# Amazon Apparel Recommendations

## [4.2] Data and Code:

https://drive.google.com/open?id=0BwNkduBnePt2VWhCYXhMV3p4dTg (https://drive.google.com/open?id=0BwNkduBnePt2VWhCYXhMV3p4dTg)

## [4.3] Overview of the data

In [2]:
```
#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
from sklearn.externals import joblib
from IPython.display import display, Image, SVG, Math, YouTubeVideo
'''
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")
'''
```

Out[2]:
```
'\nimport plotly\nimport plotly.figure_factory as ff\nfrom plotly.graph_obj
s import Scatter, Layout\n\nplotly.offline.init_notebook_mode(connected=Tru
e)\nwarnings.filterwarnings("ignore")\n'
```

In [0]:
```
# 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')
```

In [0]:
```
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

In [0]:
```
# each product/item has 19 features in the raw dataset.
data.columns # prints column-names or feature-names.
```

Out[0]:
```
Index(['asin', 'author', 'availability', 'availability_type', 'brand', 'col
or',
       'editorial_reivew', 'editorial_review', 'formatted_price',
       'large_image_url', 'manufacturer', 'medium_image_url', 'model',
       'product_type_name', 'publisher', 'reviews', 'sku', 'small_image_ur
l',
       'title'],
      dtype='object')
```

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

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

In [0]:
```
data = data[['asin', 'brand', 'color', 'medium_image_url', 'product_type_na
me', 'title', 'formatted_price']]
```

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

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

Out[0]:

| | asin | brand | color | medium_image_url | product_type_name | t |
|---|---|---|---|---|---|---|
| **0** | B016I2TS4W | FNC7C | None | https://images-na.ssl-images-amazon.com/images... | SHIRT | Minions Como Superher Ironman Long Slee R... |
| **1** | B01N49AI08 | FIG Clothing | None | https://images-na.ssl-images-amazon.com/images... | SHIRT | FIG Cloth Womens Tunic |
| **2** | B01JDPCOHO | FIG Clothing | None | https://images-na.ssl-images-amazon.com/images... | SHIRT | FIG Cloth Womens Won Top |
| **3** | B01N19U5H5 | Focal18 | None | https://images-na.ssl-images-amazon.com/images... | SHIRT | Focal18 Sailor Col Bubble Sleeve Blouse Sh |
| **4** | B004GSI2OS | FeatherLite | Onyx Black/ Stone | https://images-na.ssl-images-amazon.com/images... | SHIRT | Featherlit Ladies' Lc Sleeve St Resistan.. |

## [5.1] Missing data for various features.

#### Basic stats for the feature: product_type_name

In [0]:
```
# 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
```

In [0]:
```
# 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']
```

In [0]:
```
# find the 10 most frequent product_type_names.
product_type_count = Counter(list(data['product_type_name']))
product_type_count.most_common(10)
```

Out[0]:
```
[('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

In [0]:
```
# 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
```

In [0]:
```
brand_count = Counter(list(data['brand']))
brand_count.most_common(10)
```

Out[0]:
```
[('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

In [0]:
```
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
```

In [0]:
```
color_count = Counter(list(data['color']))
color_count.most_common(10)
```

Out[0]:
```
[(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

In [0]:

```
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
```

In [0]:

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

Out[0]:

```
[(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**

In [0]:

```
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
```

In [0]:

```
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 takes too long to run on your laptop, you may use the pickle files we give you to speed things up.

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

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

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

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

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

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

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

In [0]:
```
data.to_pickle('pickels/28k_apparel_data')
```

In [0]:
```
# 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')


'''
```

Out[0]:
```
"\nfrom PIL import Image\nimport requests\nfrom io import BytesIO\n\nfor in
dex, row in images.iterrows():\n        url = row['large_image_url']\n
response = requests.get(url)\n        img = Image.open(BytesIO(response.con
tent))\n        img.save('workshop/images/28k_images/'+row['asin']+'.jpeg')
\n\n\n"
```

## [5.2] Remove near duplicate items

### [5.2.1] Understand about duplicates.

In [0]:
```
# 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
```

#### These shirts are exactly same except in size (S, M,L,XL)

| | |
|---|---|
| :B00AQ4GMCK | :B00AQ4GMTS |
| :B00AQ4GMLQ | :B00AQ4GN3I |

#### These shirts exactly same except in color

| | |
|---|---|
| :B00G278GZ6 | :B00G278W6O |
| :B00G278Z2A | :B00G2786X8 |

#### In our data there are many duplicate products like the above examples, we need to de-dupe them for better results.

### [5.2.2] Remove duplicates : Part 1

In [0]:
```
# read data from pickle file from previous stage
data = pd.read_pickle('pickels/28k_apparel_data')
```

In [0]:
```
data.head()
```

Out[0]:

| | asin | brand | color | medium_image_url | product_type_name | |
|---|---|---|---|---|---|---|
| **4** | B004GSI2OS | FeatherLite | Onyx Black/ Stone | https://images-na.ssl-images-amazon.com/images... | SHIRT | Featherl Ladies' l Sleeve S Resistar |
| **6** | B012YX2ZPI | HX-Kingdom Fashion T-shirts | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | Women' Unique 100% C T - Spec Olympic |
| **11** | B001LOUGE4 | Fitness Etc. | Black | https://images-na.ssl-images-amazon.com/images... | SHIRT | Ladies Cotton T 2x1 Ribb Tank To |
| **15** | B003BSRPB0 | FeatherLite | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | Featherl Ladies' Moisture Free Me Sport S. |
| **21** | B014ICEDNA | FNC7C | Purple | https://images-na.ssl-images-amazon.com/images... | SHIRT | Superna Chibis S Dean Ar Castiel Short... |

In [0]:
```
# 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.shap
e[0])
```

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

In [0]:
```
# Sort the whole data based on title (alphabetical order of title)
data_sorted.sort_values('title',inplace=True, ascending=False)
data_sorted.head()
```

Out[0]:

| | asin | brand | color | medium_image_url | product_type_name | |
|---|---|---|---|---|---|---|
| **61973** | B06Y1KZ2WB | Éclair | Black/Pink | https://images-na.ssl-images-amazon.com/images... | SHIRT | Éc W Pr Th Bl Bl |
| **133820** | B010RV33VE | xiaoming | Pink | https://images-na.ssl-images-amazon.com/images... | SHIRT | xia W Sl Lo Lo sh |
| **81461** | B01DDSDLNS | xiaoming | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | xia W W Lo Sl Si Br |
| **75995** | B00X5LYO9Y | xiaoming | Red Anchors | https://images-na.ssl-images-amazon.com/images... | SHIRT | xia St Ta Pa Sl Ar |
| **151570** | B00WPJG35K | xiaoming | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | xia Sl Sh Lo Ta Ki W |

**Some examples of dupliacte 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 Pri
nt Head Shirt for woman Neon Wolf t-shirt
62. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Pri
nt Head Shirt for woman Neon Wolf t-shirt
63. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Pri
nt Head Shirt for woman Neon Wolf t-shirt
64. psychedelic colorful Howling Galaxy Wolf T-shirt/Colorful Rainbow Animal Pri
nt Head Shirt for woman Neon Wolf t-shirt
```

In [0]:

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

In [0]:
```
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', 'Th
e', 'Queen', 'of', 'Diamonds', 'Women's', 'Shirt', 'X-Large']
    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', 'Diamonds', 'Women's', 'Shirt', 'Small']
        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 bo
th strings
        count  = 0

        # itertools.zip_longest(a,b): will map the corresponding words in b
oth strings, it will appened None in case of unequal strings
        # 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 those two apperals are different
        # if the number of words in which both strings differ are < 2 , we
 are considering it as those two apperals are same, hence we are ignoring t
hem
        if (length - count) > 2: # number of words in which both sensences
 differ
            # if both strings are differ by more than 2 words we include th
e 1st string index
            stage1_dedupe_asins.append(data_sorted['asin'].loc[indices[i]])

            # if the comaprision between is between num_data_points, num_da
ta_points-1 strings and they differ in more than 2 words we include both
            if j == num_data_points-1: stage1_dedupe_asins.append(data_sort
ed['asin'].loc[indices[j]])

            # start searching for similar apperals corresponds 2nd string
            i = j
```

```
            break
        else:
            j += 1
    if previous_i == i:
        break
```

In [0]:
```
data = data.loc[data['asin'].isin(stage1_dedupe_asins)]
```

**We removed the dupliactes which differ only at the end.**

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

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

In [0]:
```
data.to_pickle('pickels/17k_apperal_data')
```

### [5.2.3] Remove duplicates : Part 2

```
In the previous cell, we sorted whole data in alphabetical order of  titles.The
n, we removed titles which are adjacent and 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, X
X-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
```

In [0]:
```
data = pd.read_pickle('pickels/17k_apperal_data')
```

In [0]:
```
# 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 apperal's title
    a = data['title'].loc[i].split()
    # store the list of words of ith string in a, ex: a = ['tokidoki', 'Th
e', 'Queen', 'of', 'Diamonds', 'Women's', 'Shirt', 'X-Large']
    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', 'Diamonds', 'Women's', 'Shirt', 'X-Large']

        length = max(len(a),len(b))

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

        # itertools.zip_longest(a,b): will map the corresponding words in b
oth strings, it will appened None in case of unequal strings
        # 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 those two apperals are same, hence we are ignoring t
hem
        if (length - count) < 3:
            indices.remove(j)
```

In [0]:
```
# 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)]
```

In [0]:
```
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:  16042
```

In [0]:
```
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

In [0]:
```
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()
```

In [0]:
```
# 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 '"#$@!%^&*()_+-~?>< e
tc.
            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: {'such', 'and', 'hers', 'up', 'she', 'd', 'further', 'a
ll', 'than', 'under', 'is', 'off', 'both', 'most', 'few', 'should', 're',
'very', 'just', 'then', 'didn', 'myself', 'in', 'too', 's', 'shouldn', 'her
self', 'because', 'how', 'itself', 'what', 'shan', 'weren', 'doing', 'the
m', 'couldn', 'their', 'so', 'ain', 'haven', 'yourself', 'now', 'll', 'is
n', 'about', 'over', 'into', 'before', 'during', 'on', 'as', 'aren', 'again
st', 'above', 'down', 'they', 'below', 'me', 'again', 'for', 'why', 'been',
'yourselves', 'more', 'her', 'that', 'can', 'am', 'was', 'themselves', 'mig
htn', 'does', 'those', 'only', 'hasn', 'any', 'ma', 'are', 'nor', 'out', 'y
ou', 'ourselves', 'the', 'an', 'has', 'where', 'i', 'while', 'ours', 'its',
'your', 'had', 'were', 'being', 'no', 'or', 'needn', 've', 'y', 'a', 'eac
h', 'have', 'through', 'when', 'mustn', 'by', 'won', 'from', 'own', 'will',
'there', 't', 'him', 'these', 'doesn', 'theirs', 'my', 'did', 'of', 'who',
'until', 'wouldn', 'we', 'do', 'having', 'yours', 'other', 'wasn', 'it', 'w
ith', 'once', 'here', 'don', 'o', 'whom', 'this', 'if', 'but', 'hadn', 'ou
r', 'some', 'm', 'not', 'between', 'himself', 'same', 'at', 'be', 'he', 'af
ter', 'which', 'to', 'his'}
```

In [0]:
```
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")
```

```
3.5727220000000006 seconds
```

In [0]:
```
data.head()
```

Out[0]:

| | asin | brand | color | medium_image_url | product_type_name | |
|---|---|---|---|---|---|---|
| **4** | B004GSI2OS | FeatherLite | Onyx Black/ Stone | https://images-na.ssl-images-amazon.com/images... | SHIRT | featherli ladies lo sleeve s resistan |
| **6** | B012YX2ZPI | HX-Kingdom Fashion T-shirts | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | womens unique ' cotton special olympics wor... |
| **15** | B003BSRPB0 | FeatherLite | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | featherli ladies moisture free mes sport sh |
| **27** | B014ICEJ1Q | FNC7C | Purple | https://images-na.ssl-images-amazon.com/images... | SHIRT | superna chibis sa dean ca neck tsh |
| **46** | B01NACPBG2 | Fifth Degree | Black | https://images-na.ssl-images-amazon.com/images... | SHIRT | fifth deg womens gold foil graphic jun... |

In [0]:
```
data.to_pickle('pickels/16k_apperal_data_preprocessed')
```

# Stemming

In [0]:
```
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

In [0]:
```
data = pd.read_pickle('pickels/16k_apperal_data_preprocessed')
data.head()
```

Out[0]:

| | asin | brand | color | medium_image_url | product_type_name | |
|---|---|---|---|---|---|---|
| **4** | B004GSI2OS | FeatherLite | Onyx Black/ Stone | https://images-na.ssl-images-amazon.com/images... | SHIRT | featherli ladies lo sleeve s resistan |
| **6** | B012YX2ZPI | HX-Kingdom Fashion T-shirts | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | womens unique ' cotton special olympics wor... |
| **15** | B003BSRPB0 | FeatherLite | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | featherli ladies moisture free mes sport sh |
| **27** | B014ICEJ1Q | FNC7C | Purple | https://images-na.ssl-images-amazon.com/images... | SHIRT | superna chibis sa dean ca neck tsh |
| **46** | B01NACPBG2 | Fifth Degree | Black | https://images-na.ssl-images-amazon.com/images... | SHIRT | fifth deg womens gold foil graphic jun... |

In [0]:
```
# 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 occur
ence of the word keys(i)
        # 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) = tfi
df(keys(i))
                # if model == 'idf weighted bag of words':labels(i) = idf(k
eys(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, ploting heat map that represents the count of commonly ocurr
ed words in title2
        ax = plt.subplot(gs[0])
        # it displays a cell in white color if the word is intersection(lis
of words of title1 and list of words of title2), in black if not
        ax = sns.heatmap(np.array([values]), annot=np.array([labels]))
        ax.set_xticklabels(keys) # set that axis labels as the words of tit
le
        ax.set_title(text) # apparel title

        # 2nd, plotting image of the the apparel
        ax = plt.subplot(gs[1])
        # we don't want any grid lines for image and no labels on x-axis an
d 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 apparels's vector, it is of a dict type {word:count}
    # vec2 : recommended apparels's vector, it is of a dict type {word:coun
t}
    # url : apparels image url
    # text: title of recomonded 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 con
tribute to the distance between two title vec's
    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 heatmap
    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(lis of words of title1 and list of words
of title2): values(i)=count of that word in title2 else values(i)=0
    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 gi
ve the tfidf value of word in given document (doc_id)
            if x in  tfidf_title_vectorizer.vocabulary_:
                labels.append(tfidf_title_features[doc_id, tfidf_title_vect
orizer.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 in given document (doc_id)
            if x in  idf_title_vectorizer.vocabulary_:
                labels.append(idf_title_features[doc_id, idf_title_vectoriz
```

```
er.vocabulary_[x]])
            else:
                labels.append(0)

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


# this function gets a list of wrods 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 also gives same result
    return Counter(words) # Counter counts the occurence of each word in li
st, it returns dict type object {word1:count}



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)

    # vector1 = 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.

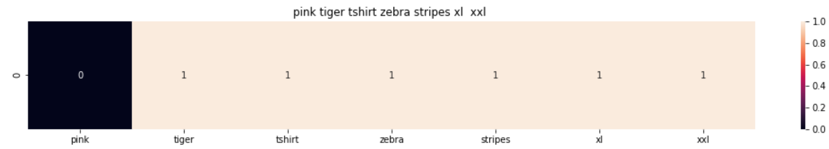
In [0]:

```
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 matr
ix.
# title_features.shape = #data_points * #words_in_corpus
# CountVectorizer().fit_transform(corpus) returns
# the a sparase 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 wor
d occured in that doc
```

Out[0]: (16042, 12609)

In [0]:
```
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 mesured 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['titl
e'].loc[df_indices[i]], data['medium_image_url'].loc[df_indices[i]], 'bag_o
f_words')
        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
```

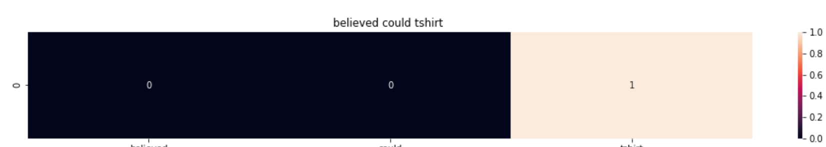

```
ASIN : B00JXQB5FQ
Brand: Si Row
Title: burnt umber tiger tshirt zebra stripes xl  xxl
Euclidean similarity with the query image : 0.0
============================================================
```



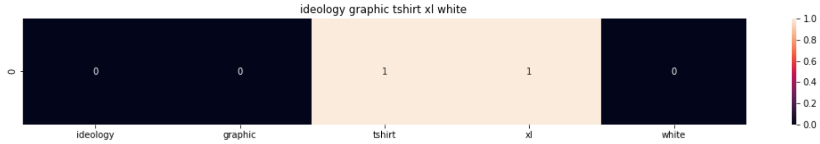

```
ASIN : B00JXQASS6
Brand: Si Row
Title: pink tiger tshirt zebra stripes xl  xxl
Euclidean similarity with the query image : 1.73205080757
============================================================
```


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

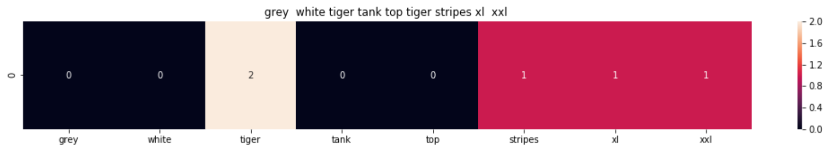

```
ASIN : B00JXQCUIC
Brand: Si Row
Title: yellow tiger tshirt tiger stripes  l
Euclidean similarity with the query image : 2.64575131106
============================================================
```




```
ASIN : B07568NZX4
Brand: Rustic Grace
Title: believed could tshirt
Euclidean similarity with the query image : 3.0
============================================================
```

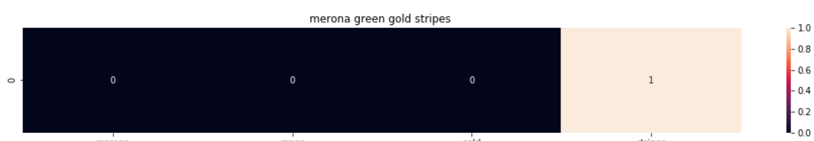



```
ASIN : B01NB0NKRO
Brand: Ideology
Title: ideology graphic tshirt xl white
Euclidean similarity with the query image : 3.0
============================================================
```




```
ASIN : B00JXQAFZ2
Brand: Si Row
Title: grey  white tiger tank top tiger stripes xl  xxl
Euclidean similarity with the query image : 3.0
============================================================
```

