [www.amazon.com/dp/B010NN9RX0](http://www.amazon.com/dp/B010NN9RX0)



Product Title: nancy lopez whimsy short sleeve whiteblacklemon drop xs

Euclidean Distance from input image: 48.478893

Amazon Url: [www.amazon.com/dp/B01MPX6IDX](http://www.amazon.com/dp/B01MPX6IDX)





Product Title: womens tops tees pastel peach ice cream cone print  
 Euclidean Distance from input image: 48.557983  
 Amazon Url: [www.amazon.com/dp/B0734GRKZL](https://www.amazon.com/dp/B0734GRKZL)



Product Title: uswomens mary j blige without tshirts shirt

## ▼ Assignment

### ▼ Weighted similarity using Brand Color and Image.

```
extra_features = hstack((brand_features, type_features, color_features,bottleneck_features_train)

def heat_map_w2v_brand(sentance1, sentance2, url, doc_id1, doc_id2, df_id1, df_id2, model):

    # sentance1 : title1, input apparel
    # sentance2 : title2, recommended apparel
    # url: apparel image url
    # doc_id1: document id of input apparel
    # doc_id2: document id of recommended apparel
    # df_id1: index of document1 in the data frame
    # df_id2: index of document2 in the data frame
    # model: it can have two values, 1. avg 2. weighted

    s1_vec = np.array(#number_of_words_title1 * 300), each row is a vector(weighted/avg) of len
    s1_vec = get_word_vec(sentance1, doc_id1, model)
    s2_vec = np.array(#number_of_words_title2 * 300), each row is a vector(weighted/avg) of len
    s2_vec = get_word_vec(sentance2, doc_id2, model)

    # s1_s2_dist = np.array(#number of words in title1 * #number of words in title2)
    # s1_s2_dist[i,j] = euclidean distance between words i, j
    s1_s2_dist = get_distance(s1_vec, s2_vec)

    data_matrix = [['Asin', 'Brand', 'Color', 'Product type'],
                  [data['asin'].loc[df_id1], brands[doc_id1], colors[doc_id1], types[doc_id1]], # in
                  [data['asin'].loc[df_id2], brands[doc_id2], colors[doc_id2], types[doc_id2]]] # re

    colorscale = [[0, '#1d004d'], [.5, '#f2e5ff'], [1, '#f2e5d1']] # to color the headings of each

    # we create a table with the data_matrix
    table = ff.create_table(data_matrix, index=True, colorscale=colorscale)
    # plot it with plotly
    plotly.offline.iplot(table, filename='simple_table')

    # devide whole figure space into 25 * 1:10 grids
    gs = gridspec.GridSpec(25, 15)
    fig = plt.figure(figsize=(25,5))
```

```

# in first 25*10 grids we plot heatmap
ax1 = plt.subplot(gs[:, :-5])
# plotting the heap map based on the pairwise distances
ax1 = sns.heatmap(np.round(s1_s2_dist,6), annot=True)
# set the x axis labels as recommended apparels title
ax1.set_xticklabels(sentance2.split())
# set the y axis labels as input apparels title
ax1.set_yticklabels(sentance1.split())
# set title as recommended apparels title
ax1.set_title(sentance2)

# in last 25 * 10:15 grids we display image
ax2 = plt.subplot(gs[:, 10:16])
# we dont display grid lines and axis labels to images
ax2.grid(False)
ax2.set_xticks([])
ax2.set_yticks([])

# pass the url it display it
display_img(url, ax2, fig)

plt.show()

from PIL import Image
import PIL.Image

def idf_w2v_brand(doc_id, w1, w2, num_results):
    # doc_id: apparel's id in given corpus
    # w1: weight for w2v features
    # w2: weight for brand and color features

    # pairwise_dist will store the distance from given input apparel to all remaining apparels
    # the metric we used here is cosine, the coside distance is mesured as K(X, Y) = <X, Y> / (|X|*|Y|)
    # http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
    idf_w2v_dist = pairwise_distances(w2v_title_weight, w2v_title_weight[doc_id].reshape(1,-1))
    ex_feat_dist = pairwise_distances(extra_features, extra_features[doc_id])
    pairwise_dist = (w1 * idf_w2v_dist + w2 * ex_feat_dist)/float(w1 + w2)

    # np.argsort will return indices of 9 smallest distances
    indices = np.argsort(pairwise_dist.flatten())[0:num_results]
    #pdists will store the 9 smallest distances
    pdists = np.sort(pairwise_dist.flatten())[0:num_results]

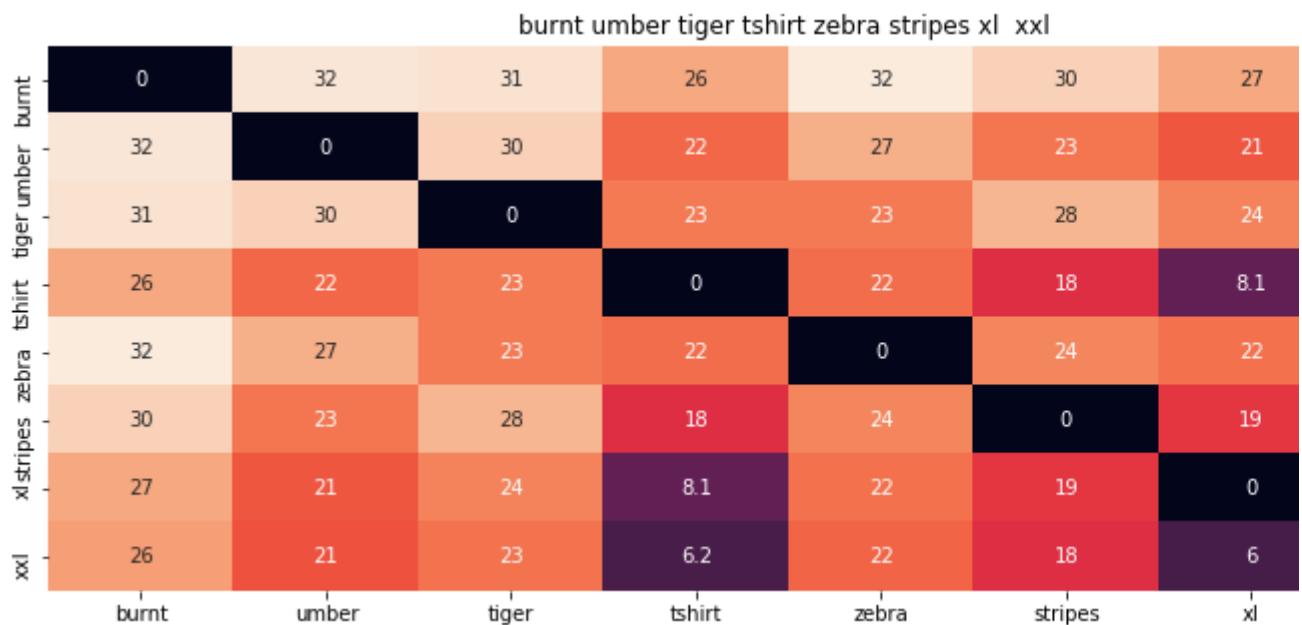
    #data frame indices of the 9 smallest distace's
    df_indices = list(data.index[indices])

    for i in range(0, len(indices)):
        heat_map_w2v_brand(data['title'].loc[df_indices[0]], data['title'].loc[df_indices[i]], data['asin'].loc[df_indices[i]])
        print('ASIN :', data['asin'].loc[df_indices[i]])
        print('Brand :', data['brand'].loc[df_indices[i]])
        print('euclidean distance from input :', pdists[i])
        print('*'*125)

idf_w2v_brand(12566, 5, 5, 20)
# in the give heat map, each cell contains the euclidean distance between words i, j

```

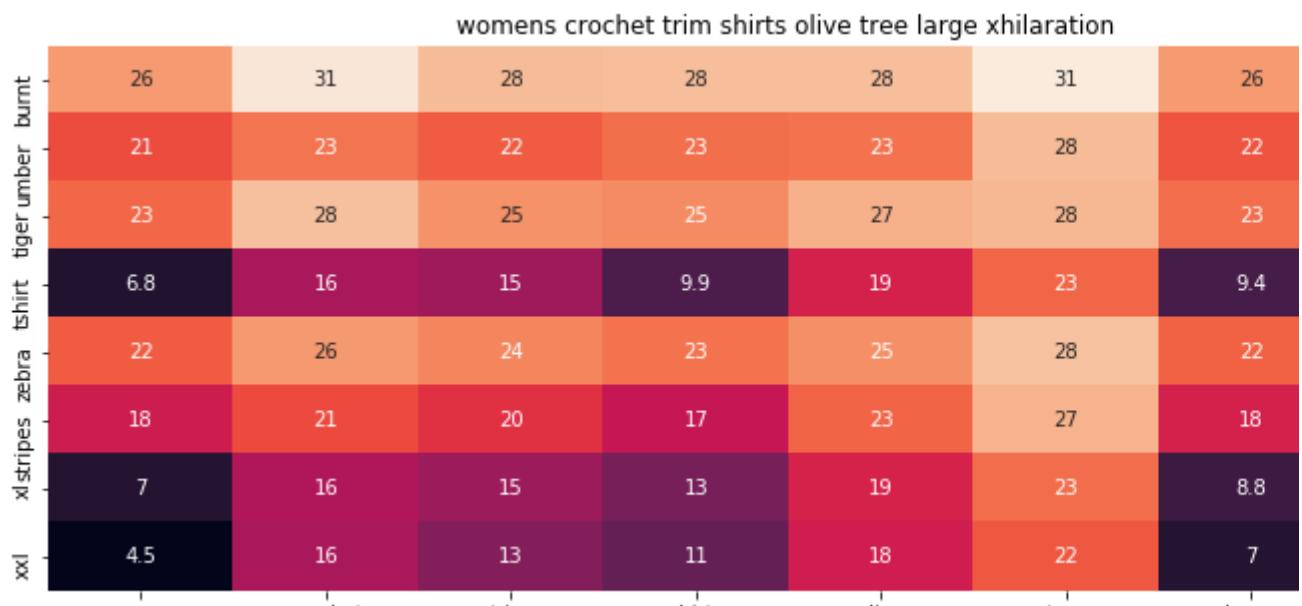




ASIN : B00JXQB5FQ

Brand : Si Row

euclidean distance from input : 0.001953125



WOMENS

DUCHESS

UNI

SHIRTS

LINE

TEE

LARGE

ASIN : B06XBHNM7J

Brand : Xhilaration

euclidean distance from input : 22.6133604375347

buffalo david bitton nipaw logo graphic tank white combo xxl

	buffalo	david	bitton	nipaw	logo	graphic	tank	white	co
burnt	31	34	26	26	28	27	26	25	2
tiger	28	28	21	21	24	22	22	20	2
umber	23	32	23	23	25	24	23	22	2
tshirt	21	22	7	7	13	11	9	7.7	2
zebra	21	31	22	22	24	23	22	21	2
stripes	26	28	18	18	19	19	18	17	2
xl	22	22	7.2	7.2	15	13	9.4	8	2
xxl	20	22	4.7	4.7	14	11	7.8	5.9	2