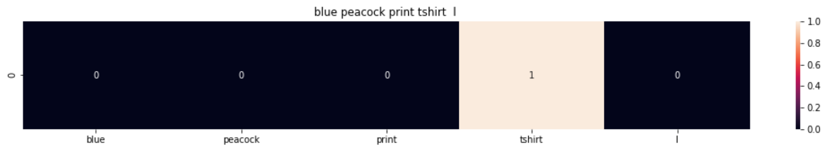

```
ASIN : B01CLS8LMW
Brand: Awake
Title: morning person tshirt  troll picture  xl
Euclidean similarity with the query image : 3.16227766017
============================================================
```




```
ASIN : B01KVZUB6G
Brand: Merona
Title: merona green gold stripes
Euclidean similarity with the query image : 3.16227766017
============================================================
```


```
ASIN : B0733R2CJK
Brand: BLVD
Title: blvd womens graphic tshirt l
Euclidean similarity with the query image : 3.16227766017
============================================================
```

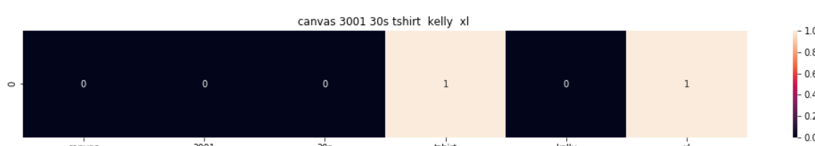

```
ASIN : B012VQLT6Y
Brand: KM T-shirt
Title: km tiger printed sleeveless vest tshirt
Euclidean similarity with the query image : 3.16227766017
============================================================
```



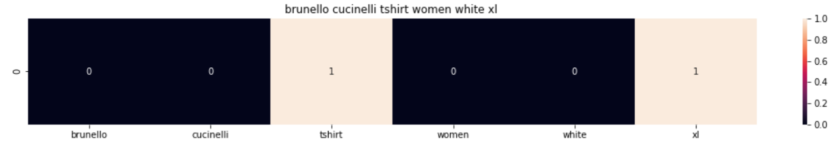

```
ASIN : B00JXQC8L6
Brand: Si Row
Title: blue peacock print tshirt  l
Euclidean similarity with the query image : 3.16227766017
============================================================
```


```
ASIN : B06XC3CZF6
Brand: Fjallraven
Title: fjallraven  womens ovik tshirt plum xxl
Euclidean similarity with the query image : 3.16227766017
============================================================
```


```
ASIN : B005IT8OBA
Brand: Hetalia
Title: hetalia us girl tshirt
Euclidean similarity with the query image : 3.16227766017
============================================================
```




```
ASIN : B0088PN0LA
Brand: Red House
Title: canvas 3001 30s tshirt  kelly  xl
Euclidean similarity with the query image : 3.16227766017
============================================================
```




```
ASIN : B06X99V6WC
Brand: Brunello Cucinelli
Title: brunello cucinelli tshirt women white xl
Euclidean similarity with the query image : 3.16227766017
============================================================
```


```
ASIN : B06Y1JPW1Q
Brand: Xhilaration
Title: xhilaration womens lace tshirt salmon xxl
Euclidean similarity with the query image : 3.16227766017
============================================================
```


```
ASIN : B06X6GX6WG
Brand: Animal
Title: animal oceania tshirt  yellow
Euclidean similarity with the query image : 3.16227766017
============================================================
```


```
ASIN : B017X8PW9U
Brand: Diesel
Title: diesel tserraf tshirt black
Euclidean similarity with the query image : 3.16227766017
============================================================
```

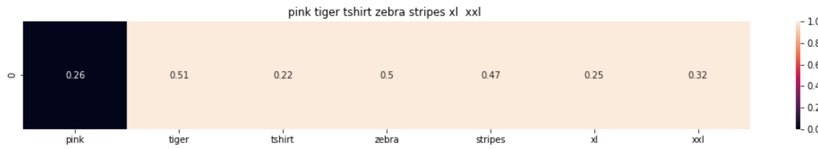

```
ASIN : B00IAA4JIQ
Brand: I Love Lucy
Title: juniors love lucywaaaaahhhh tshirt size xl
Euclidean similarity with the query image : 3.16227766017
============================================================
```

# [8.5] TF-IDF based product similarity

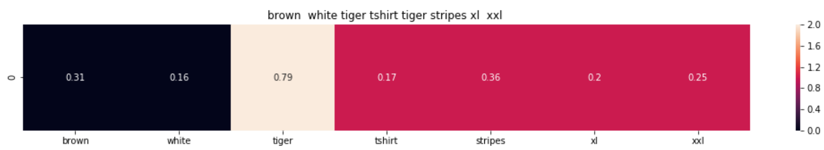
In [0]:
```
tfidf_title_vectorizer = TfidfVectorizer(min_df = 0)
tfidf_title_features = tfidf_title_vectorizer.fit_transform(data['title'])
# tfidf_title_features.shape = #data_points * #words_in_corpus
# CountVectorizer().fit_transform(courpus) returns the a sparase matrix of
 dimensions #data_points * #words_in_corpus
# tfidf_title_features[doc_id, index_of_word_in_corpus] = tfidf values of t
he word in given doc
```

In [0]:
```
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> / (||X||*||Y||)
    # http://scikit-learn.org/stable/modules/metrics.html#cosine-similarity
    pairwise_dist = pairwise_distances(tfidf_title_features,tfidf_title_fea
tures[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['titl
e'].loc[df_indices[i]], data['medium_image_url'].loc[df_indices[i]], 'tfid
f')
        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 labe
l word, the color represents the intersection with inputs title
```

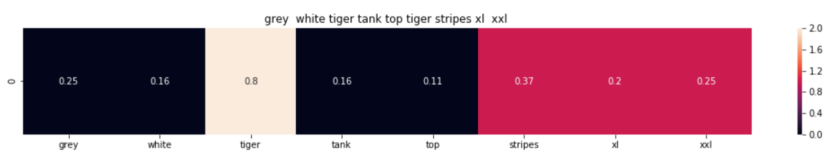
```
ASIN : B00JXQB5FQ
BRAND : Si Row
Euclidean distance from the given image : 0.0
===========================================================================
==================================================
```



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

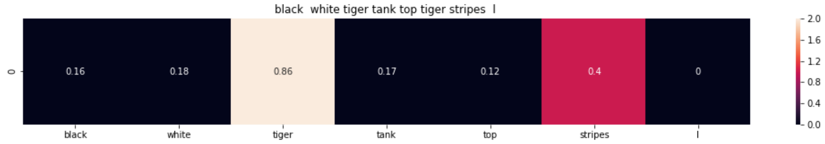


```
ASIN : B00JXQCWTO
BRAND : Si Row
Euclidean distance from the given image : 0.935764394377
===========================================================================
==================================================
```

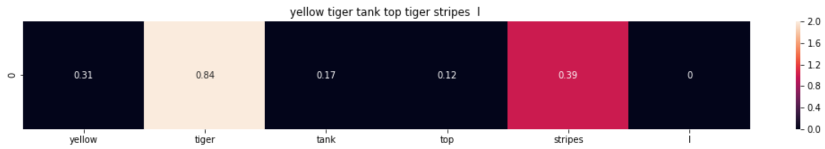


```
ASIN : B00JXQAFZ2
BRAND : Si Row
Euclidean distance from the given image : 0.95861535242
===========================================================================
==================================================
```

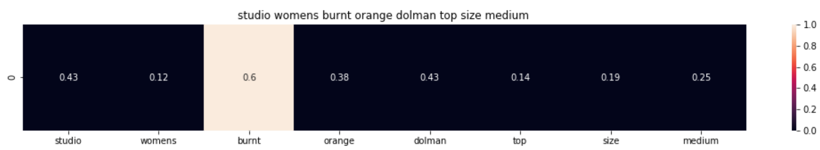
```
ASIN : B00JXQCUIC
BRAND : Si Row
Euclidean distance from the given image : 1.00007496145
===========================================================================
==================================================
```



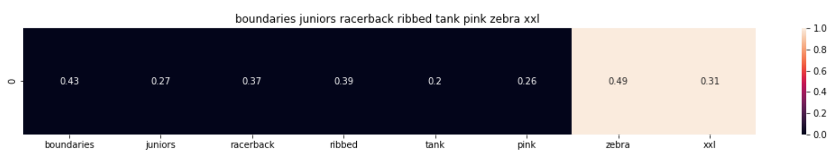

ASIN : B00JXQAO94
BRAND : Si Row
Eucliden distance from the given image : 1.02321555246
===========================================================================
==================================================




ASIN : B00JXQAUWA
BRAND : Si Row
Eucliden distance from the given image : 1.0319918463
===========================================================================
==================================================

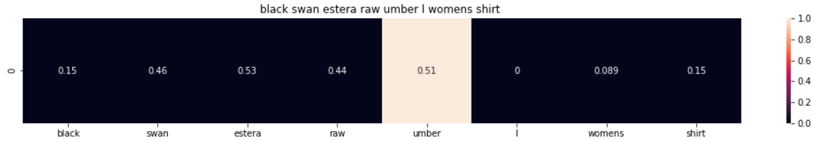



ASIN : B06XSCVFT5
BRAND : Studio M
Eucliden distance from the given image : 1.21068436704
===========================================================================
==================================================




ASIN : B06Y2GTYPM
BRAND : No Boundaries
Eucliden distance from the given image : 1.21216838107
===========================================================================
==================================================

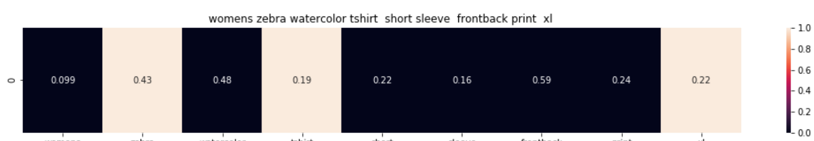

ASIN : B012VQLT6Y
BRAND : KM T-shirt
Eucliden distance from the given image : 1.21979064028
===========================================================================
==================================================




```
ASIN : B06Y1VN8WQ
BRAND : Black Swan
Eucliden distance from the given image : 1.220684966
===========================================================================
==================================================
```

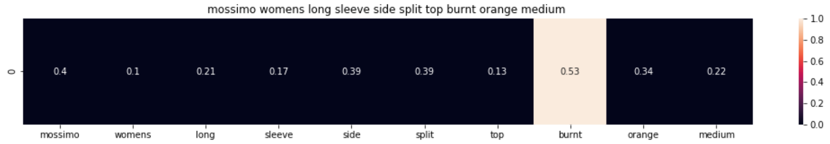

```
ASIN : B00Z6HEXWI
BRAND : Black Temptation
Eucliden distance from the given image : 1.22128139212
===========================================================================
==================================================
```


```
ASIN : B074TR12BH
BRAND : Ultra Flirt
Eucliden distance from the given image : 1.23133640946
===========================================================================
==================================================
```

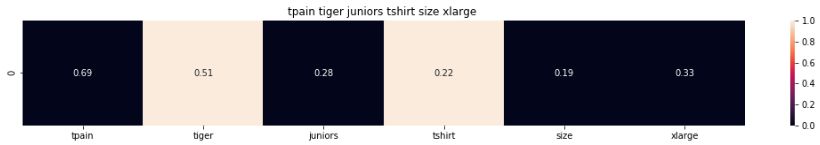



```
ASIN : B072R2JXKW
BRAND : WHAT ON EARTH
Eucliden distance from the given image : 1.23184519726
===========================================================================
==================================================
```

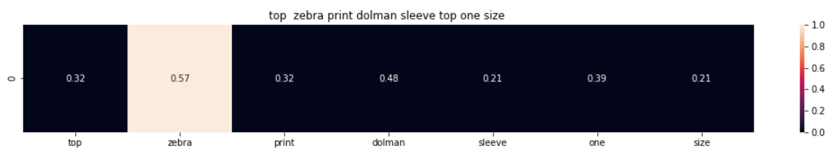

```
ASIN : B074T8ZYGX
BRAND : MKP Crop Top
Eucliden distance from the given image : 1.23406074574
===========================================================================
==================================================
```



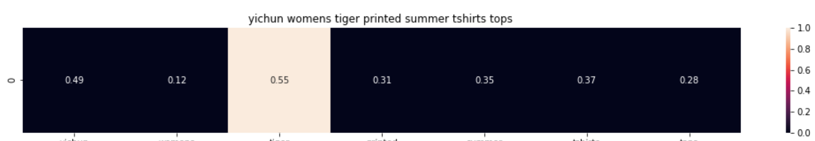

```
ASIN : B071ZDF6T2
BRAND : Mossimo
Eucliden distance from the given image : 1.23527855777
============================================================================
==================================================
```



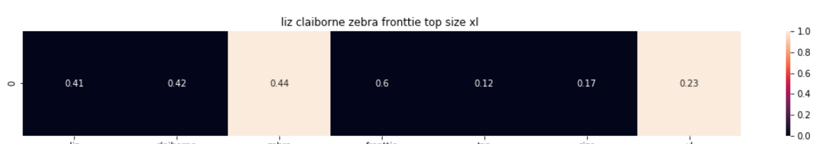

```
ASIN : B01K0H02OG
BRAND : Tultex
Eucliden distance from the given image : 1.23645729881
============================================================================
==================================================
```




```
ASIN : B00H8A6ZLI
BRAND : Vivian's Fashions
Eucliden distance from the given image : 1.24996155053
============================================================================
==================================================
```




```
ASIN : B010NN9RXO
BRAND : YICHUN
Eucliden distance from the given image : 1.25354614209
============================================================================
==================================================
```




```
ASIN : B06XBY5QXL
BRAND : Liz Claiborne
Eucliden distance from the given image : 1.25388329384
============================================================================
==================================================
```

# [8.5] IDF based product similarity

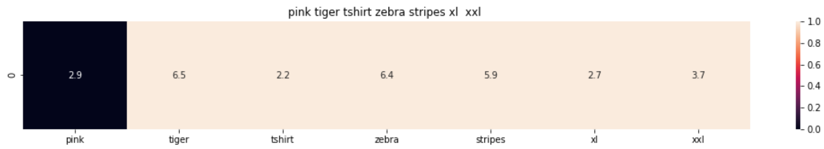
In [0]:
```
idf_title_vectorizer = CountVectorizer()
idf_title_features = idf_title_vectorizer.fit_transform(data['title'])

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

In [0]:
```
def n_containing(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)))
```

In [0]:
```
# 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 w
ith the idf values of the word
    # idf_title_features[:, idf_title_vectorizer.vocabulary_[i]].nonzero()
[0] will return all documents in which the word i present
    for j in idf_title_features[:, idf_title_vectorizer.vocabulary_[i]].non
zero()[0]:

        # we replace the count values of word i in document j with  idf_val
ue of word i
        # idf_title_features[doc_id, index_of_word_in_courpus] = idf value
 of word
        idf_title_features[j,idf_title_vectorizer.vocabulary_[i]] = idf_val
```

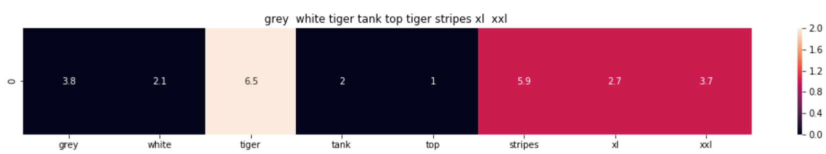
In [0]:
```
def idf_model(doc_id, num_results):
    # doc_id: apparel's id in given corpus

    # 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_feature
s[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['titl
e'].loc[df_indices[i]], data['medium_image_url'].loc[df_indices[i]], 'idf')
        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 represents the intersection with inputs title
```


```
ASIN : B00JXQB5FQ
Brand : Si Row
euclidean distance from the given image : 0.0
===========================================================================
==================================================
```

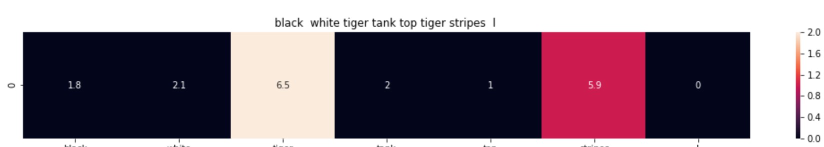



```
ASIN : B00JXQASS6
Brand : Si Row
euclidean distance from the given image : 12.2050713112
===========================================================================
==================================================
```


```
ASIN : B00JXQCWTO
Brand : Si Row
euclidean distance from the given image : 14.4683626856
===========================================================================
==================================================
```

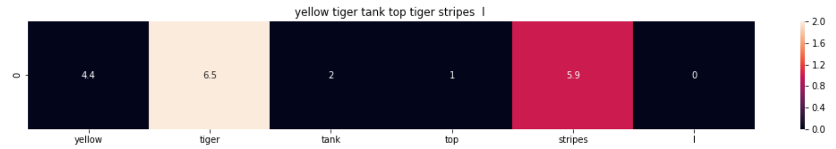



```
ASIN : B00JXQAFZ2
Brand : Si Row
euclidean distance from the given image : 14.4868329248
===========================================================================
==================================================
```

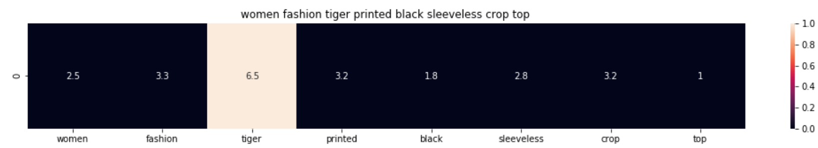



```
ASIN : B00JXQAO94
Brand : Si Row
euclidean distance from the given image : 14.8333929667
===========================================================================
==================================================
```


ASIN : B00JXQCUIC
Brand : Si Row
euclidean distance from the given image : 14.8987445167
=============================================================================
==================================================

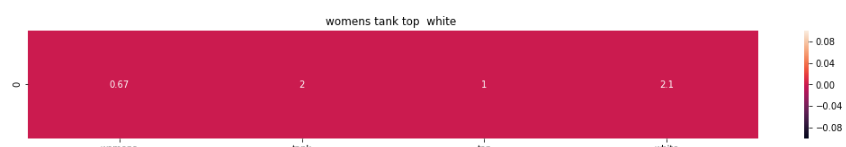



ASIN : B00JXQAUWA
Brand : Si Row
euclidean distance from the given image : 15.2244582873
=============================================================================
==================================================

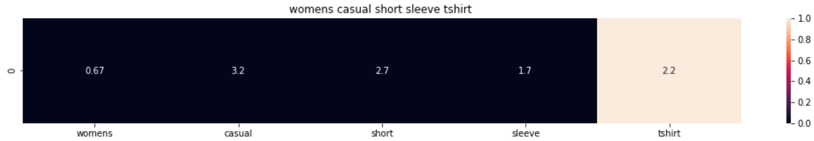



ASIN : B074T8ZYGX
Brand : MKP Crop Top
euclidean distance from the given image : 17.0808129556
=============================================================================
==================================================

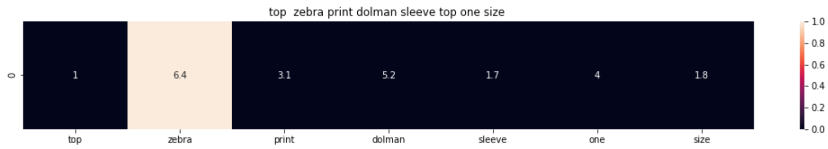

ASIN : B00KF2N5PU
Brand : Vietsbay
euclidean distance from the given image : 17.0901681256
=============================================================================
==================================================




ASIN : B00JPOZ9GM
Brand : Sofra
euclidean distance from the given image : 17.1532153376
=============================================================================
==================================================

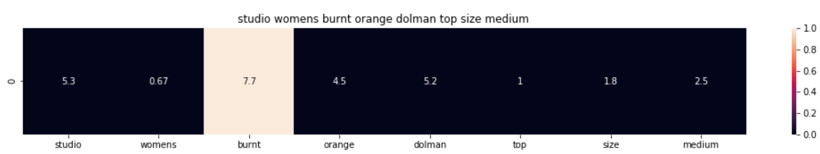



ASIN : B074T9KG9Q
Brand : Rain
euclidean distance from the given image : 17.3367152387
============================================================================================================================

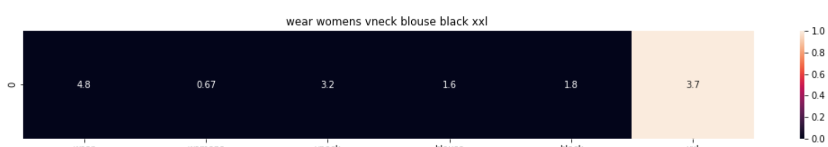



ASIN : B00H8A6ZLI
Brand : Vivian's Fashions
euclidean distance from the given image : 17.410075941
============================================================================================================================

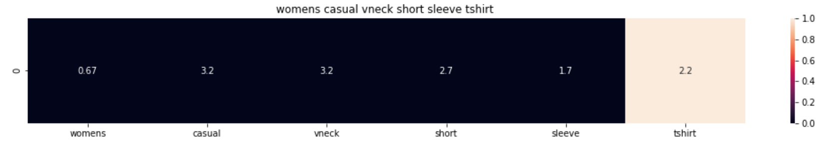

ASIN : B074G5G5RK
Brand : ERMANNO SCERVINO
euclidean distance from the given image : 17.5399213355
============================================================================================================================



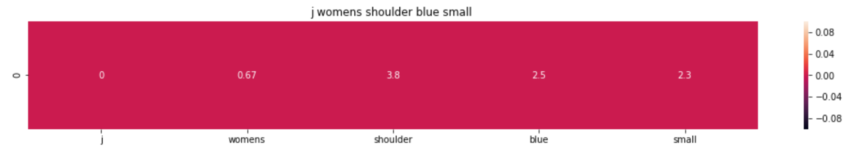

ASIN : B06XSCVFT5
Brand : Studio M
euclidean distance from the given image : 17.6127585437
============================================================================================================================




ASIN : B06Y6FH453
Brand : Who What Wear
euclidean distance from the given image : 17.6237452825
============================================================================================================================




```
ASIN : B074V45DCX
Brand : Rain
euclidean distance from the given image : 17.6343424968
============================================================================
==================================================
```




```
ASIN : B07583CQFT
Brand : Very J
euclidean distance from the given image : 17.6375371274
============================================================================
==================================================
```


```
ASIN : B073GJGVBN
Brand : Ivan Levi
euclidean distance from the given image : 17.7230738913
============================================================================
==================================================
```


```
ASIN : B012VQLT6Y
Brand : KM T-shirt
euclidean distance from the given image : 17.7625885612
============================================================================
==================================================
```


```
ASIN : B00ZZMYBRG
Brand : HP-LEISURE
euclidean distance from the given image : 17.7795368647
============================================================================
==================================================
```

# [9] Text Semantics based product similarity

In [0]:

```
# credits: https://www.kaggle.com/c/word2vec-nlp-tutorial#part-2-word-vecto
rs
# 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.Word2Vec(sen_corpus, workers=num_workers, \
            size=num_features, min_count = min_word_count, \
            window = context)

'''
```

In [0]:

```
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 wich contains a dict ,
# and it contains all our courpus 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/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit
# it's 1.9GB in size.

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

#if you do NOT have RAM >= 12GB, use the code below.
with open('word2vec_model', 'rb') as handle:
    model = pickle.load(handle)
```

In [0]:
```
# 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 representat
ion 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.vocabula
ry_:
                vec.append(idf_title_features[doc_id, idf_title_vectorizer.
vocabulary_[i]] * model[i])
            elif m_name == 'avg':
                vec.append(model[i])
        else:
            # if the word in our courpus is not there in the google word2ve
c corpus, we are just ignoring it
            vec.append(np.zeros(shape=(300,)))
    # we will return a numpy array of shape (#number of words in title * 30
0 ) 300 = len(w2v_model[word])
    # each row represents the word2vec representation of each word (weighte
d/avg) in given sentance
    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 corresponds to each word in give title
    # vec2 = np.array(#number_of_words_title2 * 300), each row is a vector
 of length 300 corresponds to each word in give title

    final_dist = []
    # for each vector in vec1 we caluclate the distance(euclidean) to all v
ectors in vec2
    for i in vec1:
        dist = []
        for j in vec2:
            # np.linalg.norm(i-j) will result the euclidean distance betwee
n 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 i
n 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
```

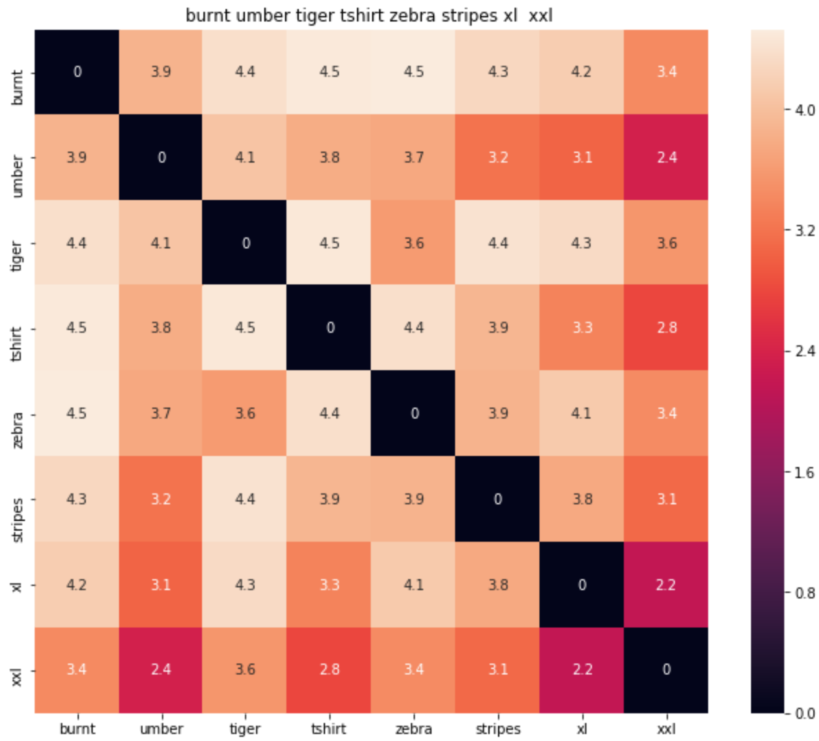
```
    # doc_id2: document id of recommended apparel
    # model: it can have two values, 1. avg 2. weighted

    #s1_vec = np.array(#number_of_words_title1 * 300), each row is a vector
(weighted/avg) of length 300 corresponds to each word in give title
    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 length 300 corresponds to each word in give title
    s2_vec = get_word_vec(sentence2, doc_id2, model)

    # s1_s2_dist = np.array(#number of words in title1 * #number of words i
n 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 d
isplays image of apparel
    gs = gridspec.GridSpec(2, 2, width_ratios=[4,1],height_ratios=[2,1])
    fig = plt.figure(figsize=(15,15))

    ax = plt.subplot(gs[0])
    # ploting 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()
```

In [0]:
```
# 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 vecto
r 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 representat
ion 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 intialize a vector of size 300 with all zeros
    # we add each word2vec(wordi) to this fetureVec
    nwords = 0

    for word in sentence.split():
        nwords += 1
        if word in vocab:
            if m_name == 'weighted' and word in  idf_title_vectorizer.vocab
ulary_:
                featureVec = np.add(featureVec, idf_title_features[doc_id,
idf_title_vectorizer.vocabulary_[word]] * model[word])
            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.

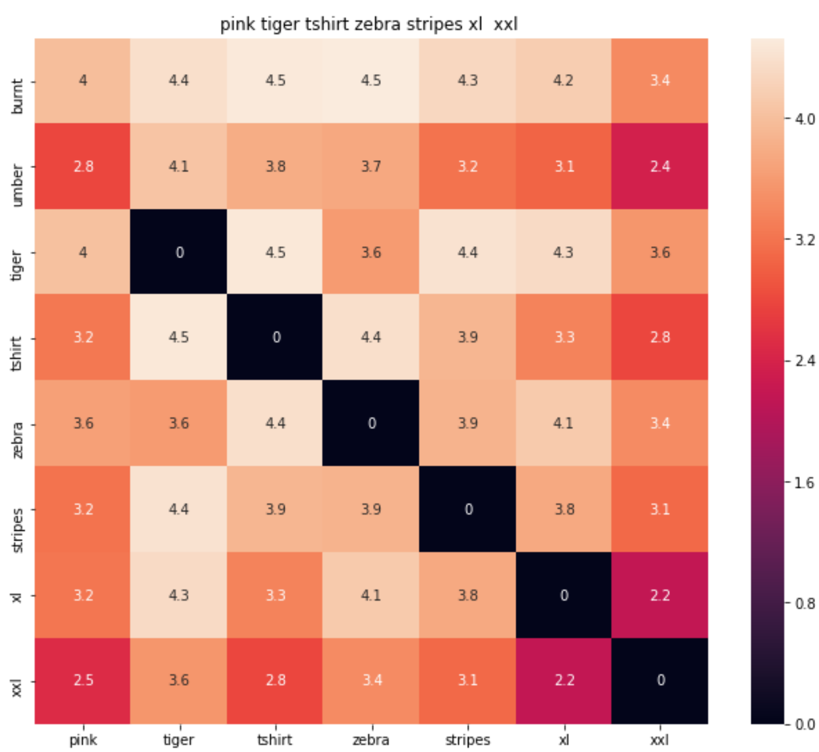
In [0]:
```
doc_id = 0
w2v_title = []
# for every title we build a avg vector representation
for i in data['title']:
    w2v_title.append(build_avg_vec(i, 300, doc_id,'avg'))
    doc_id += 1

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

In [0]:
```
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['medium_image_url'].loc[df_indices[i]], indices[0], indic
es[i], 'avg')
        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 w
ords i, j
```



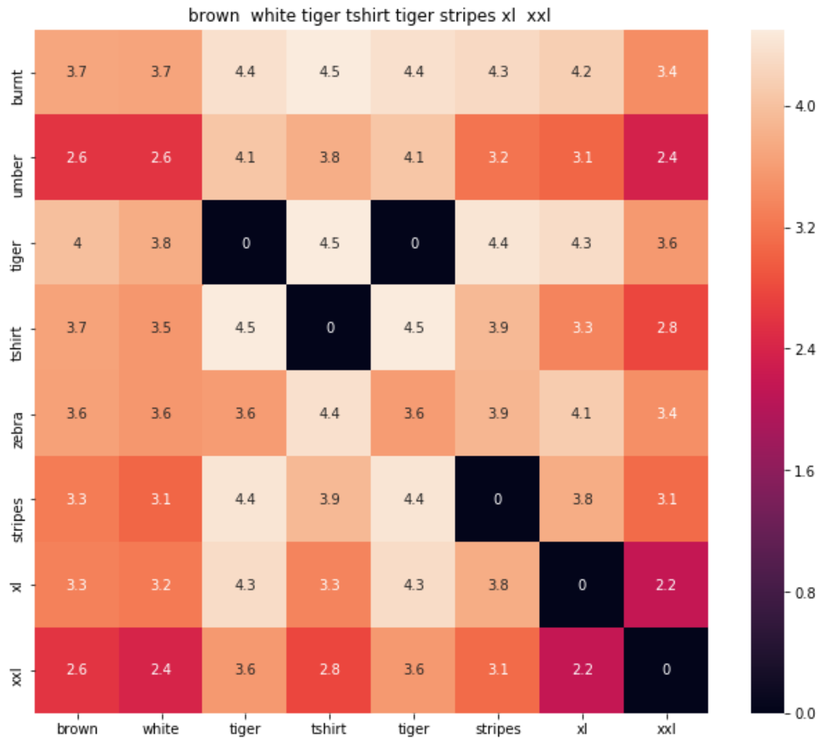

ASIN : B00JXQB5FQ
BRAND : Si Row
euclidean distance from given input image : 0.000690534

===========================================================================
==================================================

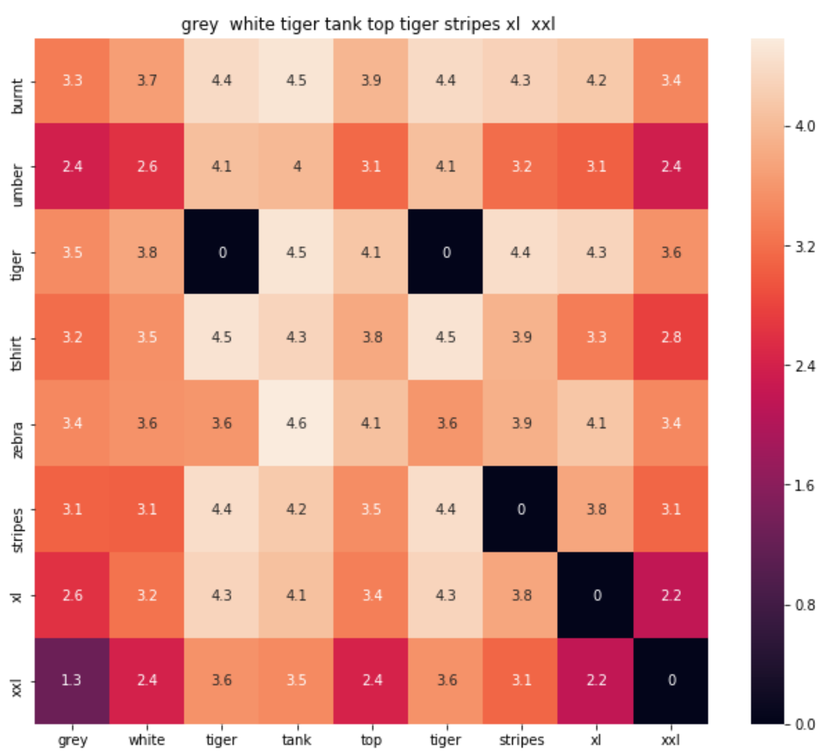



ASIN : B00JXQASS6
BRAND : Si Row
euclidean distance from given input image : 0.589193

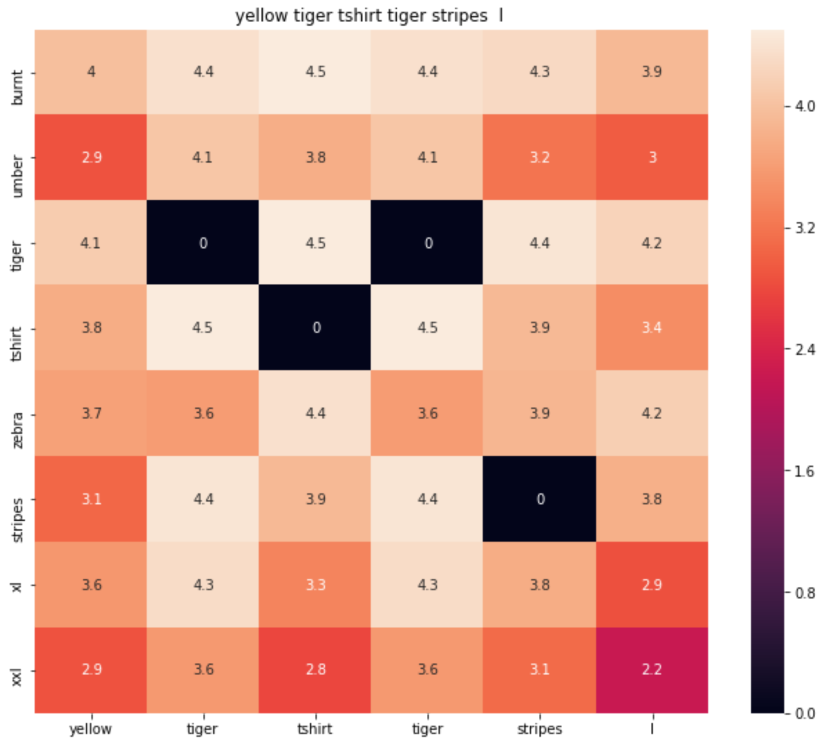
===========================================================================
==================================================



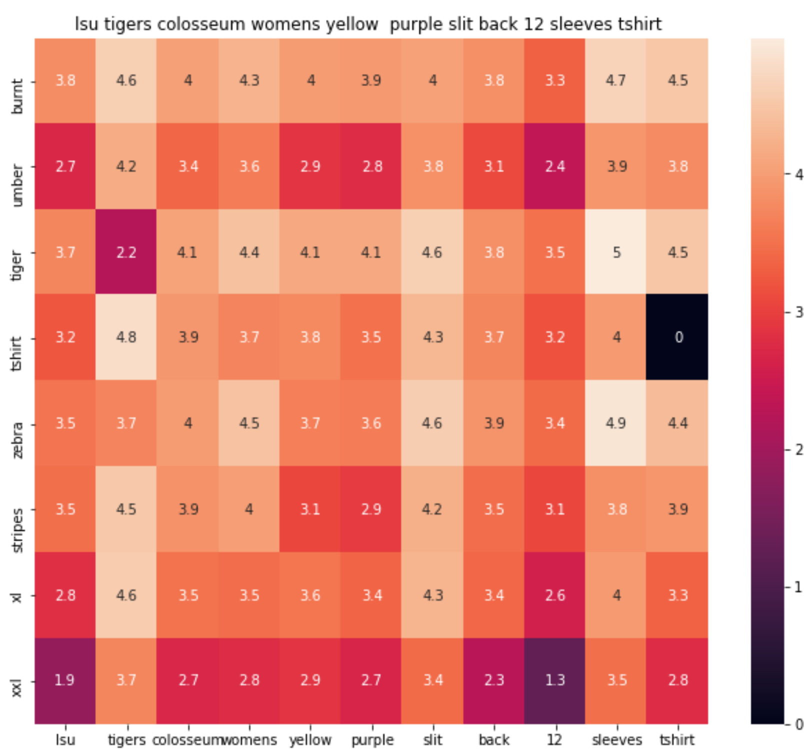