ASIN : B018H5AZXQ

Brand : Buffalo

euclidean distance from input : 23.542129858315462

j brand womens pinstripe shirt xs blue

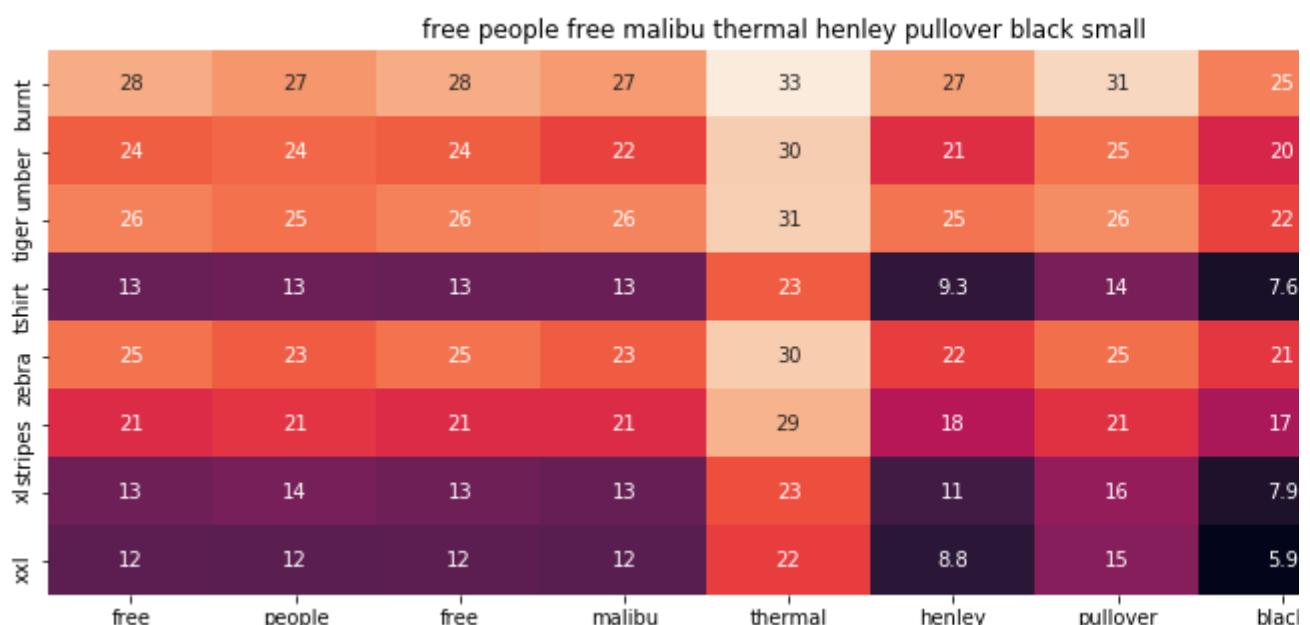
	29	26	36	26	26
burnt	29	26	36	26	26
umber	27	21	29	22	21
tiger	27	23	35	23	23
tshirt	16	6.8	26	6	7.7
zebra	26	22	33	22	23
stripes	23	18	26	17	19



ASIN : B06XYP1X1F

Brand : J Brand Jeans

euclidean distance from input : 23.915188829383087



ASIN : B074MXY984

Brand : We The Free

euclidean distance from input : 23.954450721745104



ts	32	26	26	22	32	24	22
zebra	30	23	24	19	32	19	18
xl stripes	25	16	17	8.5	26	13	7.2
xxl	25	15	16	6.6	26	12	4.7
	completely	liz	lange	long	flyaway	vest	249682

ASIN : B074LTBWSW

Brand : Liz Lange

euclidean distance from input : 24.072081048712228

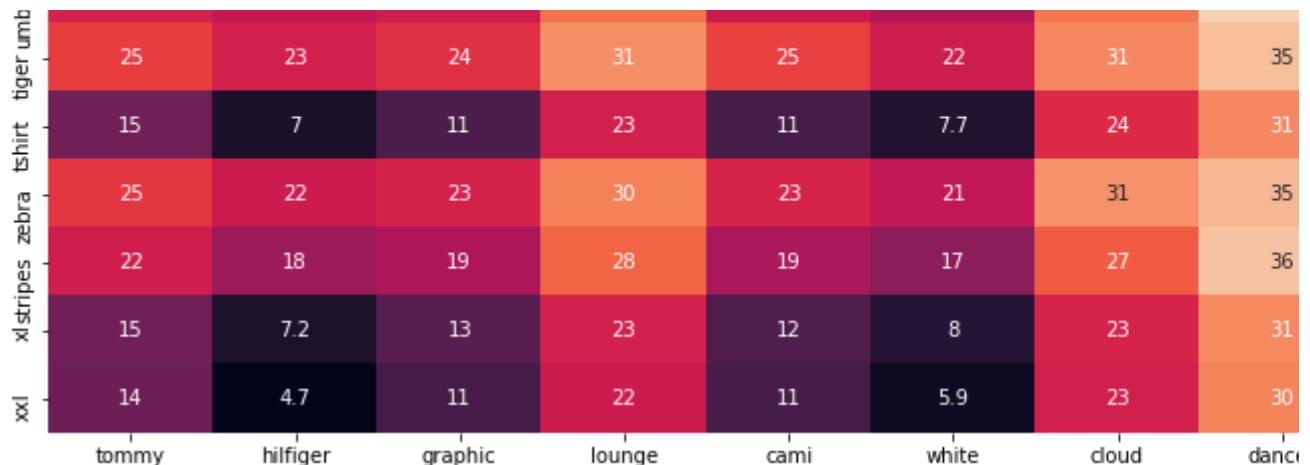
	bila	size	small	womens	sleeveless	blouse	red
burnt	26	26	26	26	27	26	
umber	22	21	21	21	22	21	
tiger	24	23	23	23	25	23	
tshirt	13	8.3	8.5	6.8	11	7.3	
zebra	24	22	22	22	23	22	
stripes	21	18	19	18	18	18	
xl	12	7.7	8.4	7	12	8.4	
xxl	11	5.9	6.4	4.5	10	6.2	
	bila	size	small	womens	sleeveless	blouse	

ASIN : B01L7ROZNC

Brand : Bila

euclidean distance from input : 24.361907411018894

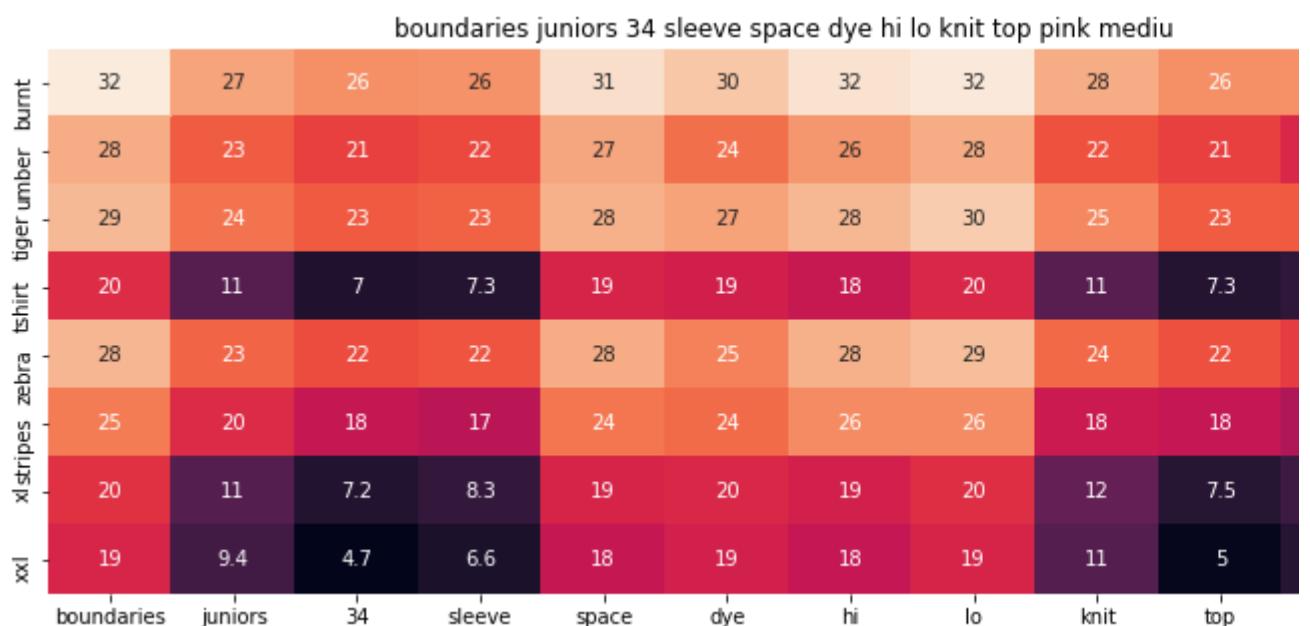
	tommy	hilfiger	graphic	lounge	cami	white	cloud	dancer	xlarge
burnt	29	26	27	33	28	25	34	39	
ier	23	21	22	29	22	20	29	37	



ASIN : B01BMSFYW2

Brand : igertommy hilf

euclidean distance from input : 24.403390482965563



ASIN : B01EXXFS4M

Brand : No Boundaries

euclidean distance from input : 24.98053757374054

## kongyii womens charlotte hornets a sport pique polo

	kongyii	womens	charlotte	hornets	a	sport
xxl	4.7	4.5	23	31	14	18
xl stripes	7.2	7	24	31	15	19
zebra	18	18	30	34	21	24
tshirt	22	22	30	34	25	28
tiger umber	23	23	33	34	26	29
burnt	21	21	29	35	25	26
burnt	26	26	36	37	29	31

ASIN : B01FJVZST2

Brand : KONGYII

euclidean distance from input : 25.04757956004972

=====

## byoung womens henya womens light blue shirt size 40l light blue

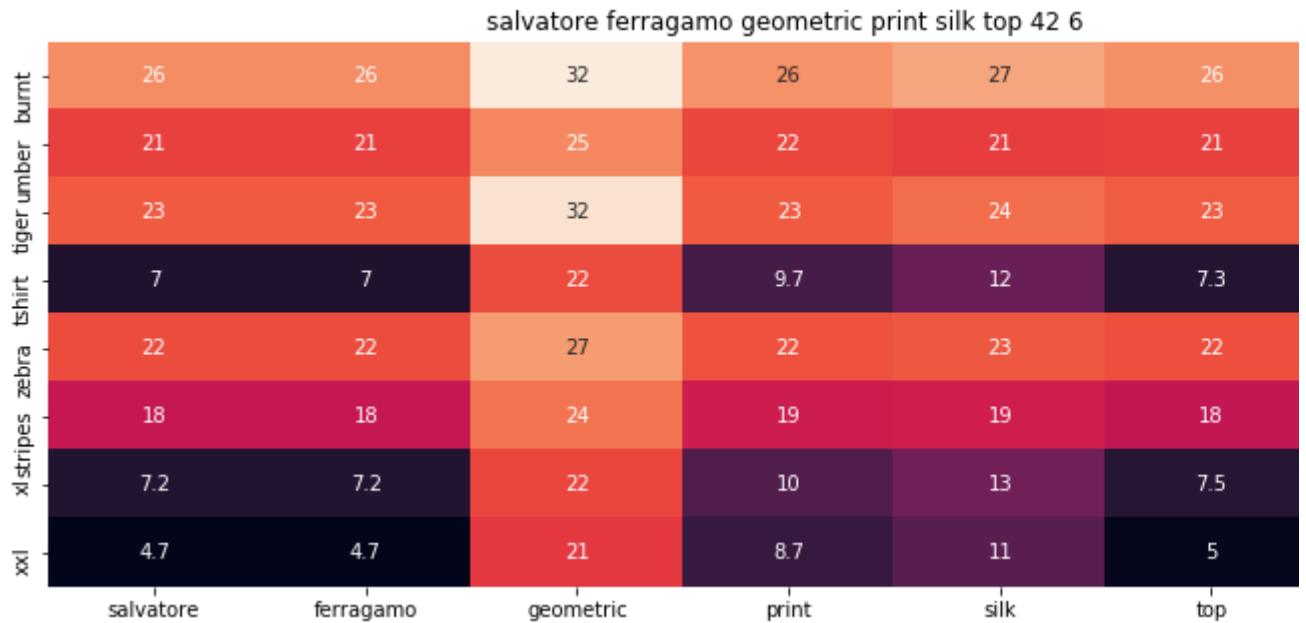
	byoung	womens	henya	womens	light	blue	shirt	size	40l	light blue
xxl	4.7	4.5	4.7	4.5	12	7	6.3	5.9	4.7	
xl stripes	7.2	7	7.2	7	13	8.7	8.4	7.7	7.2	
zebra	22	22	22	22	24	21	22	22	22	
tshirt	7	6.8	7	6.8	13	8	6	8.3	7	
tiger umber	23	23	23	23	25	23	23	23	23	
burnt	21	21	21	21	22	20	22	21	21	
burnt	26	26	26	26	27	26	26	26	26	

ASIN : B06Y41MRCH

Brand : Byoung

euclidean distance from input : 25.07057458486874

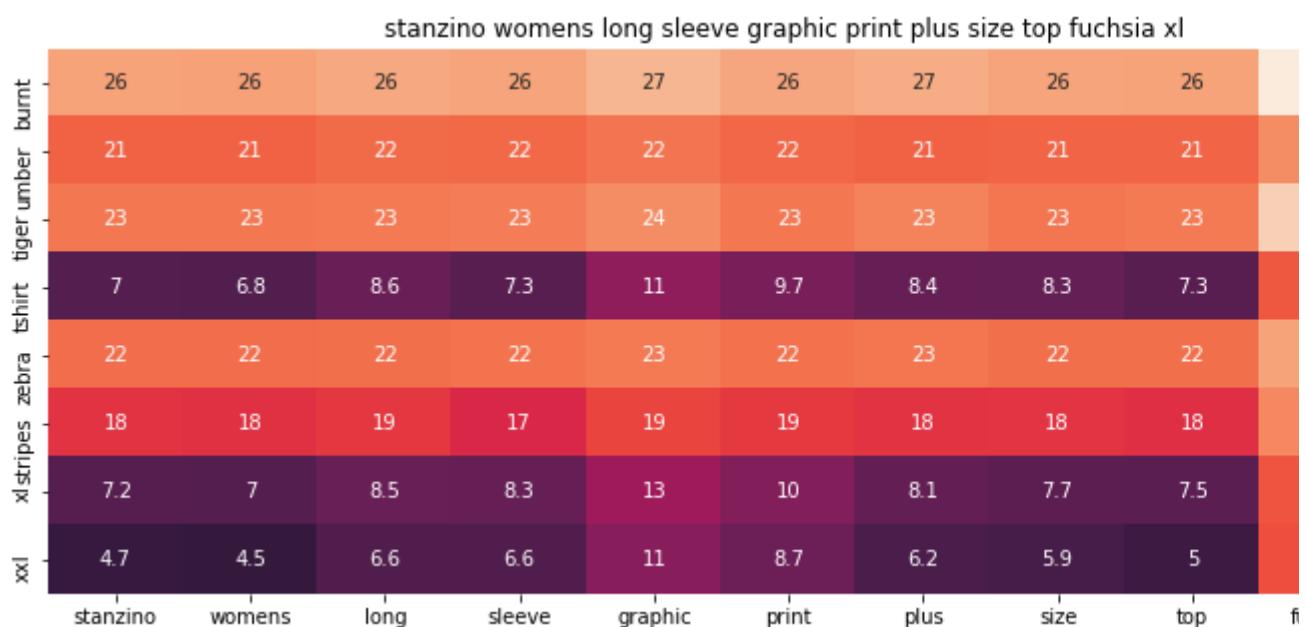
=====



ASIN : B0756JTS1F

Brand : Salvatore Ferragamo

euclidean distance from input : 25.0759369449104



ASIN : B00DP4VH1I

Brand : Stanzino

euclidean distance from input : 25.096636124141895

	1state	womens	medium	chambray	crochet	solid	blouse
1state	26	26	26	30	31	27	26
tiger	21	21	22	23	23	23	21
umber	23	23	24	28	28	25	23
burnt	7	6.8	9.7	16	16	12	7.3
zebra	22	22	23	25	26	24	22
stripes	18	18	19	21	21	20	18
xl	7.2	7	9.4	17	16	12	8.4
xxl	4.7	4.5	8	16	16	11	6.2

ASIN : B074MK6LV2

Brand : 1.State

euclidean distance from input : 25.11281129449343

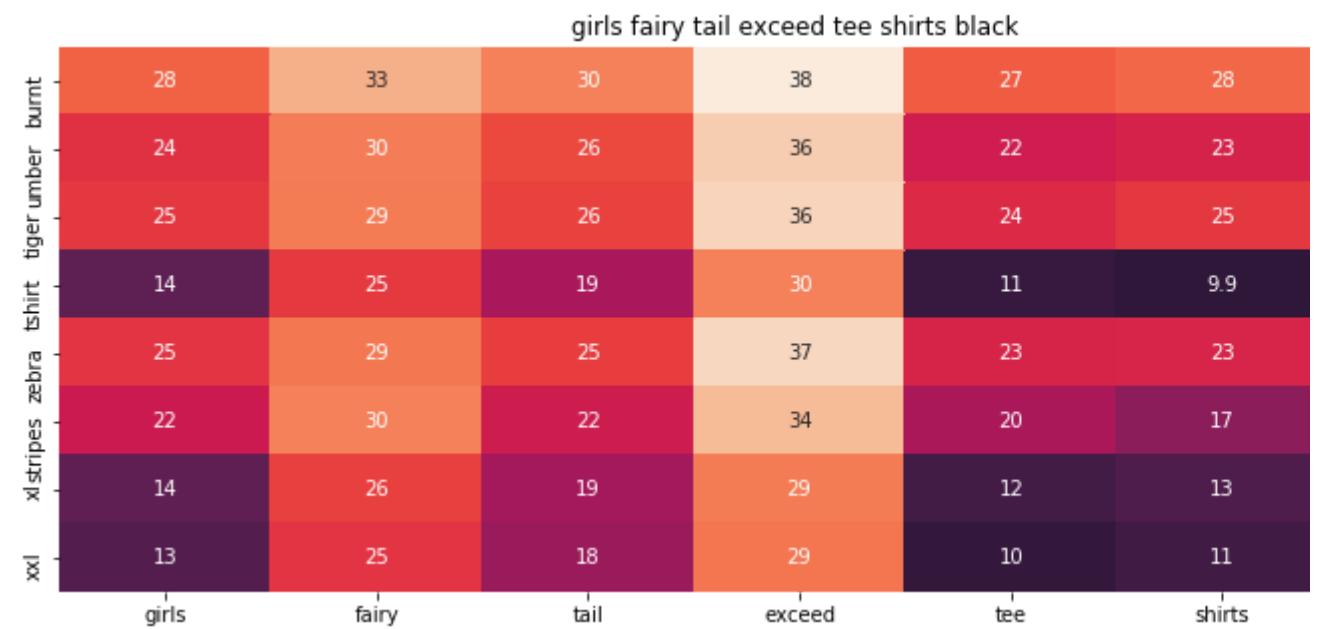
	usstore	women	stripes	oversized	beach	shirt	long	sleeve	casual	blouse	tee	tops
1state	26	26	30	29	28	26	26	26	27	26	27	26
tiger	21	22	23	24	26	22	22	22	22	21	22	22
umber	23	23	28	26	27	23	23	23	24	23	24	24
burnt	7	9	18	15	16	6	8.6	7.3	9.9	7.3	11	9.6
zebra	22	23	24	23	25	22	22	22	23	22	23	23
stripes	18	19	0	21	23	17	19	17	19	18	20	18
xl	7.2	9.3	19	16	17	8.4	8.5	8.3	11	8.4	12	9.8
xxl	4.7	7.5	18	15	15	6.3	6.6	6.6	8.9	6.2	10	8.1