```
ASIN : B00JXQCWTO
BRAND : Si Row
euclidean distance from given input image : 0.700344
============================================================================
==================================================
```



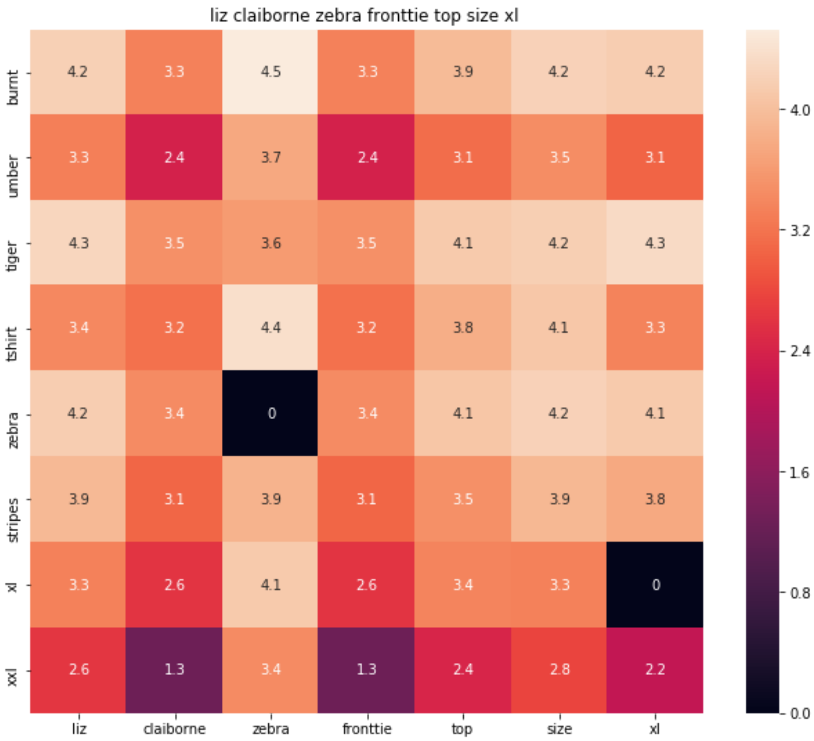

```
ASIN : B00JXQAFZ2
BRAND : Si Row
euclidean distance from given input image : 0.89284
============================================================================
==================================================
```



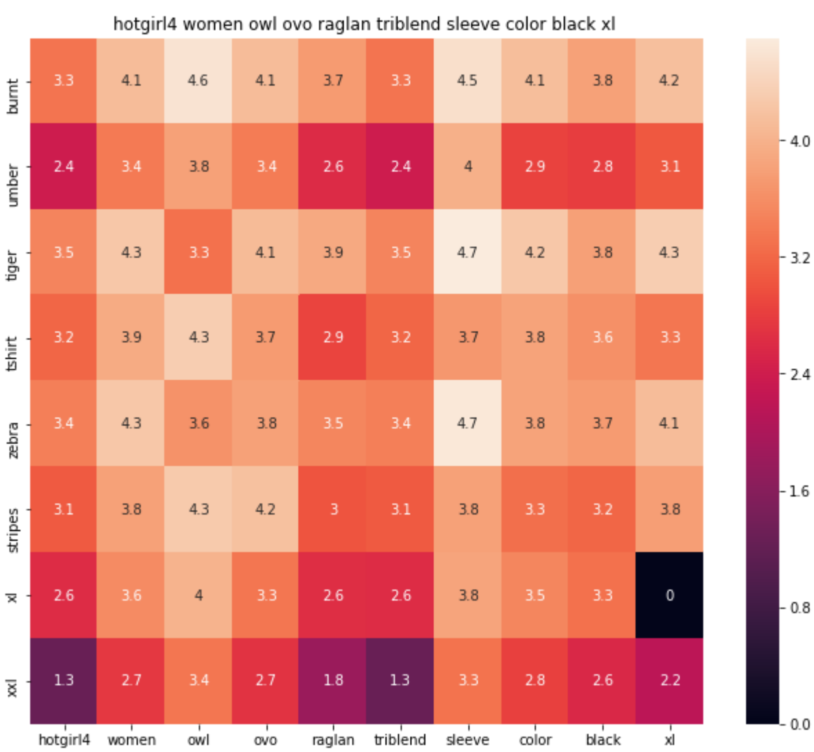

ASIN : B00JXQCUIC
BRAND : Si Row
euclidean distance from given input image : 0.956013
=============================================================================
====================================================



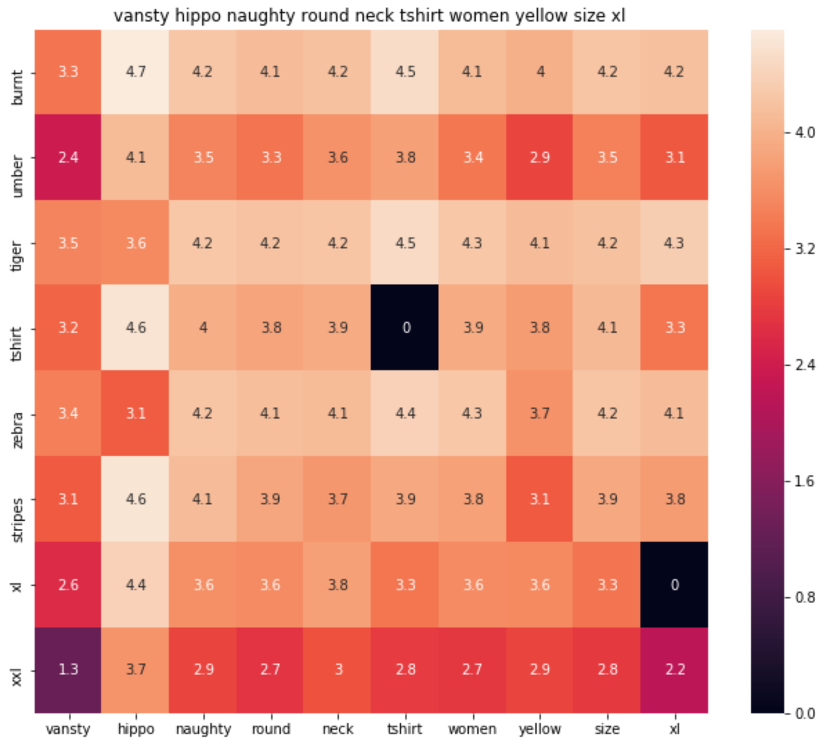

ASIN : B073R5Q8HD
BRAND : Colosseum
euclidean distance from given input image : 1.02297
=============================================================================
====================================================



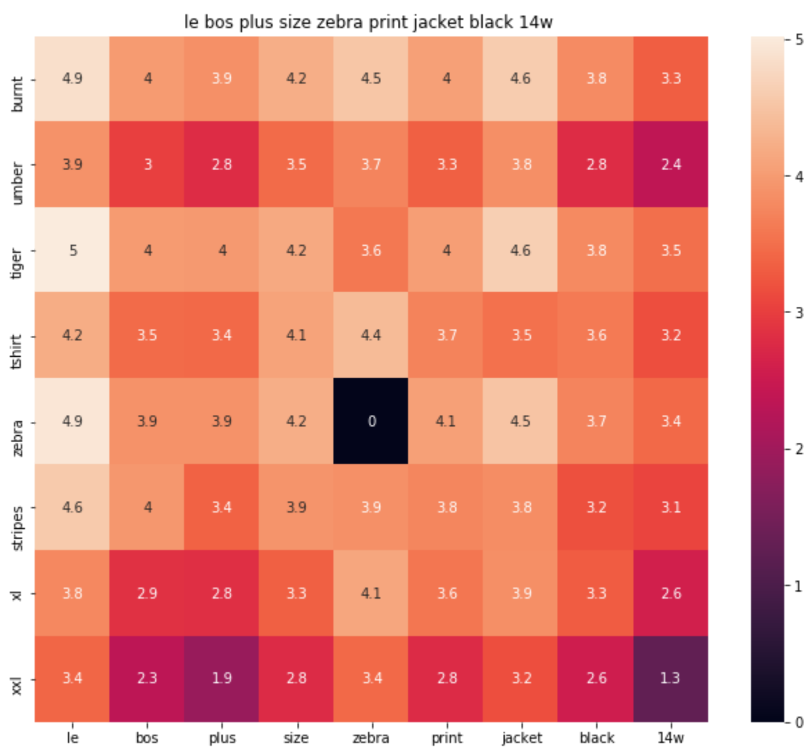

ASIN : B06XBY5QXL
BRAND : Liz Claiborne
euclidean distance from given input image : 1.06693
===========================================================================
==================================================



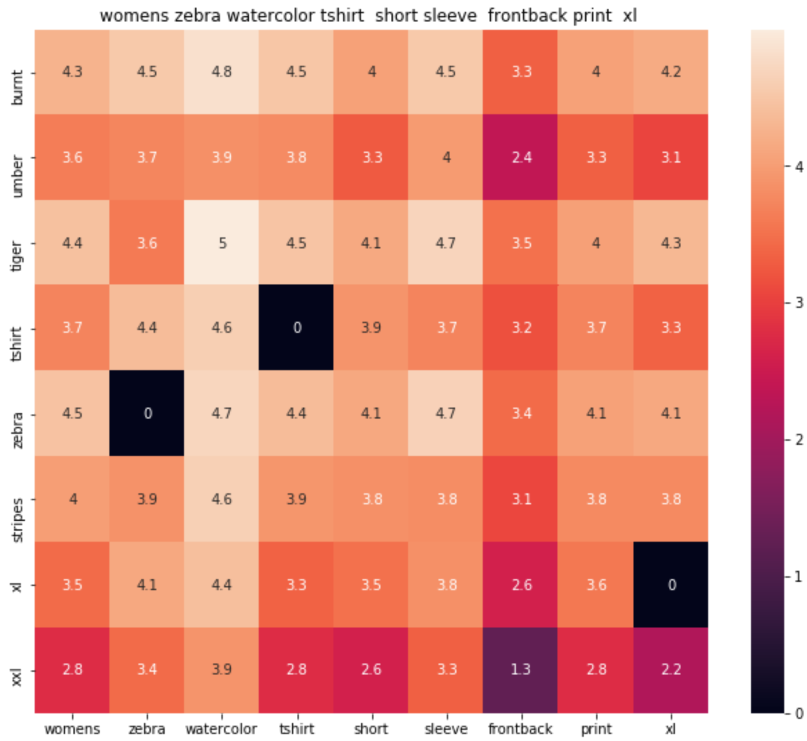

ASIN : B01L8L73M2
BRAND : Hotgirl4 Raglan Design
euclidean distance from given input image : 1.07314
===========================================================================
==================================================



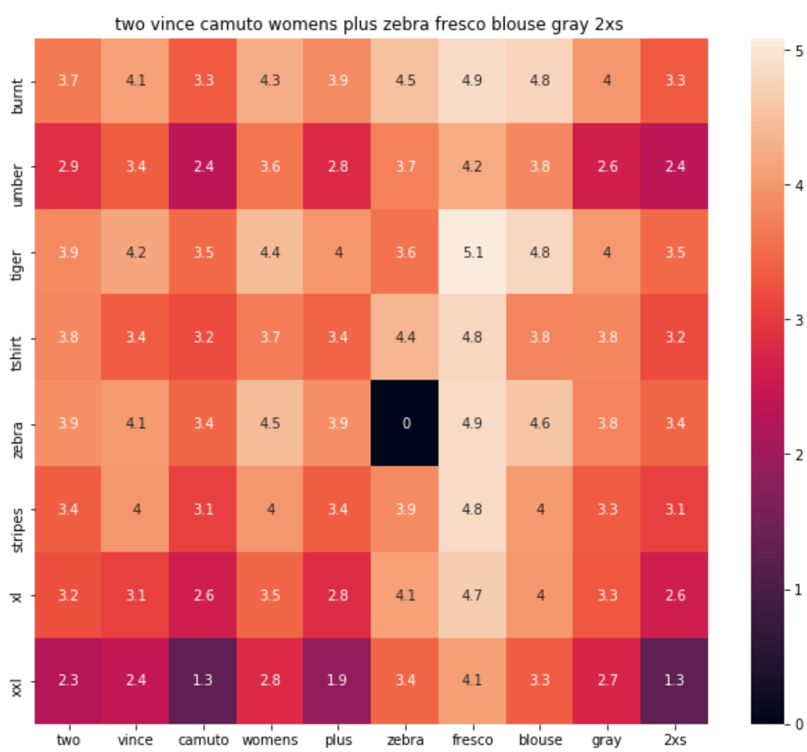

```
ASIN : B01EJS5H06
BRAND : Vansty
euclidean distance from given input image : 1.07572
===========================================================================
==================================================
```



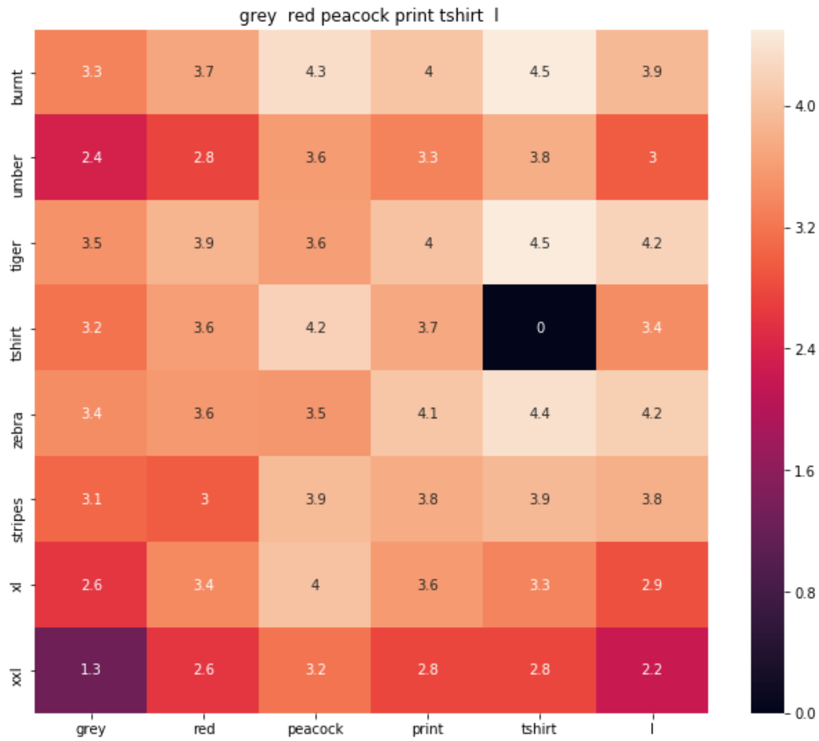

```
ASIN : B01BO1XRK8
BRAND : Le Bos
euclidean distance from given input image : 1.084
===========================================================================
==================================================
```



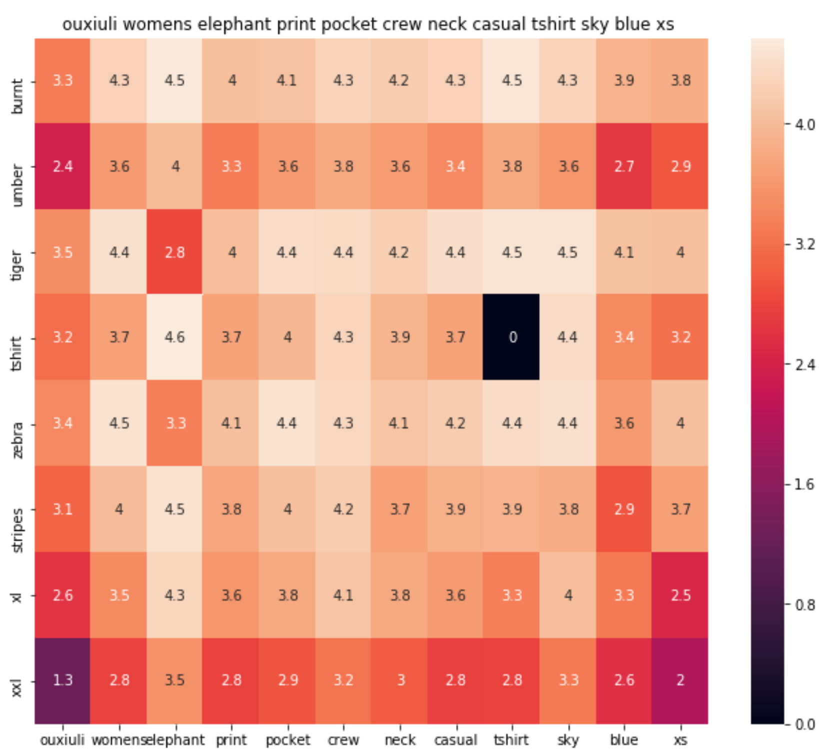

```
ASIN : B072R2JXKW
BRAND : WHAT ON EARTH
euclidean distance from given input image : 1.08422
===========================================================================
==================================================
```



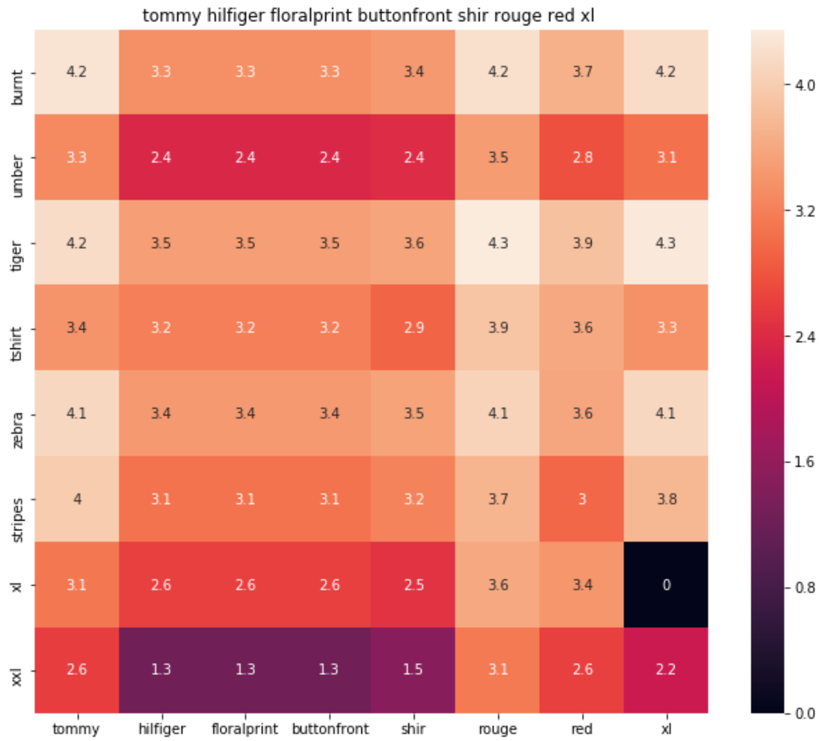