ASIN : B01DNNI1RO

Brand : Usstore

euclidean distance from input : 25.116077171955645

=====

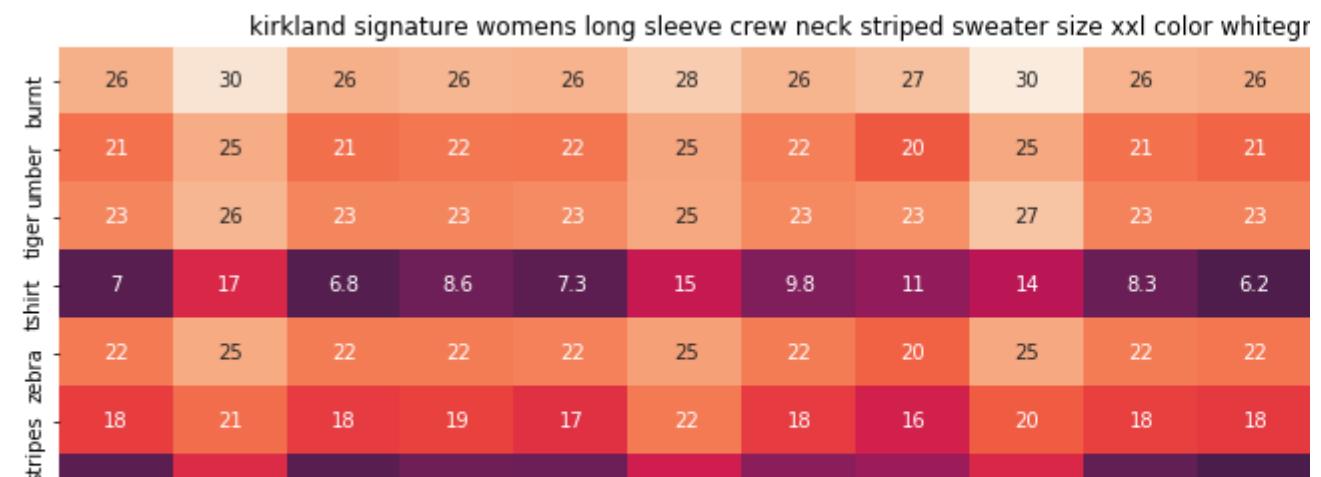


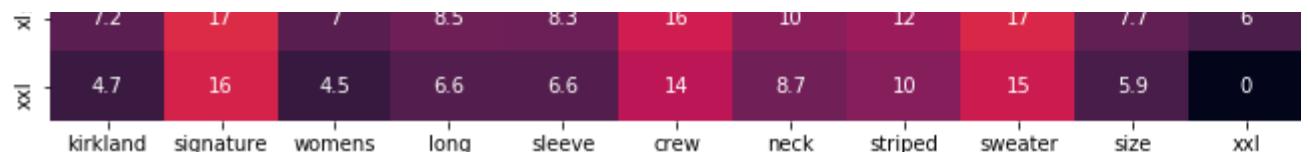
ASIN : B01L9F153U

Brand : ATYPEMX

euclidean distance from input : 25.259611147348046

=====





ASIN : B06XTPC3FP

Brand : Kirkland Signature

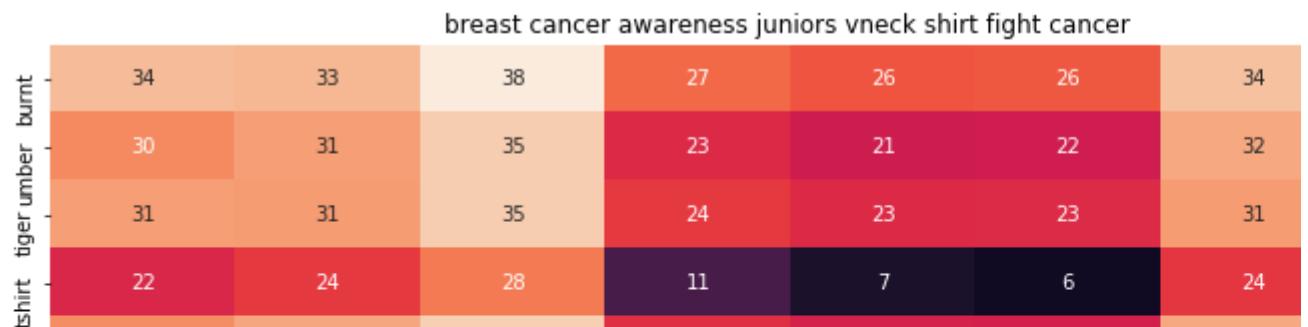
euclidean distance from input : 25.263297331832465



ASIN : B00JMAASRO

Brand : Wotefusi

euclidean distance from input : 25.277009383810842



	breast	cancer	awareness	juniors	vneck	shirt	fight
xxl	22	24	29	9.4	4.7	6.3	24
xl	23	24	29	11	7.2	8.4	24
large	28	30	33	20	18	17	28
medium	30	32	35	23	22	22	32

ASIN : B016CU40IY

Brand : Juiceclouds

euclidean distance from input : 25.287093489595204

	line	animal	print	tunic	top	pink	animal
burnt	29	30	26	28	26	26	30
tiger	24	27	22	22	21	20	27
umber	26	20	23	25	23	23	20
tshirt	14	18	9.7	12	7.3	8.1	18
zebra	24	21	22	23	22	21	21
stripes	20	24	19	19	18	17	24
xl	15	19	10	13	7.5	9.1	19
xxl	14	17	8.7	12	5	7.4	17

ASIN : B01LMJ5XYK

Brand : Romeo 4 Ever

euclidean distance from input : 25.29505585973226

```
# brand,color and image weight =50
# title vector weight = 5
```

```
idf_w2v_brand(1654, 5, 50, 20)
```



sophie finzi womens asymmetric point shirt sz 1x kiwi green 280079e									
sophie	finzi	womens	asymmetric	point	shirt	sz	1x	kiwi	green
0	11	11	21	19	12	13	17	31	
11	0	1.9	20	16	5.7	13	15	32	
1.9	1.9	0	20	16	5.8	12	15	31	
21	20	20	0	25	21	23	23	36	
19	16	16	25	0	16	19	21	34	
12	5.7	5.8	21	16	0	13	16	32	
13	13	12	23	19	13	0	17	32	
17	15	15	23	21	16	17	0	35	
31	32	31	36	34	32	32	35	0	
15	9.7	9.7	22	18	10	16	17	32	
11	0	1.9	20	16	5.7	13	15	32	

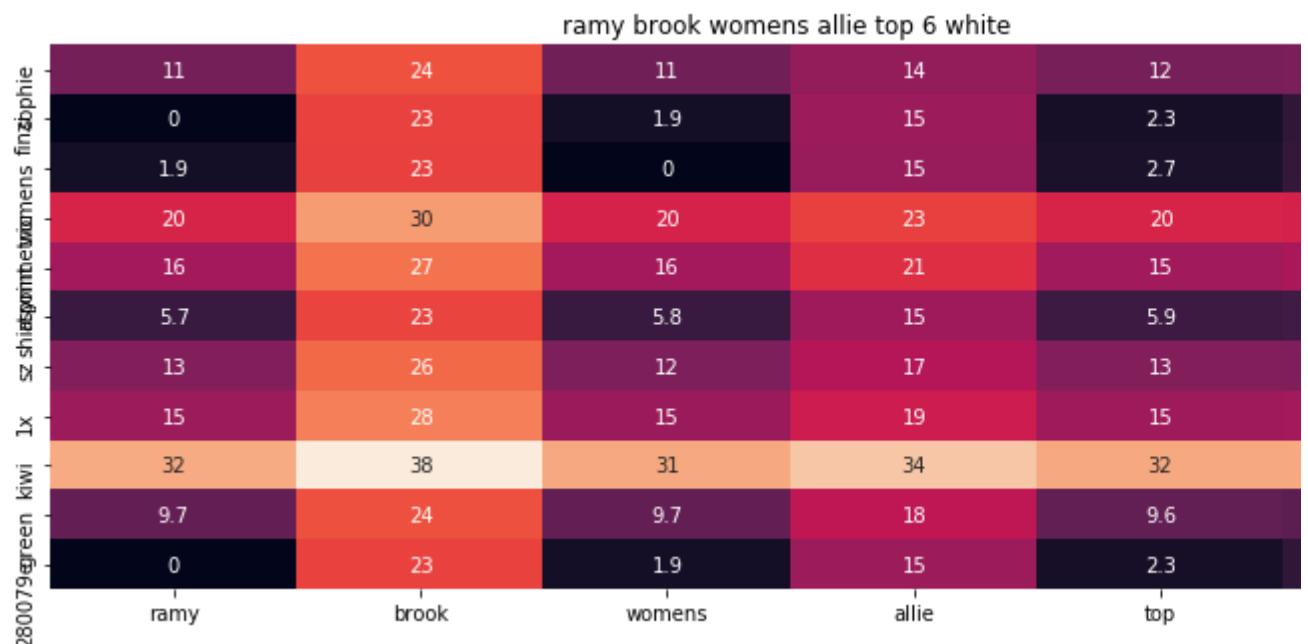
ASIN : B07533QTN4

Brand : Sophie Finzi

euclidean distance from input : 0.0

stanzino womens long sleeve graphic print plus size top fuchsia xl									
stanzino	womens	long	graphic	print	plus	size	top	fuchsia	xl
11	11	12	12	14	13	12	12	12	
0	1.9	4.7	5.8	11	8.4	5.8	4.9	2.3	
1.9	0	5	6.1	11	8.4	5.9	5	2.7	
20	20	20	20	21	21	20	20	20	
16	16	15	16	18	17	16	16	15	
5.7	5.8	7.4	6.1	11	9.4	8	7.5	5.9	
13	12	14	13	16	14	13	13	13	
15	15	15	15	18	17	15	15	15	
32	31	32	32	33	32	32	32	32	
9.7	9.7	10	10	14	12	11	10	9.6	
0	1.9	4.7	5.8	11	8.4	5.8	4.9	2.3	

28007 stanzino womens long sleeve graphic print plus size top  
ASIN : B00DP4VHWI  
Brand : Stanzino  
euclidean distance from input : 38.97896491774121  
=====



ASIN : B074T22HPN  
Brand : Ramy Brook  
euclidean distance from input : 39.10597561802268



	24	29	18	17	23	20	16	18	16	15	14	18
1x	24	29	18	17	23	20	16	18	16	15	14	18
kiwi	37	40	34	32	35	33	32	32	32	32	31	33
green	22	29	15	12	19	15	10	14	11	10	10	13
color	21	28	12	7.9	17	15	5.7	11	7.7	4.9	4.7	12
gh		bass	co	ladies	woven	plaid	shirt	knit	back	size	xxl	color

ASIN : B06XVB3DHT

Brand : Bass

euclidean distance from input : 40.17982211834641

	j brand womens pinstripe shirt xs blue				
1x	19	11	29	12	11
kiwi	16	1.9	28	5.7	6.4
green	16	0	27	5.8	6.3
blue	26	20	31	21	20
pink	22	16	31	16	16
yellow	16	5.8	26	0	7.8
black	20	12	28	13	11
white	22	15	31	16	14
orange	33	31	40	32	32
red	17	9.7	27	10	11
purple	16	1.9	28	5.7	6.4
grey	j	brand	womens	pinstripe	shirt

ASIN : B06XYYP1X1F

Brand : J Brand Jeans

euclidean distance from input : 40.50884978501607

	american rag womens boulder fest crown jewel combo large							
mens fin	15	17	11	35	31	25	11	16
phie	15	15	1.9	34	31	24	0	13
mens	15	15	0	34	31	23	1.9	13
fin	24	24	20	20	20	20	20	20

	24	24	20	39	37	30	20	22
24	21	21	16	37	34	27	16	19
16	14	5.8	34	31	23	5.7	14	
18	18	12	35	32	26	13	15	
1x	20	21	15	38	34	27	15	17
kiwi	33	34	31	44	43	38	32	33
green	17	17	9.7	35	32	25	9.7	16
9	15	15	1.9	34	31	24	0	13
american	rag	womens	boulder	fest	crown	jewl	comb	

ASIN : B0738J8RMC

Brand : American Rag

euclidean distance from input : 40.524460369120426

	fifth	degree	womens	fitness	exercise	gym	workout	shirts	jersey	1
fifth	22	26	11	26	25	24	26	16	20	
degree	19	24	1.9	24	23	21	24	12	17	
womens	19	24	0	24	23	20	24	12	17	
fitness	28	30	20	32	29	29	31	23	26	
exercise	22	26	16	29	27	25	28	19	22	
gym	19	25	5.8	25	23	20	24	8.4	14	
workout	22	27	12	28	26	24	27	16	21	
shirts	22	28	15	28	27	26	28	19	22	
jersey	37	40	31	39	38	37	39	33	34	
1	21	26	9.7	25	24	22	24	14	18	
1	19	24	1.9	24	23	21	24	12	17	
1	fifth	degree	womens	fitness	exercise	gym	workout	shirts	jersey	1

ASIN : B01M4LXFH0

Brand : Fifth Degree

euclidean distance from input : 40.54371925461317

	dsquared2	womens	white	long	sleeve	blouse	shirt	us
280079green kiwi	11	11	12	12	12	12	12	13
sz	0	1.9	4.9	4.7	5.8	5.8	5.7	7.8
shirts	1.9	0	5.1	5	6.1	5.6	5.8	8
pointe	20	20	20	20	20	20	21	21
womens	16	16	16	15	16	16	16	16
finzphie	5.7	5.8	6.4	7.4	6.1	5.2	0	9.2
280079green kiwi	13	12	12	14	13	13	13	16
1x	15	15	15	15	15	15	16	16
sz	32	31	31	32	32	32	32	32
shirts	9.7	9.7	9.1	10	10	11	10	12
pointe	0	1.9	4.9	4.7	5.8	5.8	5.7	7.8
womens	dsquared2	womens	white	long	sleeve	blouse	shirt	us

ASIN : B01758P216

Brand : DSQUARED2

euclidean distance from input : 40.710227910610755

	covergirl	activewear	womens	premium	long	line	racer	back	tank	xlarge	patriot	blue
280079green kiwi	30	39	11	22	12	18	22	14	13	12	11	
sz	31	39	1.9	20	4.7	13	20	7.7	6.5	2.3	0	
shirts	31	38	0	20	5	13	20	7.8	6.6	2.7	1.9	
pointe	35	42	20	27	20	23	28	22	21	20	20	
womens	35	42	16	24	15	18	25	15	16	15	16	
finzphie	31	38	5.8	21	7.4	14	21	8.9	8	5.9	5.7	
280079green kiwi	31	39	12	22	14	18	23	15	14	13	13	
1x	34	40	15	22	15	18	23	16	16	15	15	
sz	40	46	31	36	32	34	35	32	32	32	32	
shirts	32	39	9.7	21	10	16	22	11	11	9.6	9.7	
pointe	31	39	1.9	20	4.7	13	20	7.7	6.5	2.3	0	
womens	covergirl	activewear	womens	premium	long	line	racer	back	tank	top	xlarge	

ASIN : B07343JGVJ

Brand : Covergirl

euclidean distance from input : 41.02111839838225

	buffalo	david	bitton	nipaw	logo	graphic	tank	white	combo
280079green kiwi	23	20	11	11	18	14	13	12	11
sz shirts	20	23	0	0	14	11	6.5	4.9	7.1
280079green kiwi	20	23	1.9	1.9	14	11	6.6	5.1	7.1
sz shirts	29	28	20	20	24	21	21	20	21
280079green kiwi	25	28	16	16	21	18	16	16	17
sz shirts	21	23	5.7	5.7	13	11	8	6.4	7.1
280079green kiwi	23	23	13	13	18	16	14	12	13
1x	25	26	15	15	21	18	16	15	16
280079green kiwi	34	34	32	32	34	33	32	31	32
sz shirts	21	24	9.7	9.7	14	14	11	9.1	10
280079green kiwi	20	23	0	0	14	11	6.5	4.9	7.1

ASIN : B018H5AZXQ

Brand : Buffalo

euclidean distance from input : 41.04549807306019

	umgee	womens	umgee	u	neck	tee	tunic	wside	sl
280079green kiwi	11	11	11	13	15	15	11	31	31
sz shirts	0	1.9	0	8.2	9.8	12	0	31	31
280079green kiwi	1.9	0	1.9	8.3	9.8	12	1.9	30	30
sz shirts	20	20	20	21	22	22	20	32	32
280079green kiwi	16	16	16	17	18	20	16	33	33
sz shirts	5.7	5.8	5.7	8.5	9.8	11	5.7	30	30
280079green kiwi	13	12	13	14	16	16	13	31	31
1x	15	15	15	16	17	18	15	33	33
280079green kiwi	32	31	32	32	32	33	32	43	43
sz shirts	9.7	9.7	9.7	12	11	14	9.7	30	30
280079green kiwi	0	1.9	0	8.2	9.8	12	0	31	31

ASIN : B0725L97Z6

Brand : NRS

euclidean distance from input : 41.12023981221844

	french laundry womens ribbed tunic lace trim olive camo xl								
sz	shirts	laundry	womens	ribbed	tunic	lace	trim	olive	camo
280079green	17	21	11	18	15	13	18	20	21
kiwi	19	18	1.9	16	12	9.7	14	18	21
1x	18	18	0	16	12	9.4	14	18	21
23	25	27	20	23	22	20	23	25	27
20	24	23	16	22	20	19	20	24	26
21	19	18	5.8	16	11	9.7	14	18	21
22	20	21	12	18	16	14	17	19	22
23	23	23	15	21	18	17	20	23	26
24	35	36	31	34	33	32	33	33	34
25	20	19	9.7	18	14	13	15	17	20
26	19	18	1.9	16	12	9.7	14	18	21
27	french	laundry	womens	ribbed	tunic	lace	trim	olive	camo