```
ASIN : B074MJRGW6
BRAND : Two by Vince Camuto
euclidean distance from given input image : 1.0895
===========================================================================
==================================================
```



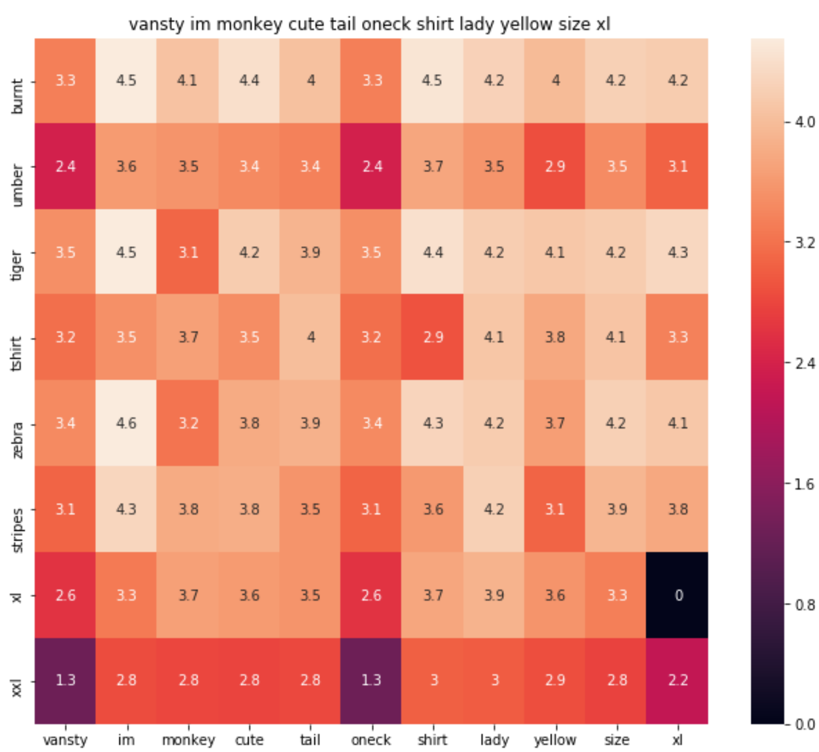

```
ASIN : B00JXQCFRS
BRAND : Si Row
euclidean distance from given input image : 1.09006
===========================================================================
==================================================
```




```
ASIN : B01I53HU6K
BRAND : ouxiuli
euclidean distance from given input image : 1.09201
===========================================================================
==================================================
```

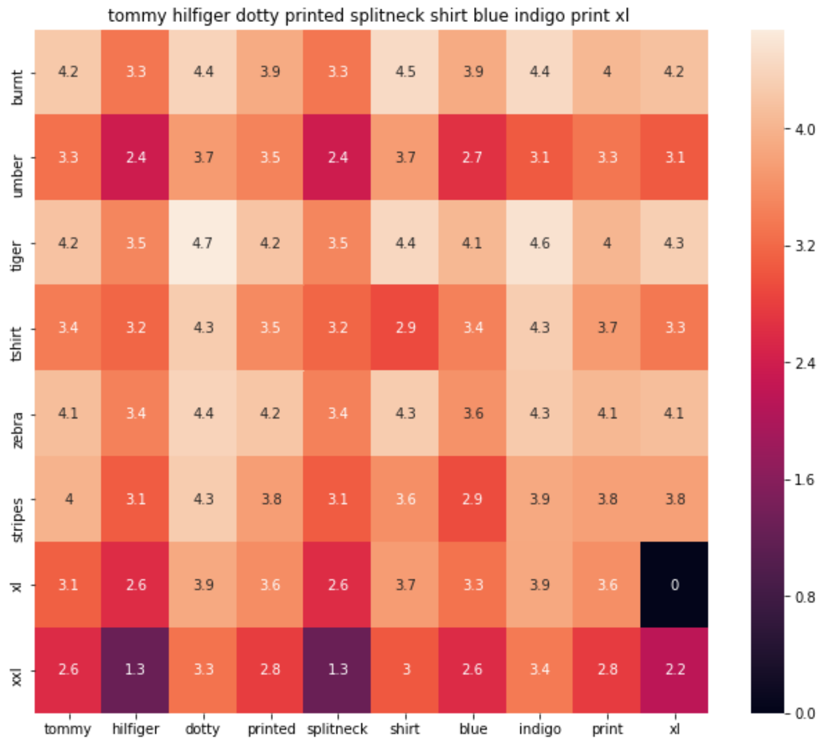



```
ASIN : B0711NGTQM
BRAND : THILFIGER RTW
euclidean distance from given input image : 1.09234
===========================================================================
==================================================
```

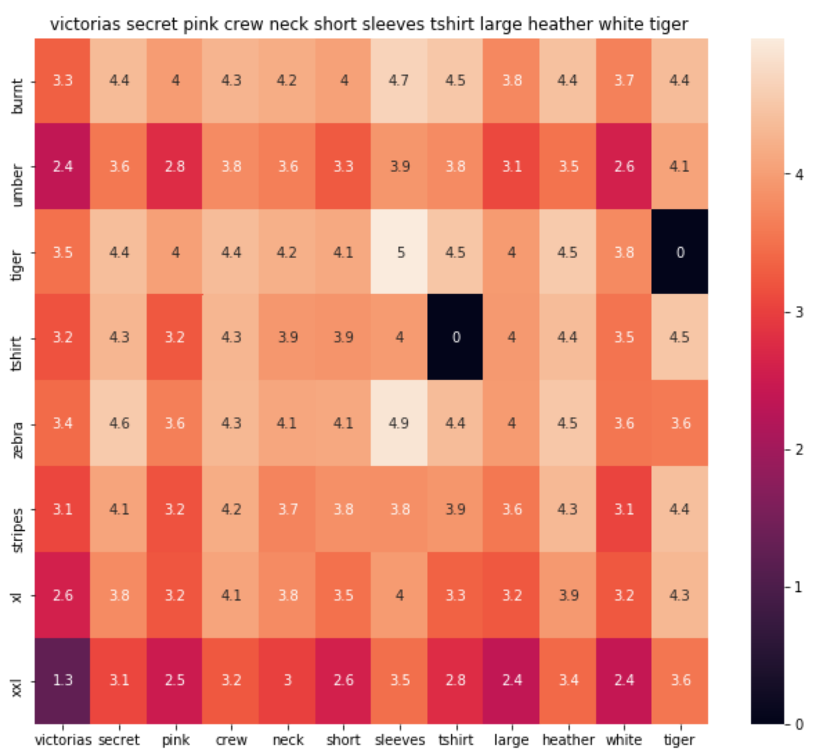




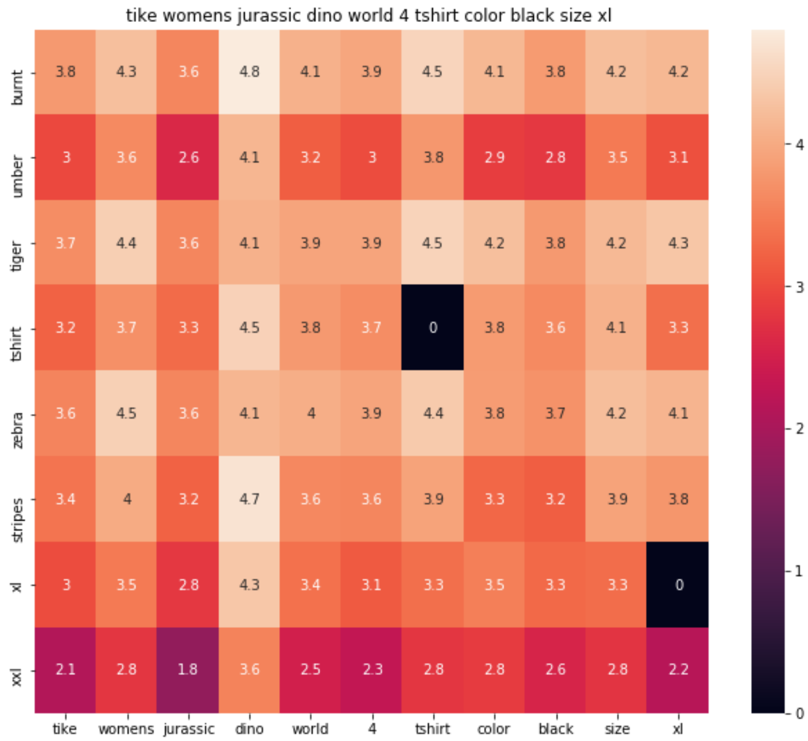
```
ASIN : B01EFSLO8Y
BRAND : Vansty
euclidean distance from given input image : 1.0934
===========================================================================
==================================================
```



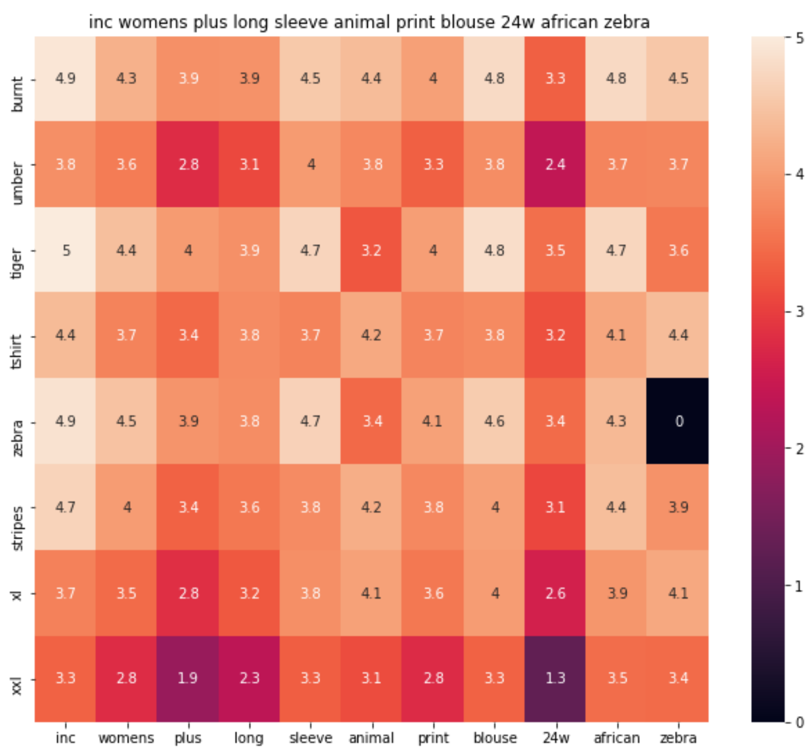

```
ASIN : B0716TVWQ4
BRAND : THILFIGER RTW
euclidean distance from given input image : 1.0942
===========================================================================
==================================================
```




```
ASIN : B0716MVPGV
BRAND : V.Secret
euclidean distance from given input image : 1.09483
===========================================================================
==================================================
```




```
ASIN : B016OPN4OI
BRAND : TIKE Fashions
euclidean distance from given input image : 1.09513
===============================================================================
====================================================
```

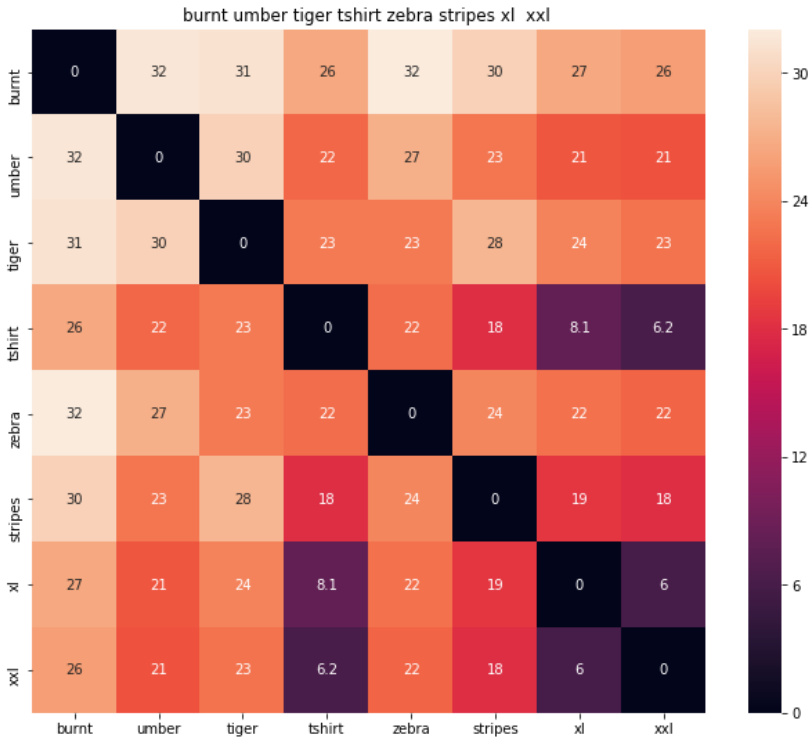



```
ASIN : B018WDJCUA
BRAND : INC - International Concepts Woman
euclidean distance from given input image : 1.09669
===============================================================================
====================================================
```

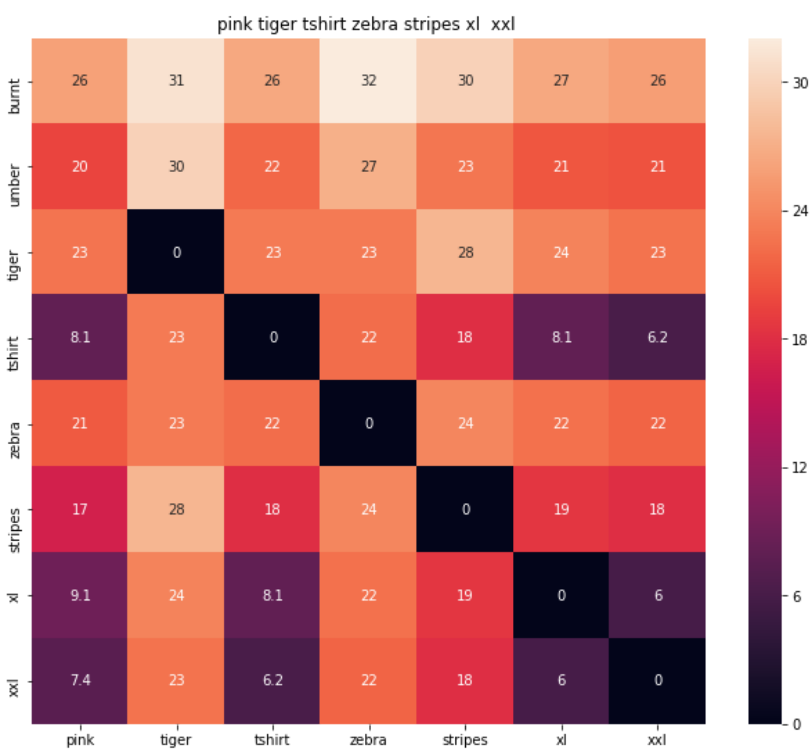
## [9.4] IDF weighted Word2Vec for product similarity

In [0]:
```
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 correspo
nds to a doc
w2v_title_weight = np.array(w2v_title_weight)
```

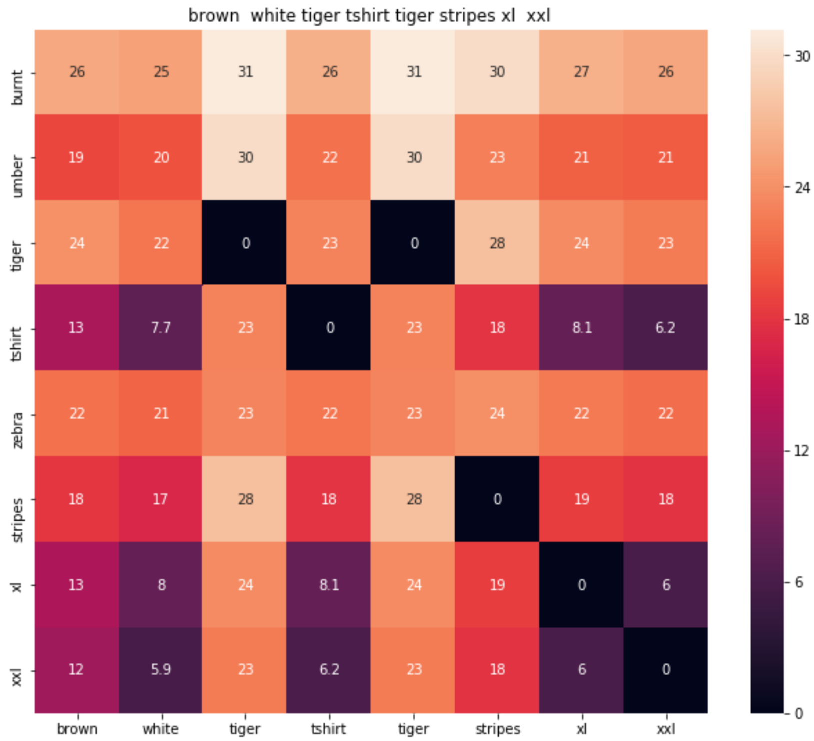
In [0]:
```
def weighted_w2v_model(doc_id, num_results):
    # doc_id: apparel's id in given corpus

    # 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[d
oc_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['medium_image_url'].loc[df_indices[i]], indices[0], indic
es[i], 'weighted')
        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 w
ords i, j
```



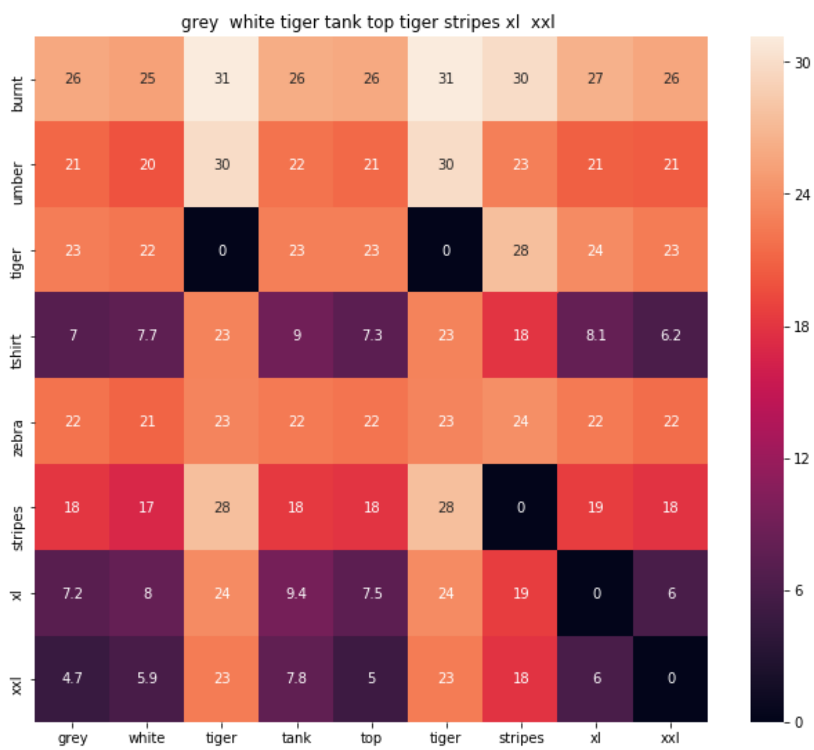