ASIN : B06XNJHPDS

Brand : French Laundry

euclidean distance from input : 41.14539906302429

	eisbrecher eiszeit classic ladies v neck tshirt black					
sz	shirts	laundry	womens	ribbed	tunic	lace
280079green	11	11	15	12	13	11
kiwi	0	0	11	7.9	8.2	7
1x	1.9	1.9	12	7.1	8.3	6.8
20	20	20	21	21	21	21
21	16	16	18	17	17	17
22	5.7	5.7	12	9.1	8.5	6
23	13	13	16	13	14	13
24	15	15	18	17	16	15
25	32	32	33	32	32	31
26	9.7	9.7	14	12	12	11
27	eisbrecher	eiszeit	classic	ladies	v neck	tshirt
28	29	30	31	32	33	34
29	30	31	32	33	34	35
30	31	32	33	34	35	36
31	32	33	34	35	36	37
32	33	34	35	36	37	38
33	34	35	36	37	38	39
34	35	36	37	38	39	40
35	36	37	38	39	40	41
36	37	38	39	40	41	42
37	38	39	40	41	42	43
38	39	40	41	42	43	44
39	40	41	42	43	44	45
40	41	42	43	44	45	46
41	42	43	44	45	46	47
42	43	44	45	46	47	48
43	44	45	46	47	48	49
44	45	46	47	48	49	50
45	46	47	48	49	50	51
46	47	48	49	50	51	52
47	48	49	50	51	52	53
48	49	50	51	52	53	54
49	50	51	52	53	54	55
50	51	52	53	54	55	56
51	52	53	54	55	56	57
52	53	54	55	56	57	58
53	54	55	56	57	58	59
54	55	56	57	58	59	60
55	56	57	58	59	60	61
56	57	58	59	60	61	62
57	58	59	60	61	62	63
58	59	60	61	62	63	64
59	60	61	62	63	64	65
60	61	62	63	64	65	66
61	62	63	64	65	66	67
62	63	64	65	66	67	68
63	64	65	66	67	68	69
64	65	66	67	68	69	70
65	66	67	68	69	70	71
66	67	68	69	70	71	72
67	68	69	70	71	72	73
68	69	70	71	72	73	74
69	70	71	72	73	74	75
70	71	72	73	74	75	76
71	72	73	74	75	76	77
72	73	74	75	76	77	78
73	74	75	76	77	78	79
74	75	76	77	78	79	80
75	76	77	78	79	80	81
76	77	78	79	80	81	82
77	78	79	80	81	82	83
78	79	80	81	82	83	84
79	80	81	82	83	84	85
80	81	82	83	84	85	86
81	82	83	84	85	86	87
82	83	84	85	86	87	88
83	84	85	86	87	88	89
84	85	86	87	88	89	90
85	86	87	88	89	90	91
86	87	88	89	90	91	92
87	88	89	90	91	92	93
88	89	90	91	92	93	94
89	90	91	92	93	94	95
90	91	92	93	94	95	96
91	92	93	94	95	96	97
92	93	94	95	96	97	98
93	94	95	96	97	98	99
94	95	96	97	98	99	100
95	96	97	98	99	100	101
96	97	98	99	100	101	102
97	98	99	100	101	102	103
98	99	100	101	102	103	104
99	100	101	102	103	104	105
100	101	102	103	104	105	106
101	102	103	104	105	106	107
102	103	104	105	106	107	108
103	104	105	106	107	108	109
104	105	106	107	108	109	110
105	106	107	108	109	110	111
106	107	108	109	110	111	112
107	108	109	110	111	112	113
108	109	110	111	112	113	114
109	110	111	112	113	114	115
110	111	112	113	114	115	116
111	112	113	114	115	116	117
112	113	114	115	116	117	118
113	114	115	116	117	118	119
114	115	116	117	118	119	120
115	116	117	118	119	120	121
116	117	118	119	120	121	122
117	118	119	120	121	122	123
118	119	120	121	122	123	124
119	120	121	122	123	124	125
120	121	122	123	124	125	126
121	122	123	124	125	126	127
122	123	124	125	126	127	128
123	124	125	126	127	128	129
124	125	126	127	128	129	130
125	126	127	128	129	130	131
126	127	128	129	130	131	132
127	128	129	130	131	132	133
128	129	130	131	132	133	134
129	130	131	132	133	134	135
130	131	132	133	134	135	136
131	132	133	134	135	136	137
132	133	134	135	136	137	138
133	134	135	136	137	138	139
134	135	136	137	138	139	140
135	136	137	138	139	140	141
136	137	138	139	140	141	142
137	138	139	140	141	142	143
138	139	140	141	142	143	144
139	140	141	142	143	144	145
140	141	142	143	144	145	146
141	142	143	144	145	146	147
142	143	144	145	146	147	148
143	144	145	146	147	148	149
144	145	146	147	148	149	150
145	146	147	148	149	150	151
146	147	148	149	150	151	152
147	148	149	150	151	152	153
148	149	150	151	152	153	154
149	150	151	152	153	154	155
150	151	152	153	154	155	156
151	152	153	154	155	156	157
152	153	154	155	156	157	158
153	154	155	156	157	158	159
154	155	156	157	158	159	160
155	156	157	158	159	160	161
156	157	158	159	160	161	162
157	158	159	160	161	162	163
158	159	160	161	162	163	164
159	160	161	162	163	164	165
160	161	162	163	164	165	166
161	162	163	164	165	166	167
162	163	164	165	166	167	168
163	164	165	166	167	168	169
164	165	166	167	168	169	170
165	166	167	168	169	170	171
166	167	168	169	170	171	172
167	168	169	170	171	172	173
168	169	170	171	172	173	174
169	170	171	172	173	174	175
170	171	172	173	174	175	176
171	172	173	174	175	176	177
172	173	174	175	176	177	178
173	174	175	176	177	178	179
174	175	176	177	178	179	180
175	176	177	178	179	180	181
176	177	178	179	180	181	182
177	178	179	180	181	182	183
178	179	180	181	182	183	184
179	180	181	182	183	184	185
180	181	182	183	184	185	186
181	182</td					

280079gregre	0	0	11	7.9	8.2	7
	eisbrecher	eiszeit	classic	ladies	v	neck

ASIN : B01IT9KKBC

Brand : Bunny Angle

euclidean distance from input : 41.201075010351076

=====

authentic pigment ladies french terry crew black xxlarge

21	28	12	17	20	18	12
18	27	7.9	19	23	13	4.5
18	27	7.1	18	22	14	4.6
26	31	21	25	29	25	20
23	30	17	24	27	20	16
18	27	9.1	19	22	14	6.3
22	28	13	20	22	18	12
23	30	17	23	25	20	15
37	39	32	35	37	33	31
20	27	12	20	24	16	9.4
18	27	7.9	19	23	13	4.5
authentic	pigment	ladies	french	terry	crew	black

ASIN : B01GESXRTC

Brand : Authentic Pigment

euclidean distance from input : 41.27643632812717

=====

acevog women asymmetric boat neck shoulder long sleeve hilow blouse

11	12	21	22	13	16	12	12	11
0	6.3	20	18	8.2	12	4.7	5.8	5.9
1.9	5.2	20	18	8.3	12	5	6.1	5.9
20	21	0	27	21	22	20	20	21
16	16	25	23	17	18	15	16	17
5.7	8.2	21	18	8.5	12	7.4	6.1	5.9
hilow	blouse	acevog	women	asymmetric	boat	neck	shoulder	long
hi	low	blouse	women	asymmetric	boat	neck	shoulder	long

sz	13	13	23	22	14	17	14	13	
1x	15	16	23	23	16	18	15	15	
kiwi	32	32	36	35	32	33	32	32	
green	9.7	11	22	20	12	14	10	10	9
acevog	0	6.3	20	18	8.2	12	4.7	5.8	hi
women									
asymmetric									
boat									
neck									
shoulder									
long									
sleeve									
hi									

ASIN : B01CE4GOG8

Brand : ACEVOG

euclidean distance from input : 41.27762497785919

=====

	lush	white	womens	large	stripe	high	low	tank	blouse	black	l
sz	21	12	11	13	18	15	17	13			
1x	18	4.9	1.9	5.8	14	11	13	6.5			
kiwi	19	5.1	0	6.1	14	11	13	6.6			
green	26	20	20	20	23	22	22	21			
acevog	25	16	16	16	17	17	18	16			
acevog	19	6.4	5.8	8.2	14	12	14	8			
women	21	12	12	13	17	16	17	14			
1x	24	15	15	16	19	17	18	16			
kiwi	35	31	31	32	33	32	32	32			
green	19	9.1	9.7	11	15	13	15	11			
acevog	18	4.9	1.9	5.8	14	11	13	6.5			
women	lush	white	womens	large	stripe	high	low	tank	blouse	black	l

ASIN : B07285DZ7S

Brand : Lush Clothing

euclidean distance from input : 41.361609474953354

=====

finzbphie	18	25	11	16	12	14	12	13	
finzbphie	16	23	0	11	2.3	8.7	4.9	6.7	

	16	23	19	11	2.7	8.4	5	6.8	ro
sz	25	31	20	22	20	22	20	21	22
shirts	22	28	16	18	15	17	16	17	18
spacedyed	16	23	5.7	13	5.9	10	7.5	8.8	10
performance	20	27	13	17	13	13	13	15	17
top	22	28	15	18	15	16	15	16	18
juniors	35	38	32	33	32	32	32	32	35
size	18	24	9.7	15	9.6	13	10	10	12
new	16	23	0	11	2.3	8.7	4.9	6.7	10
ro									

ASIN : B01N2SWN8U

Brand : Inspired Hearts

euclidean distance from input : 41.5337983377732

---

**bcx womens juniors sleeveless necklace shirt coral xs**

	11	11	14	14	22	12	20
sz	0	1.9	8.7	11	21	5.7	18
shirts	1.9	0	8.4	10	21	5.8	17
womens	20	20	22	20	28	21	25
juniors	16	16	17	19	25	16	23
sleeveless	5.7	5.8	10	9.5	19	0	18
necklace	13	12	13	15	23	13	20
shirt	15	15	16	18	25	16	22
coral	32	31	32	33	36	32	32
bcx	9.7	9.7	13	13	22	10	19
ro	0	1.9	8.7	11	21	5.7	18

ASIN : B071RQKPKF

Brand : BCX

euclidean distance from input : 41.636849939721046

	usstore	women	stripes	oversized	beach	shirt	long	sleeve	casual	blouse	tee	tops
280079green kiwi	11	12	21	19	19	12	12	12	13	12	15	13
sz	0	6.3	18	15	16	5.7	4.7	5.8	8.6	5.8	9.8	7.8
shirtsizes	1.9	5.2	18	15	15	5.8	5	6.1	8.5	5.6	9.8	7.6
280079green kiwi	20	21	24	22	26	21	20	20	21	20	22	21
sz	16	16	23	21	21	16	15	16	17	16	18	17
shirtsizes	5.7	8.2	17	15	16	0	7.4	6.1	9.5	5.2	9.8	8.7
280079green kiwi	13	13	21	18	20	13	14	13	15	13	16	13
sz	15	16	23	21	21	16	15	15	17	15	17	16
shirtsizes	32	32	35	35	33	32	32	32	32	32	32	32
280079green kiwi	9.7	11	17	17	17	10	10	10	13	11	11	11
sz	0	6.3	18	15	16	5.7	4.7	5.8	8.6	5.8	9.8	7.8

ASIN : B01DNNI1RO

Brand : Usstore

euclidean distance from input : 41.649860077044124

	ro	de	womens	small	button	print	blouse	shirt
280079green kiwi	18	20	11	12	16	13	12	12
sz	19	20	1.9	5	13	8.4	5.8	5.7
shirtsizes	18	20	0	5.2	13	8.4	5.6	5.8
280079green kiwi	26	26	20	20	23	21	20	21
sz	25	24	16	16	19	17	16	16
shirtsizes	19	20	5.8	7.5	12	9.4	5.2	0
280079green kiwi	19	22	12	13	17	14	13	13
sz	19	22	12	13	17	14	13	13
shirtsizes	22	24	15	16	17	17	15	16
280079green kiwi	36	38	31	32	33	32	32	32
sz	21	22	9.7	9.9	14	12	11	10
shirtsizes	19	20	1.9	5	13	8.4	5.8	5.7

ASIN : B01N7IZW2Z

Brand : Rode

euclidean distance from input : 41.68785225172645

```
# brand,color and image weight =20
# title vector weight = 10
```

```
idf_w2v_brand(1654, 10, 20, 20)
```



sophie finzi womens asymmetric point shirt sz 1x kiwi green 280079e									
sophie	finzi	womens	asymmetric	point	shirt	sz	1x	kiwi	green
0	11	11	21	19	12	13	17	31	
11	0	1.9	20	16	5.7	13	15	32	
11	1.9	0	20	16	5.8	12	15	31	
21	20	20	0	25	21	23	23	36	
19	16	16	25	0	16	19	21	34	
12	5.7	5.8	21	16	0	13	16	32	
13	13	12	23	19	13	0	17	32	
17	15	15	23	21	16	17	0	35	
31	32	31	36	34	32	32	35	0	
15	9.7	9.7	22	18	10	16	17	32	
11	0	1.9	20	16	5.7	13	15	32	

ASIN : B07533QTN4

Brand : Sophie Finzi

euclidean distance from input : 0.0

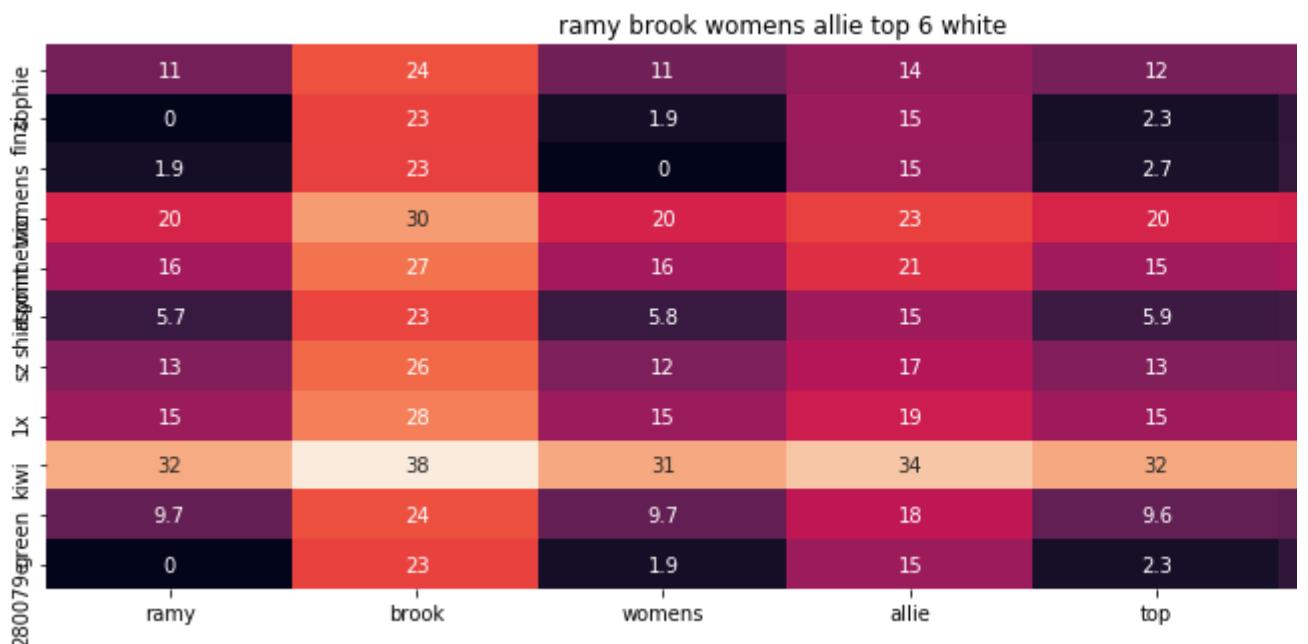
stanzino womens long sleeve graphic print plus size top fuchsia xl									
stanzino	womens	long	sleeve	graphic	print	plus	size	top	fuchsia
11	11	12	12	14	13	12	12	12	
0	1.9	4.7	5.8	11	8.4	5.8	4.9	2.3	
1.9	0	5	6.1	11	8.4	5.9	5	2.7	
20	20	20	20	21	21	20	20	20	
16	16	15	16	18	17	16	16	15	
5.7	5.8	7.4	6.1	11	9.4	8	7.5	5.9	
13	12	14	13	16	14	13	13	13	
15	15	15	15	18	17	15	15	15	
32	31	32	32	33	32	32	32	32	
9.7	9.7	10	10	14	12	11	10	9.6	
0	1.9	4.7	5.8	11	8.4	5.8	4.9	2.3	

2800079green	stanzino	womens	long	sleeve	graphic	print	plus	size	top	fi
--------------	----------	--------	------	--------	---------	-------	------	------	-----	----

ASIN : B00DP4VHWI

Brand : Stanzino

euclidean distance from input : 29.873087863505013



ASIN : B074T22HPN

Brand : Ramy Brook

euclidean distance from input : 30.232872517018723



1x	24	29	18	17	23	20	16	18	16	15	14	18
kiwi	37	40	34	32	35	33	32	32	32	32	31	33
green	22	29	15	12	19	15	10	14	11	10	10	13
280079	21	28	12	7.9	17	15	5.7	11	7.7	4.9	4.7	12
gh	bass	co	ladies	woven	plaid	shirt	knit	back	size	xxl	color	

ASIN : B06XVB3DHT

Brand : Bass

euclidean distance from input : 30.961254827705332

dsquared2 womens white long sleeve blouse shirt us 40									
finzphie	11	11	12	12	12	12	12	12	13
sz	0	1.9	4.9	4.7	5.8	5.8	5.7	7.8	
shirtpoint	1.9	0	5.1	5	6.1	5.6	5.8	8	
womens	20	20	20	20	20	20	21	21	
280079	16	16	16	15	16	16	16	16	
dsquared2	5.7	5.8	6.4	7.4	6.1	5.2	0	9.2	
white	13	12	12	14	13	13	13	16	
long	15	15	15	15	15	15	16	16	
sleeve	32	31	31	32	32	32	32	32	
blouse	9.7	9.7	9.1	10	10	11	10	12	
shirt	0	1.9	4.9	4.7	5.8	5.8	5.7	7.8	
us	dsquared2	womens	white	long	sleeve	blouse	shirt		

ASIN : B01758P216

Brand : DSQUARED2

euclidean distance from input : 31.201177937670543

j brand womens pinstripe shirt xs blue					
mens finzphie	19	11	29	12	11
	16	1.9	28	5.7	6.4
	16	0	27	5.8	6.3
	26	20	21	21	20

	26	20	51	21	20
sz	22	16	31	16	16
shirts	16	5.8	26	0	7.8
jeans	20	12	28	13	11
lx	22	15	31	16	14
kiwi	33	31	40	32	32
green	17	9.7	27	10	11
079	16	1.9	28	5.7	6.4
j		brand	womens	pinstripe	shirt

ASIN : B06XYP1X1F

Brand : J Brand Jeans

euclidean distance from input : 31.40246782899876

	buffalo david bitton nipaw logo graphic tank white combo xxl								
sz	23	20	11	11	18	14	13	12	
shirts	20	23	0	0	14	11	6.5	4.9	
womens	20	23	1.9	1.9	14	11	6.6	5.1	
finaphe	29	28	20	20	24	21	21	20	
079	25	28	16	16	21	18	16	16	
green	21	23	5.7	5.7	13	11	8	6.4	
079	23	23	13	13	18	16	14	12	
kiwi	25	26	15	15	21	18	16	15	
079	34	34	32	32	34	33	32	31	
black	21	24	9.7	9.7	14	14	11	9.1	
white	20	23	0	0	14	11	6.5	4.9	
combo		buffalo	david	bitton	nipaw	logo	graphic	tank	white

ASIN : B018H5AZXQ

Brand : Buffalo

euclidean distance from input : 31.53501279589518

	eisbrecher	eiszeit	classic	ladies	v	neck
280079green kiwi	11	11	15	12	13	11
sz	0	0	11	7.9	8.2	7
shirts pointelle	1.9	1.9	12	7.1	8.3	6.8
womens finzphie	20	20	21	21	21	21
1x	16	16	18	17	17	17
1x	5.7	5.7	12	9.1	8.5	6
sz	13	13	16	13	14	13
shirts pointelle	15	15	18	17	16	15
womens finzphie	32	32	33	32	32	31
1x	9.7	9.7	14	12	12	11
280079green kiwi	0	0	11	7.9	8.2	7

ASIN : B01IT9KKBC

Brand : Bunny Angle

euclidean distance from input : 31.556996735292614

	acevog	women	asymmetric	boat	neck	shoulder	long	sleeve	hi
280079green kiwi	11	12	21	22	13	16	12	12	11
sz	0	6.3	20	18	8.2	12	4.7	5.8	10
shirts pointelle	1.9	5.2	20	18	8.3	12	5	6.1	11
womens finzphie	20	21	0	27	21	22	20	20	21
1x	16	16	25	23	17	18	15	16	17
1x	5.7	8.2	21	18	8.5	12	7.4	6.1	9
sz	13	13	23	22	14	17	14	13	14
shirts pointelle	15	16	23	23	16	18	15	15	16
womens finzphie	32	32	36	35	32	33	32	32	33
1x	9.7	11	22	20	12	14	10	10	9
280079green kiwi	0	6.3	20	18	8.2	12	4.7	5.8	10

ASIN : B01CE4GOG8

Brand : ACEVOG

euclidean distance from input : 31.597731258177472

	umgee	womens	umgee	u	neck	tee	tunic	wside	sl
sz	11	11	11	13	15	15	11	31	31
shirts	0	19	0	8.2	9.8	12	0	31	31
pointe	1.9	0	1.9	8.3	9.8	12	1.9	30	30
womens	20	20	20	21	22	22	20	32	32
fin	16	16	16	17	18	20	16	33	33
phie	5.7	5.8	5.7	8.5	9.8	11	5.7	30	30
sz	13	12	13	14	16	16	13	31	31
1x	15	15	15	16	17	18	15	33	33
kiwi	32	31	32	32	32	33	32	43	43
green	9.7	9.7	9.7	12	11	14	9.7	30	30
79	0	1.9	0	8.2	9.8	12	0	31	31