```
ASIN : B00JXQB5FQ
Brand : Si Row
euclidean distance from input : 0.00390625
=============================================================================
===================================================
```



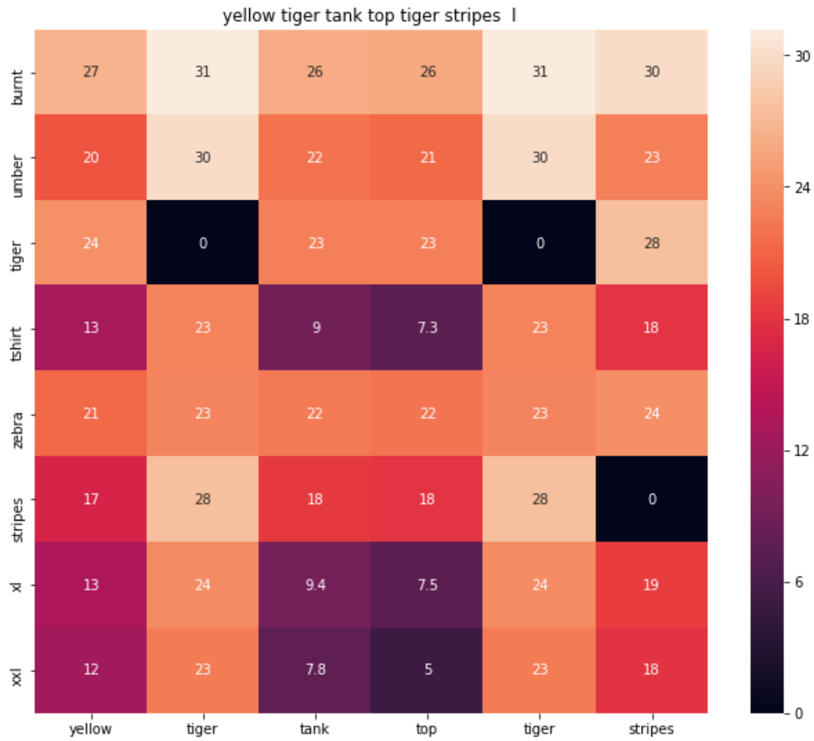

```
ASIN : B00JXQASS6
Brand : Si Row
euclidean distance from input : 4.06389
=============================================================================
===================================================
```



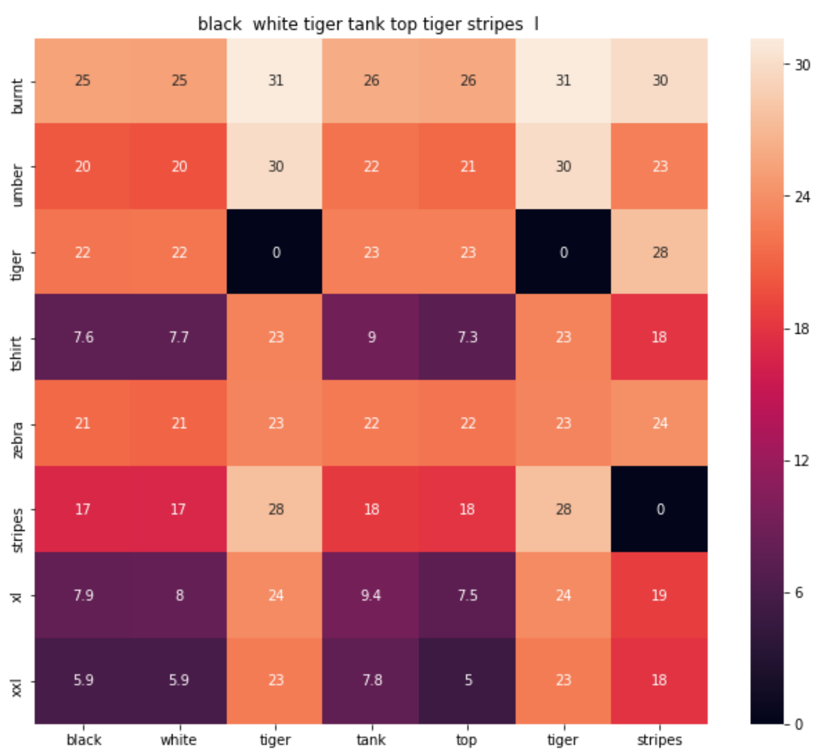

```
ASIN : B00JXQCWTO
Brand : Si Row
euclidean distance from input : 4.77094
==============================================================================
==================================================
```



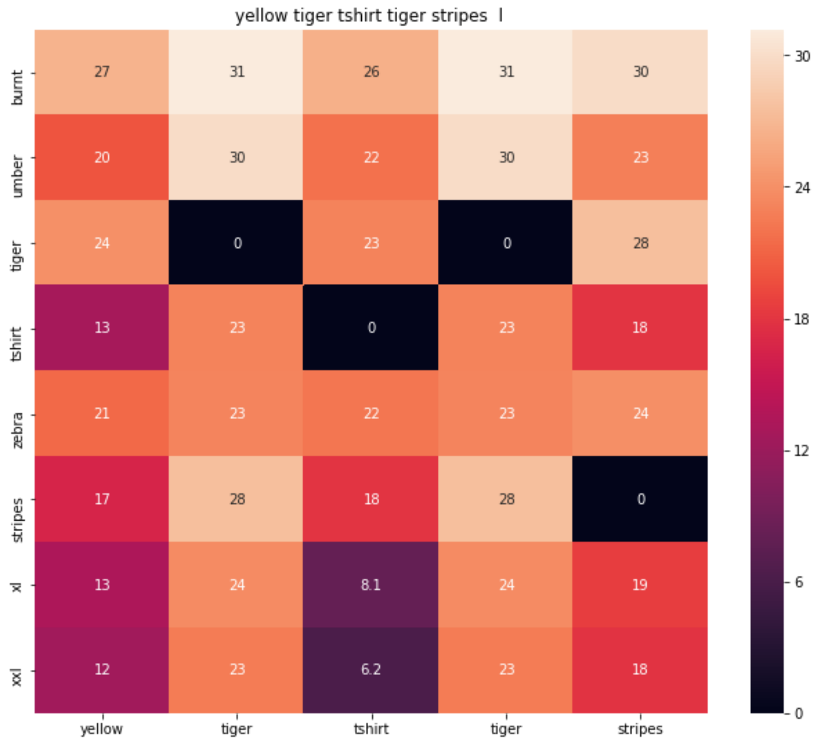

```
ASIN : B00JXQAFZ2
Brand : Si Row
euclidean distance from input : 5.36016
==============================================================================
==================================================
```



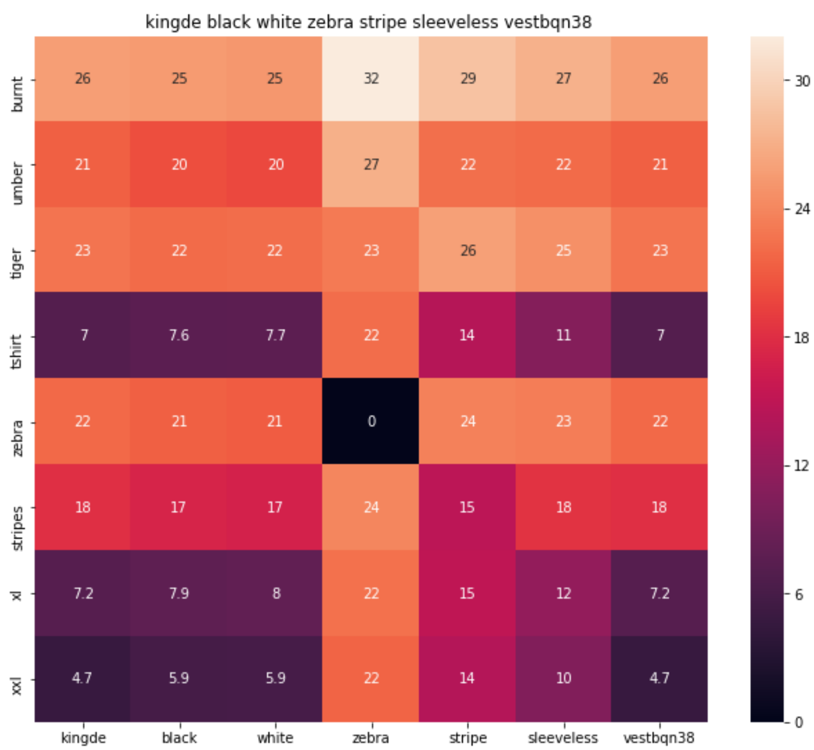

```
ASIN : B00JXQAUWA
Brand : Si Row
euclidean distance from input : 5.68952
================================================================================
==================================================
```



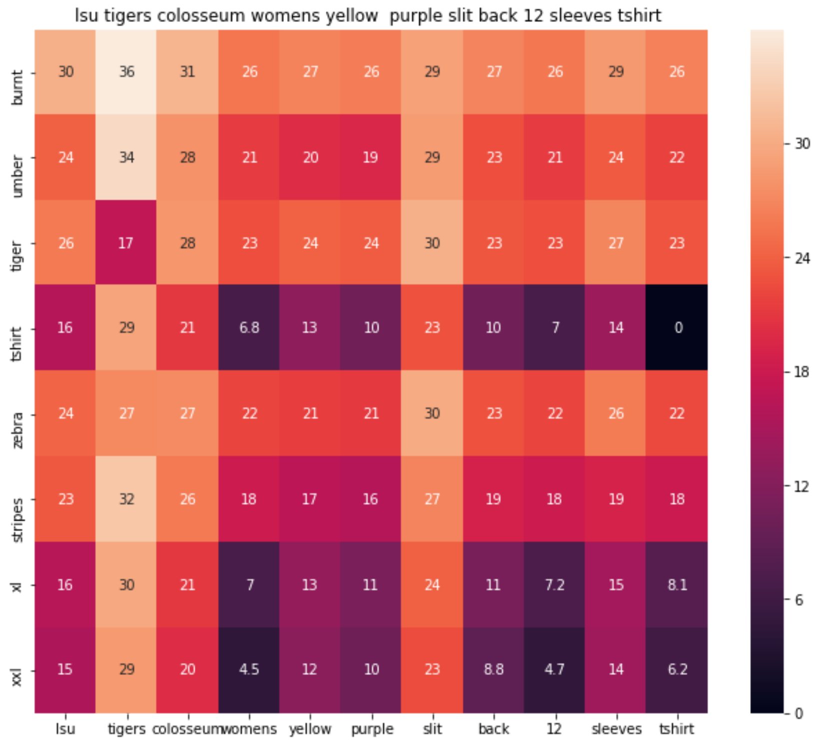

```
ASIN : B00JXQAO94
Brand : Si Row
euclidean distance from input : 5.69302
================================================================================
==================================================
```



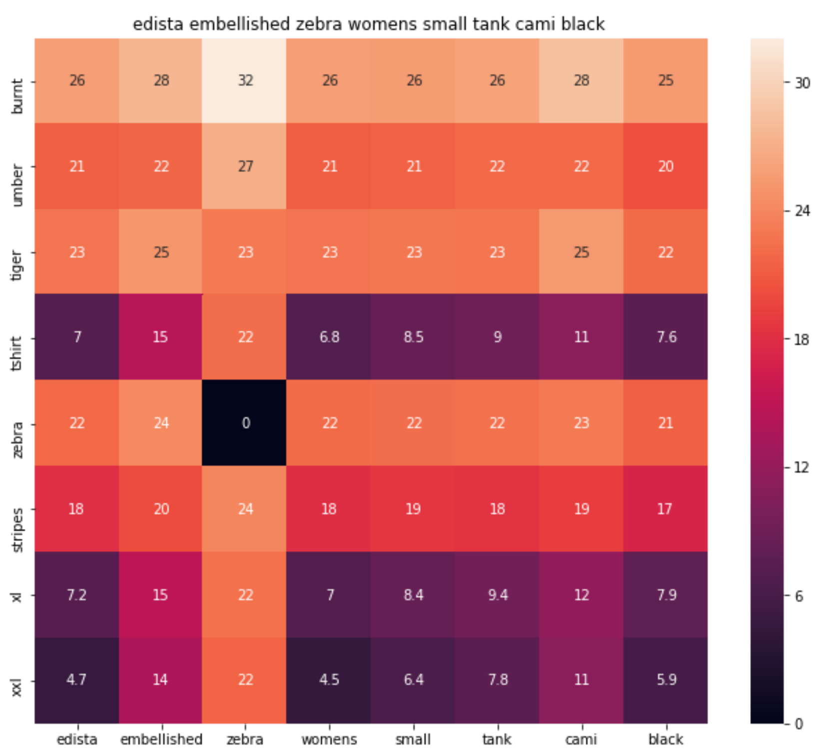

```
ASIN : B00JXQCUIC
Brand : Si Row
euclidean distance from input : 5.89344
================================================================================
=====================================================
```



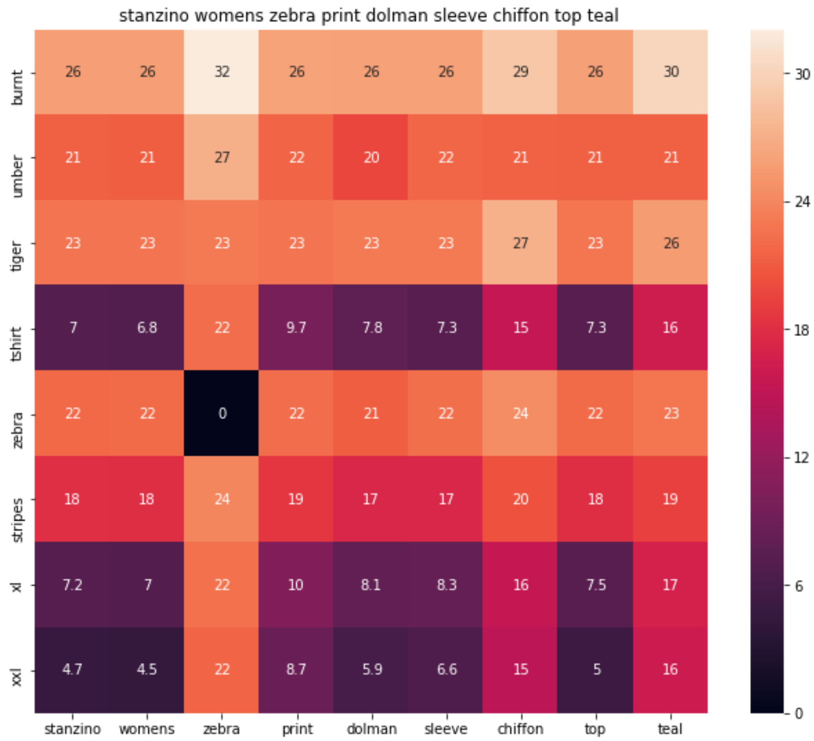

```
ASIN : B015H41F6G
Brand : KINGDE
euclidean distance from input : 6.13299
================================================================================
=====================================================
```



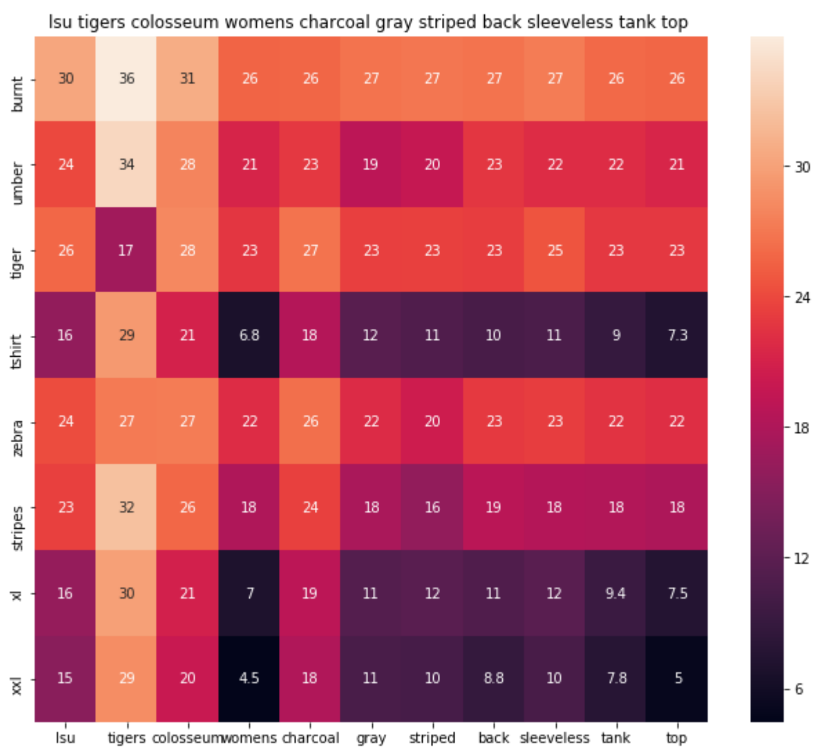

```
ASIN : B073R5Q8HD
Brand : Colosseum
euclidean distance from input : 6.25671
============================================================================
==================================================
```



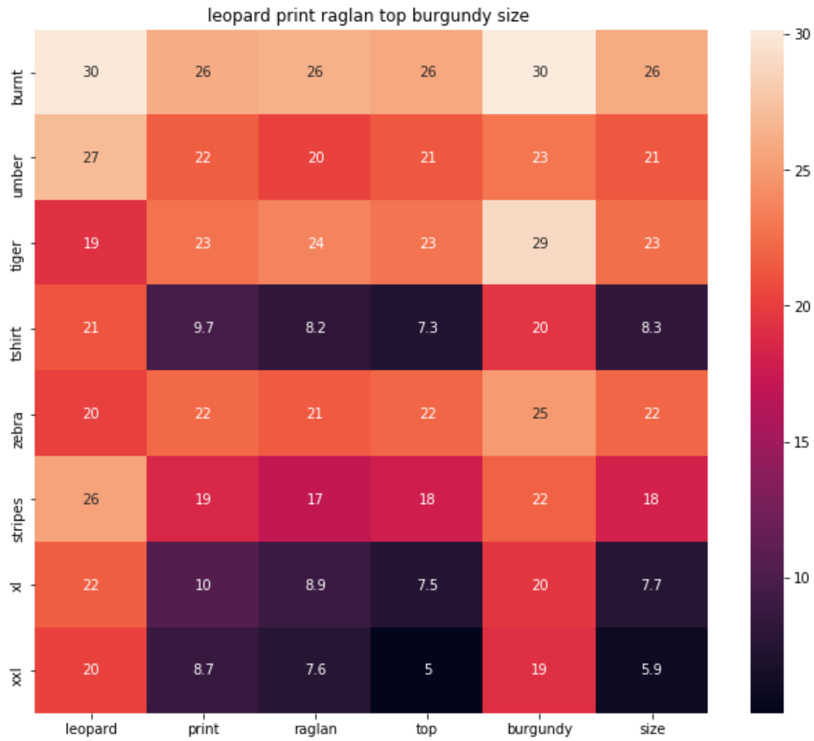