ASIN : B0725L97Z6

Brand : NRS

euclidean distance from input : 31.662463887195873

	lush	white	womens	large	stripe	high	low	tank	blc
sz	21	12	11	13	18	15	17	13	13
shirts	18	4.9	1.9	5.8	14	11	13	6.5	5.5
pointe	19	5.1	0	6.1	14	11	13	6.6	5.5
womens	26	20	20	20	23	22	22	21	21
fin	25	16	16	16	17	17	18	16	16
phie	19	6.4	5.8	8.2	14	12	14	8	5.5
sz	21	12	12	13	17	16	17	14	14
1x	24	15	15	16	19	17	18	16	16
kiwi	35	31	31	32	33	32	32	32	32
green	19	9.1	9.7	11	15	13	15	11	11
79	18	4.9	1.9	5.8	14	11	13	6.5	5.5

ASIN : B07285DZ7S

Brand : Lush Clothing

euclidean distance from input : 31.70653353277179

	american	rag	womens	boulder	fest	crown	jewl	large
280079green kiwi	15	17	11	35	31	25	11	16
sz shirtgymnastics finnbphie	15	15	1.9	34	31	24	0	13
1x	15	15	0	34	31	23	1.9	13
en	24	24	20	39	37	30	20	22
en	21	21	16	37	34	27	16	19
en	16	14	5.8	34	31	23	5.7	14
en	18	18	12	35	32	26	13	15
en	20	21	15	38	34	27	15	17
en	33	34	31	44	43	38	32	33
en	17	17	9.7	35	32	25	9.7	16
en	15	15	1.9	34	31	24	0	13

ASIN : B0738J8RMC

Brand : American Rag

euclidean distance from input : 31.710670961769043

	covergirl	activewear	womens	premium	long	line	racer	back	tank	top	xlarge	patriot	blu
280079green kiwi	30	39	11	22	12	18	22	14	13	12	11		
sz shirtgymnastics finnbphie	31	39	1.9	20	4.7	13	20	7.7	6.5	2.3	0		
1x	31	38	0	20	5	13	20	7.8	6.6	2.7	1.9		
en	35	42	20	27	20	23	28	22	21	20	20		
en	35	42	16	24	15	18	25	15	16	15	16		
en	31	38	5.8	21	7.4	14	21	8.9	8	5.9	5.7		
en	31	39	12	22	14	18	23	15	14	13	13		
en	34	40	15	22	15	18	23	16	16	15	15		
en	40	46	31	36	32	34	35	32	32	32	32		
en	32	39	9.7	21	10	16	22	11	11	9.6	9.7		



ASIN : B07343JGVJ

Brand : Covergirl

euclidean distance from input : 31.83352454520688

=====

french laundry womens ribbed tunic lace trim olive camo xl



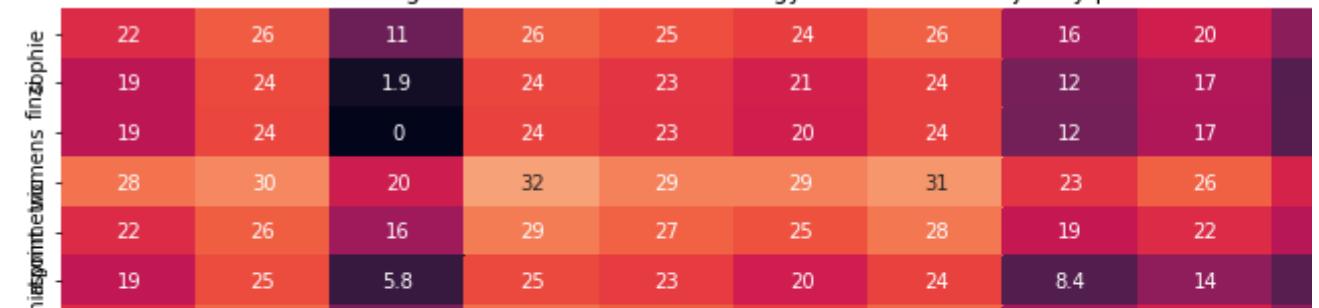
ASIN : B06XNJHPDS

Brand : French Laundry

euclidean distance from input : 31.908886721738654

=====

fifth degree womens fitness exercise gym workout shirts jersey printed tshirt l



	22	27	12	28	26	24	27	16	21	
sz	22	28	15	28	27	26	28	19	22	
1x	37	40	31	39	38	37	39	33	34	
kiwi	21	26	9.7	25	24	22	24	14	18	
green	19	24	1.9	24	23	21	24	12	17	
fifth	degree	womens	fitness	exercise	gym	workout	shirts	jersey	tops	

ASIN : B01M4LXFH0

Brand : Fifth Degree

euclidean distance from input : 31.910234207940285

	usstore	women	stripes	oversized	beach	shirt	long sleeve	casual	blouse	tee	tops	
sz	11	12	21	19	19	12	12	12	13	12	15	13
z	0	6.3	18	15	16	5.7	4.7	5.8	8.6	5.8	9.8	7.8
sz	1.9	5.2	18	15	15	5.8	5	6.1	8.5	5.6	9.8	7.6
z	20	21	24	22	26	21	20	20	21	20	22	21
sz	16	16	23	21	21	16	15	16	17	16	18	17
z	5.7	8.2	17	15	16	0	7.4	6.1	9.5	5.2	9.8	8.7
sz	13	13	21	18	20	13	14	13	15	13	16	13
z	15	16	23	21	21	16	15	15	17	15	17	16
sz	32	32	35	35	33	32	32	32	32	32	32	32
z	9.7	11	17	17	17	10	10	10	13	11	11	11
sz	0	6.3	18	15	16	5.7	4.7	5.8	8.6	5.8	9.8	7.8
usstore	women	stripes	oversized	beach	shirt	long	sleeve	casual	blouse	tee	tops	

ASIN : B01DNNI1RO

Brand : Usstore

euclidean distance from input : 31.927589345398765

	ro	de	womens	small	button	print	blouse	shirt	black
sz	18	20	11	12	16	13	12	12	12
z	19	20	1.9	5	13	8.4	5.8	5.8	5.7
fin	zphie								

	18	20	0	5.2	13	8.4	5.6	5.8
	26	26	20	20	23	21	20	21
	25	24	16	16	19	17	16	16
	19	20	5.8	7.5	12	9.4	5.2	0
	19	22	12	13	17	14	13	13
Ix	22	24	15	16	17	17	15	16
kiwi	36	38	31	32	33	32	32	32
green	21	22	9.7	9.9	14	12	11	10
280079	19	20	1.9	5	13	8.4	5.8	5.7
ro	de	womens	small	button	print	blouse	shirt	

ASIN : B01N7IZW2Z

Brand : Rode

euclidean distance from input : 32.031513811096794

bcx womens juniors sleeveless necklace shirt coral xs							
	11	11	14	14	22	12	20
	0	1.9	8.7	11	21	5.7	18
	1.9	0	8.4	10	21	5.8	17
	20	20	22	20	28	21	25
	16	16	17	19	25	16	23
	5.7	5.8	10	9.5	19	0	18
Ix	13	12	13	15	23	13	20
kiwi	15	15	16	18	25	16	22
green	32	31	32	33	36	32	32
280079	9.7	9.7	13	13	22	10	19
ro	0	1.9	8.7	11	21	5.7	18
bcx	womens	juniors	sleeveless	necklace	shirt		coral

ASIN : B071RQKPKF

Brand : BCX

euclidean distance from input : 32.03696774713658

	como	vintage	womens	vneck	heart	overprint	tiered	bell	sleeve	blouse	top	bluewhit
280079green kiwi	33	16	11	11	19	29	22	20	12	12	12	11
sz	34	12	1.9	0	15	29	19	18	5.8	5.8	2.3	0
280079green kiwi	34	12	0	1.9	15	29	19	18	6.1	5.6	2.7	1.9
1x	37	22	20	20	25	30	23	27	20	20	20	20
280079green kiwi	38	19	16	16	20	33	23	23	16	16	15	16
280079green kiwi	34	13	5.8	5.7	15	29	20	18	6.1	5.2	5.9	5.7
280079green kiwi	33	15	12	13	20	28	23	22	13	13	13	13
280079green kiwi	36	18	15	15	21	29	23	23	15	15	15	15
280079green kiwi	46	33	31	32	35	39	36	35	32	32	32	32
280079green kiwi	36	15	9.7	9.7	17	29	20	20	10	11	9.6	9.7
280079green kiwi	34	12	1.9	0	15	29	19	18	5.8	5.8	2.3	0

ASIN : B0741FBSHW

Brand : Como vintage

euclidean distance from input : 32.038587056826984

	inspired	hearts	spacedyed	performance	top	juniors	size	new	royal aqua
280079green kiwi	18	25	11	16	12	14	12	13	1
sz	16	23	0	11	2.3	8.7	4.9	6.7	1
280079green kiwi	16	23	1.9	11	2.7	8.4	5	6.8	1
280079green kiwi	25	31	20	22	20	22	20	21	1
280079green kiwi	22	28	16	18	15	17	16	17	1
280079green kiwi	16	23	5.7	13	5.9	10	7.5	8.8	1
280079green kiwi	20	27	13	17	13	13	13	15	1
280079green kiwi	22	28	15	18	15	16	15	16	1
280079green kiwi	35	38	32	33	32	32	32	32	1
280079green kiwi	18	24	9.7	15	9.6	13	10	10	1
280079green kiwi	16	23	0	11	2.3	8.7	4.9	6.7	1

ASIN : B01N2SWN8U

Brand : Inspired Hearts

euclidean distance from input : 32.06276233572118

## ▼ Conclusion:

1. First we do some Pre-Processings steps like removing duplicates and text pre-processing. After pre- processing we do some feature engineering.
2. We convert the title text into vector using Bag of Words and TF-IDF Vectorization. After that we do text based product similarity.
3. We have small problem with tf-idf. In our title text mostly the words are not repeated so TF (Term Frequency) based product similarity.
4. We do semantic based (Avg W2V and IDF weighted W2V) product similarity.
5. We do feature engineering the brand and color using one-hot encoding.
6. We convert image into vector using deep learning techniques CNN with VGG-16.
7. We simply combine these four features (IDF Weighted W2V + Brand + Color + Image).
8. We give different weights to features and do features based similarity.