```
ASIN : B074P8MD22
Brand : Edista
euclidean distance from input : 6.3922
============================================================================
==================================================
```



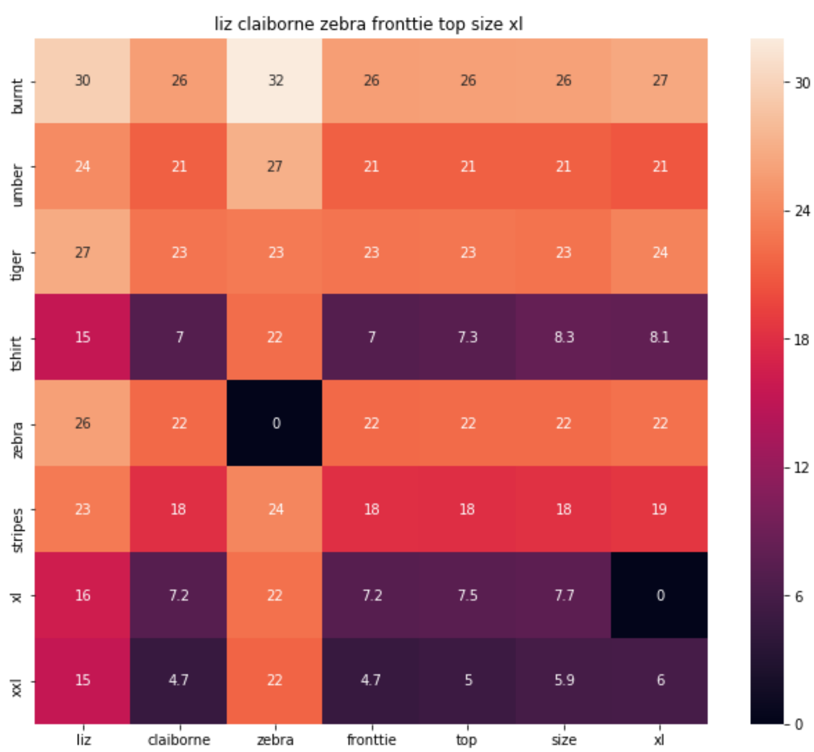

```
ASIN : B00C0I3U3E
Brand : Stanzino
euclidean distance from input : 6.4149
===========================================================================
==================================================
```



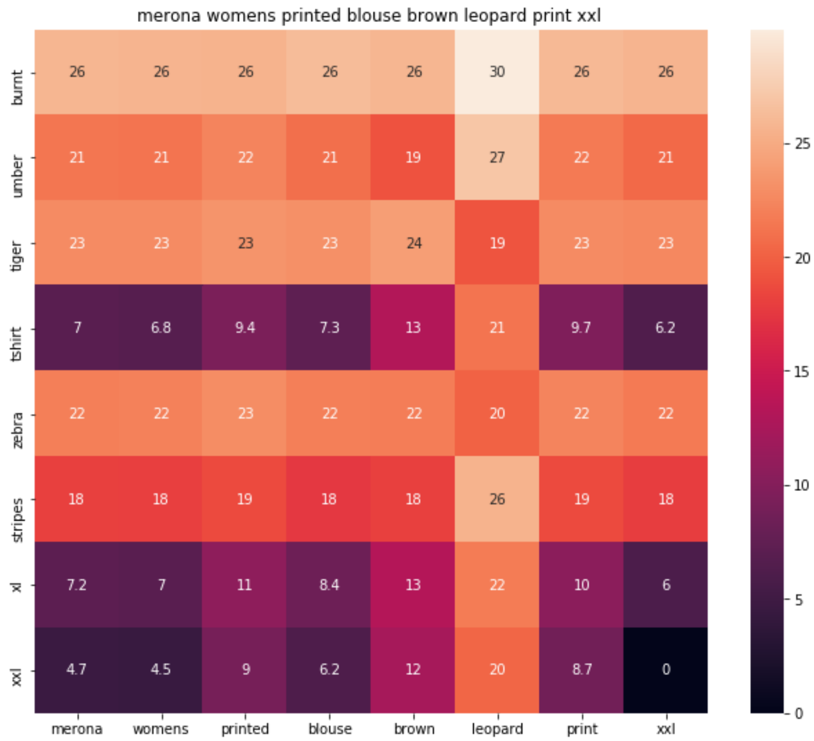

```
ASIN : B073R4ZM7Y
Brand : Colosseum
euclidean distance from input : 6.45096
===========================================================================
==================================================
```



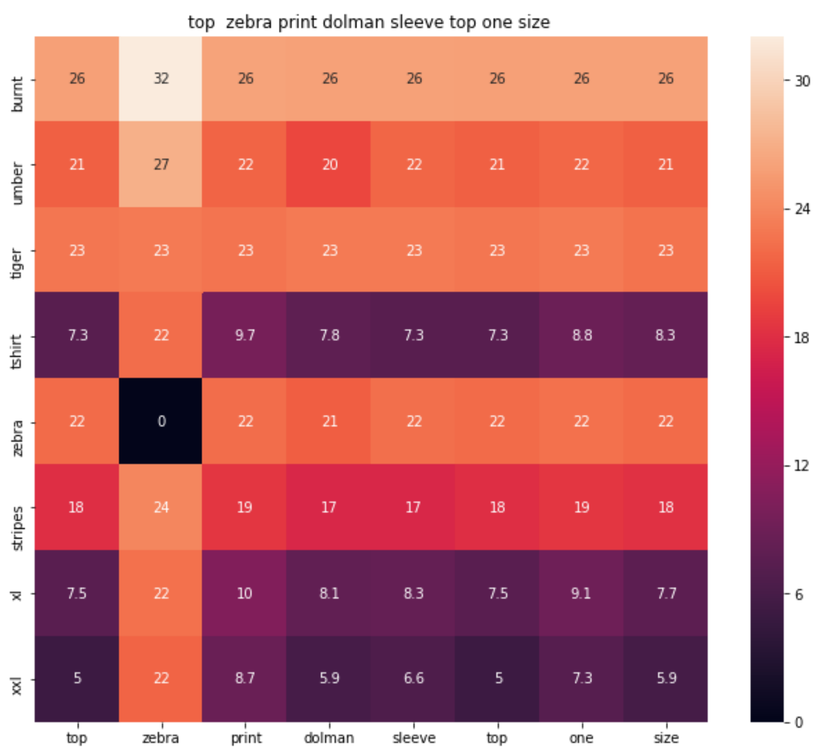

```
ASIN : B01C6ORLDQ
Brand : 1 Mad Fit
euclidean distance from input : 6.46341
===========================================================================
==================================================
```



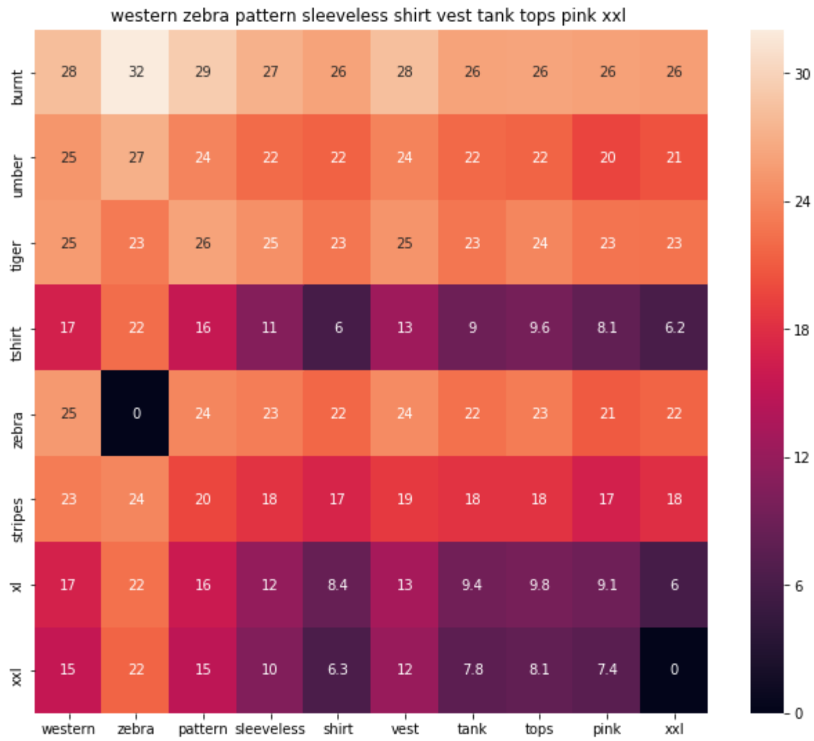

```
ASIN : B06XBY5QXL
Brand : Liz Claiborne
euclidean distance from input : 6.53922
===========================================================================
==================================================
```



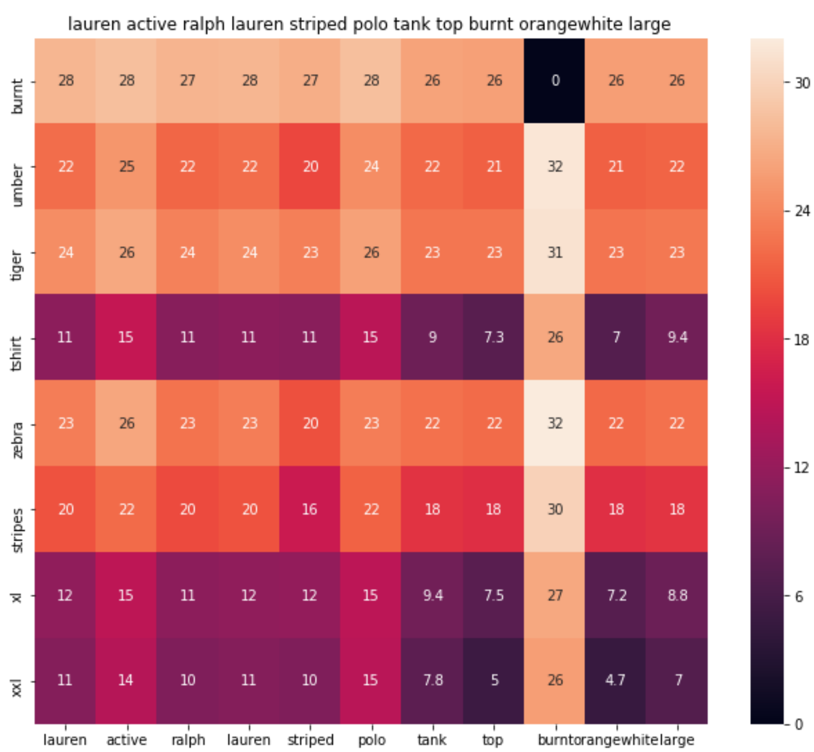

```
ASIN : B071YF3WDD
Brand : Merona
euclidean distance from input : 6.5755
===========================================================================
==================================================
```



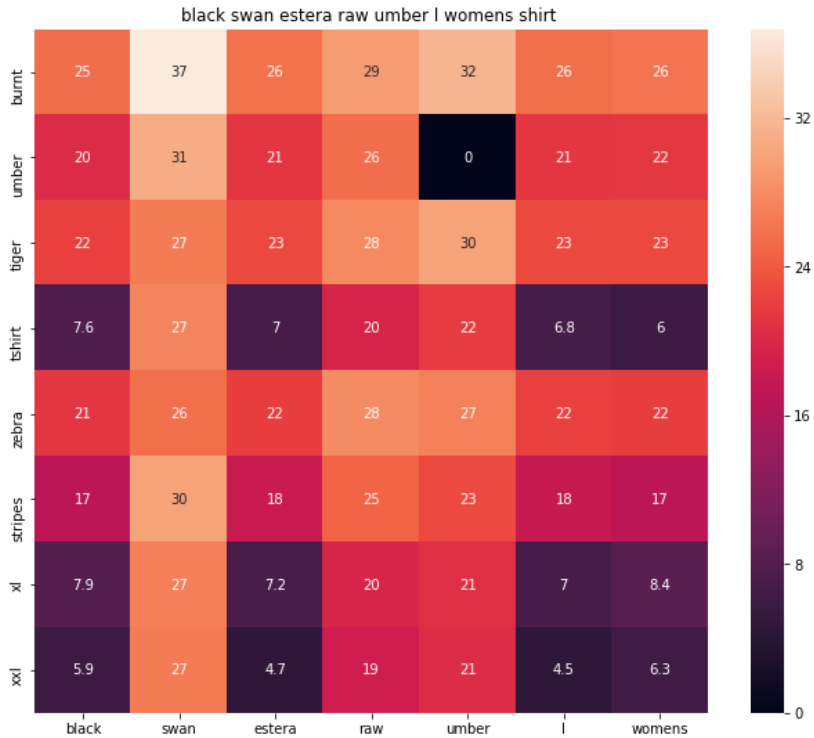

```
ASIN : B00H8A6ZLI
Brand : Vivian's Fashions
euclidean distance from input : 6.63821
===========================================================================
==================================================
```



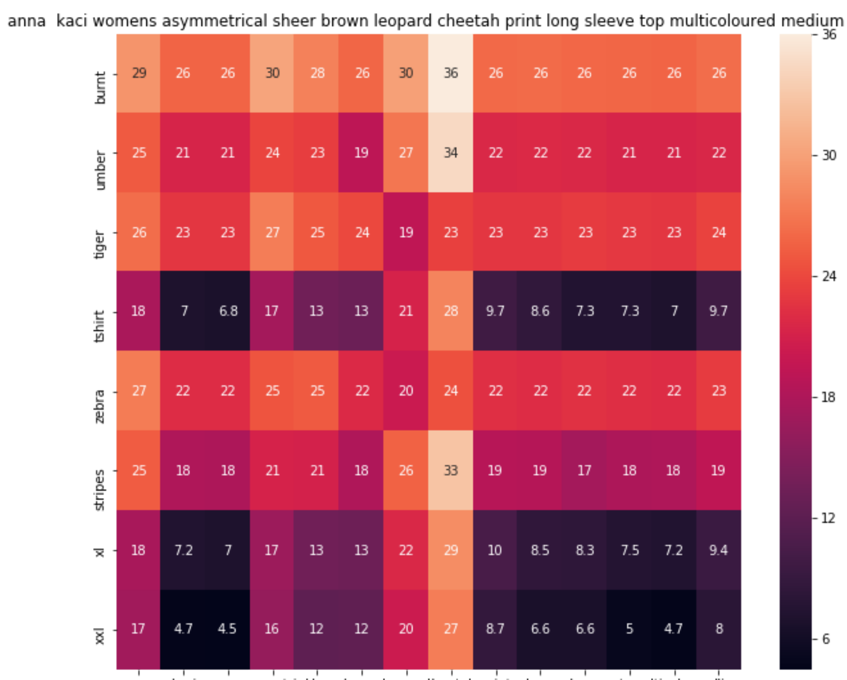

```
ASIN : B00Z6HEXWI
Brand : Black Temptation
euclidean distance from input : 6.66074
================================================================================================================================
```




```
ASIN : B00ILGH5OY
Brand : Ralph Lauren Active
euclidean distance from input : 6.68391
================================================================================================================================
```




```
ASIN : B06Y1VN8WQ
Brand : Black Swan
euclidean distance from input : 6.70576
===========================================================================
==================================================
```




```
ASIN : B00KSNTY7Y
Brand : Anna-Kaci
euclidean distance from input : 6.70612
===========================================================================
==================================================
```

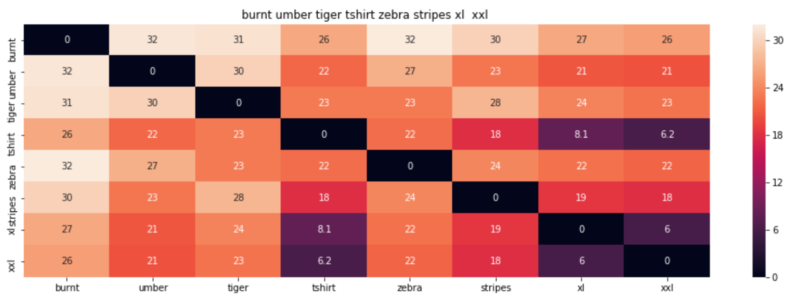
## [9.6] Weighted similarity using brand and color.

In [0]:
```
# 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 hypen
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)).to
csr()
```

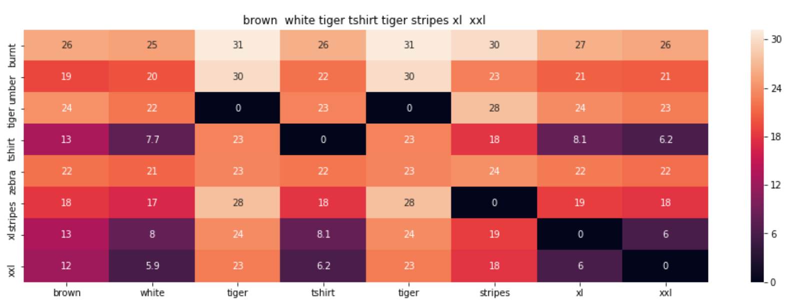
In [0]:
```
def heat_map_w2v_brand(sentance1, sentance2, url, doc_id1, doc_id2, df_id1,
df_id2, model):

    # 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 length 300 corresponds to each word in give title
    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 length 300 corresponds to each word in give title
    s2_vec = get_word_vec(sentance2, doc_id2, model)

    # s1_s2_dist = np.array(#number of words in title1 * #number of words i
n 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]], # input apparel's features
               [data['asin'].loc[df_id2],brands[doc_id2], colors[doc_id2],
types[doc_id2]]] # recommonded apparel's features

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

    # 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])
    # ploting 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 lins 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()
```

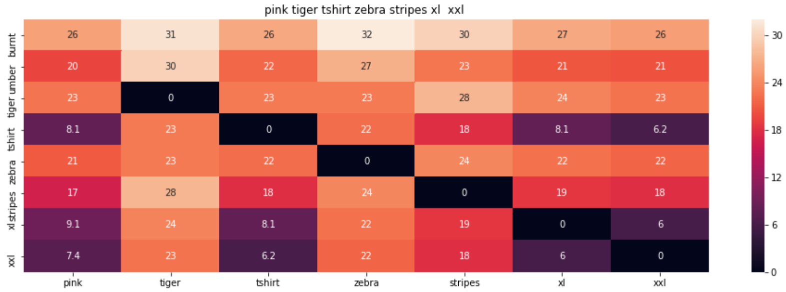
In [0]:
```
def idf_w2v_brand(doc_id, w1, w2, num_results):
    # doc_id: apparel's id in given corpus
    # w1: weight for  w2v features
    # w2: weight for brand and color features

    # 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[d
oc_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 + w
2)

    # 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'].l
oc[df_indices[i]], data['medium_image_url'].loc[df_indices[i]], indices[0],
indices[i],df_indices[0], df_indices[i], 'weighted')
        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 w
ords i, j
```



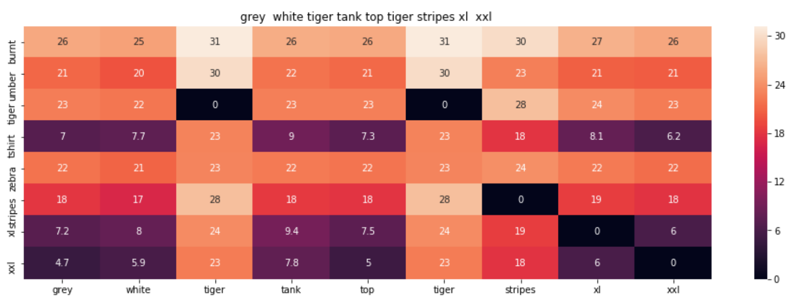

```
ASIN : B00JXQB5FQ
Brand : Si Row
euclidean distance from input : 0.001953125
============================================================================
===================================================
```



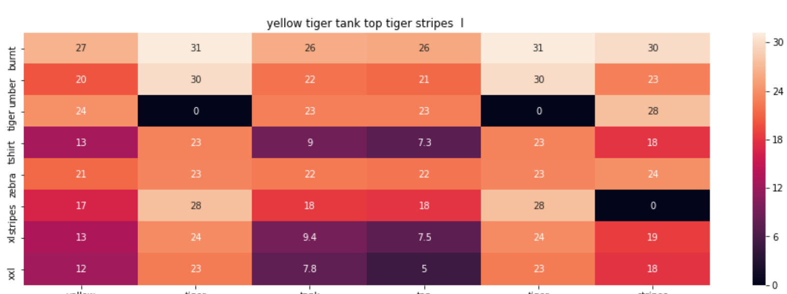

```
ASIN : B00JXQCWTO
Brand : Si Row
euclidean distance from input : 2.38547115326
============================================================================
===================================================
```



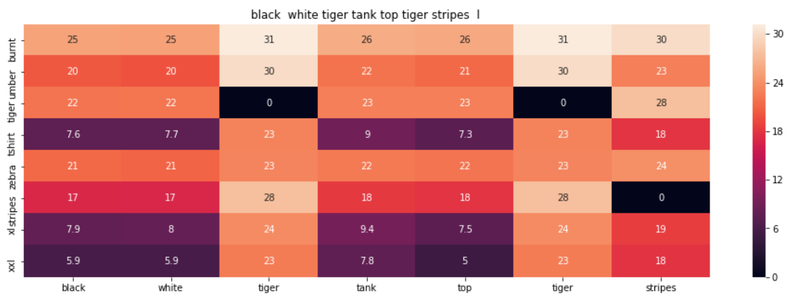

```
ASIN : B00JXQASS6
Brand : Si Row
euclidean distance from input : 2.73905105609
===========================================================================
==================================================
```



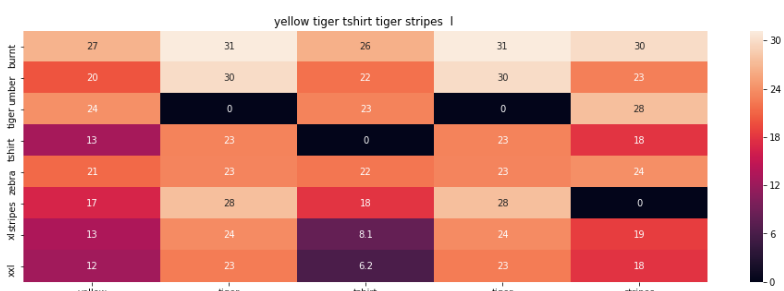

```
ASIN : B00JXQAFZ2
Brand : Si Row
euclidean distance from input : 3.387187195
===========================================================================
==================================================
```



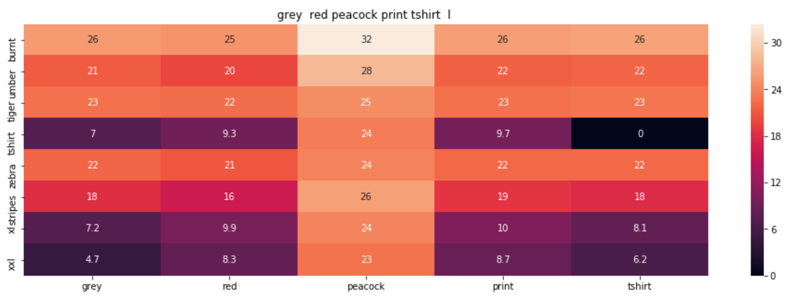

ASIN : B00JXQAUWA
Brand : Si Row
euclidean distance from input : 3.5518684389
==================================================================================================================================



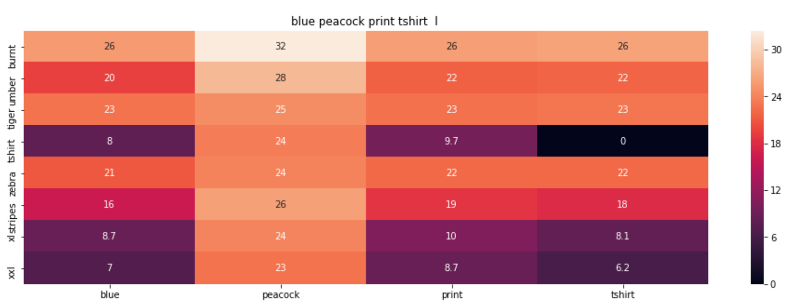

ASIN : B00JXQAO94
Brand : Si Row
euclidean distance from input : 3.5536174776
==================================================================================================================================



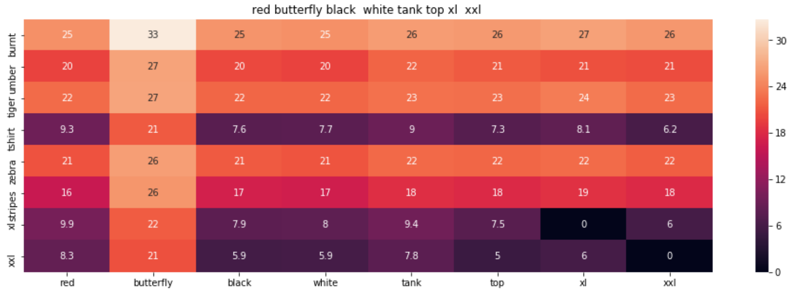

ASIN : B00JXQCUIC
Brand : Si Row
euclidean distance from input : 3.65382804889
==================================================================================================================================



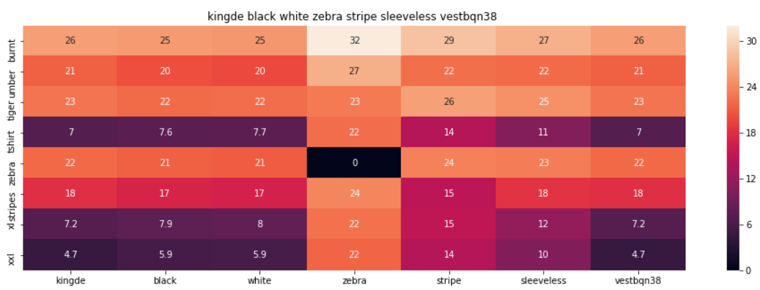

```
ASIN : B00JXQCFRS
Brand : Si Row
euclidean distance from input : 4.12881164569
===========================================================================
==================================================
```



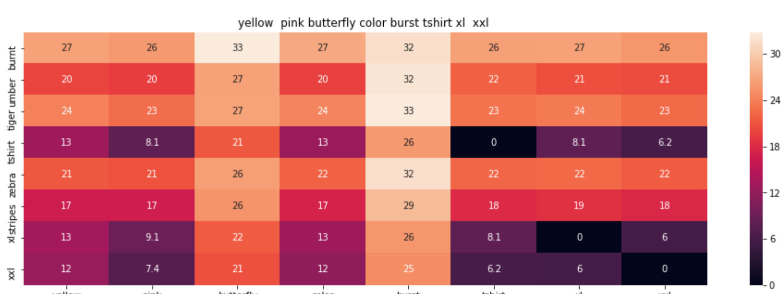

```
ASIN : B00JXQC8L6
Brand : Si Row
euclidean distance from input : 4.20390052813
===========================================================================
==================================================
```




```
ASIN : B00JV63CW2
Brand : Si Row
euclidean distance from input : 4.28658676166
=============================================================================
===================================================
```

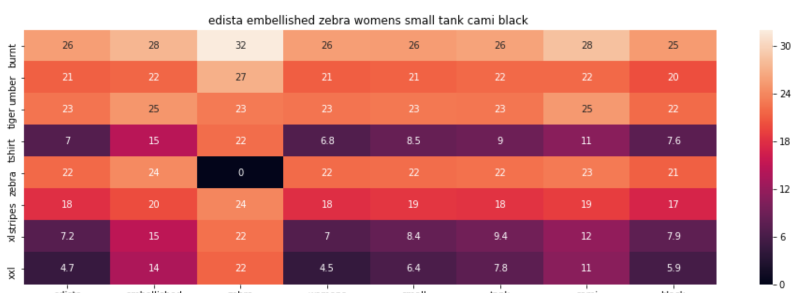



```
ASIN : B015H41F6G
Brand : KINGDE
euclidean distance from input : 4.38937078798
=============================================================================
===================================================
```

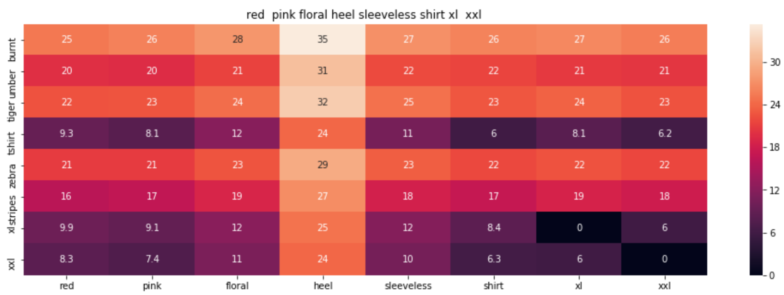



```
ASIN : B00JXQBBMI
Brand : Si Row
euclidean distance from input : 4.39790992755
================================================================================
=====================================================
```

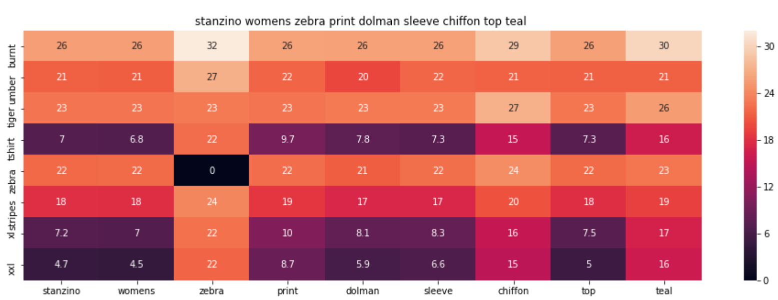

```
ASIN : B073R5Q8HD
Brand : Colosseum
euclidean distance from input : 4.45122858369
================================================================================
=====================================================
```



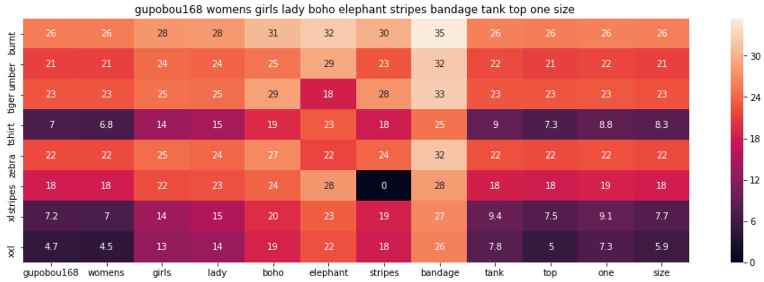

```
ASIN : B074P8MD22
Brand : Edista
euclidean distance from input : 4.51897779787
================================================================================
=====================================================
```



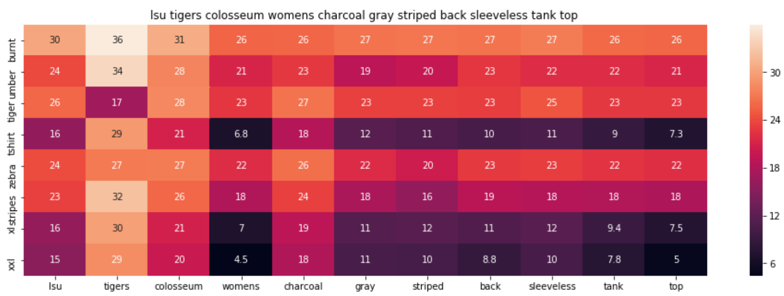

```
ASIN : B00JV63QQE
Brand : Si Row
euclidean distance from input : 4.52937545794
============================================================================
==================================================
```



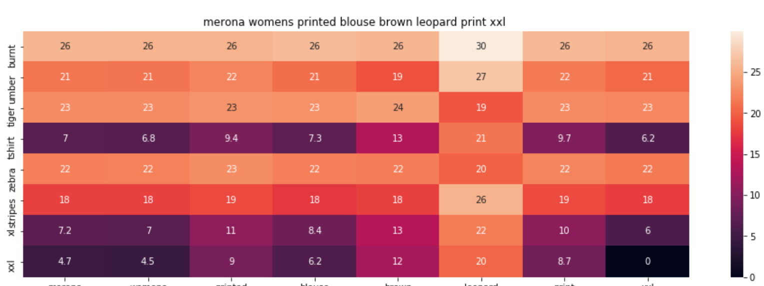

```
ASIN : B00C0I3U3E
Brand : Stanzino
euclidean distance from input : 4.53032614076
============================================================================
==================================================
```



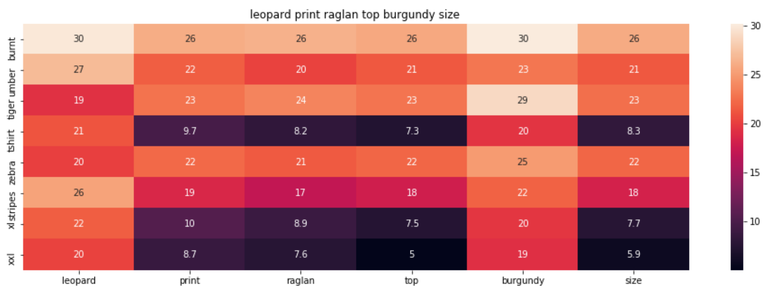

ASIN : B01ER184O6
Brand : GuPoBoU168
euclidean distance from input : 4.54681702403
=============================================================================================================================




ASIN : B073R4ZM7Y
Brand : Colosseum
euclidean distance from input : 4.54835554445
=============================================================================================================================

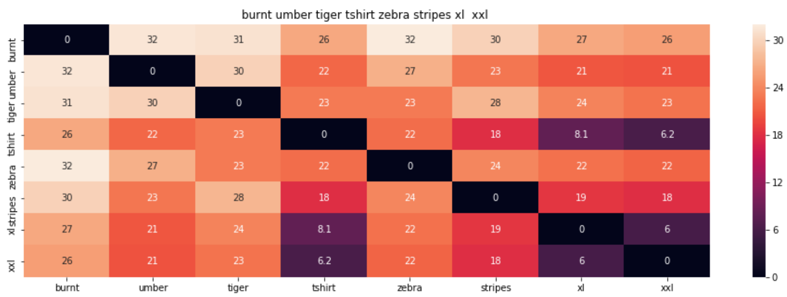



```
ASIN : B071YF3WDD
Brand : Merona
euclidean distance from input : 4.61062742555
================================================================================
=====================================================
```

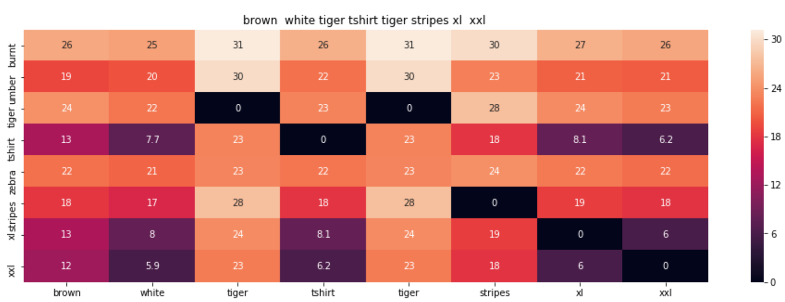



```
ASIN : B01C6ORLDQ
Brand : 1 Mad Fit
euclidean distance from input : 4.64591789282
================================================================================
=====================================================
```

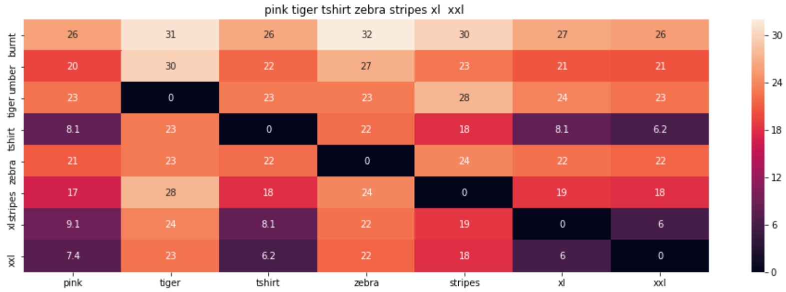
In [0]:
```
# brand and color weight =50
# title vector weight = 5

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



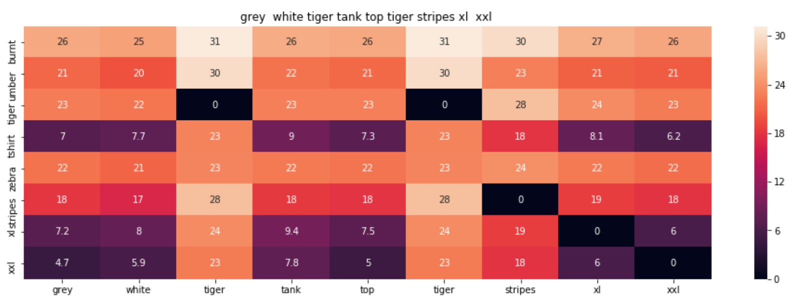

```
ASIN : B00JXQB5FQ
Brand : Si Row
euclidean distance from input : 0.000355113636364
================================================================================
=====================================================
```



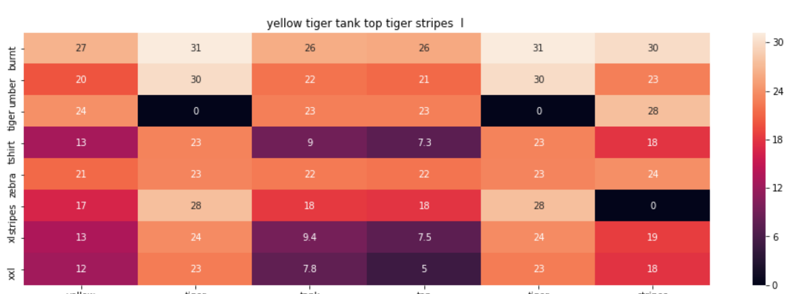

```
ASIN : B00JXQCWTO
Brand : Si Row
euclidean distance from input : 0.433722027865
================================================================================
=====================================================
```



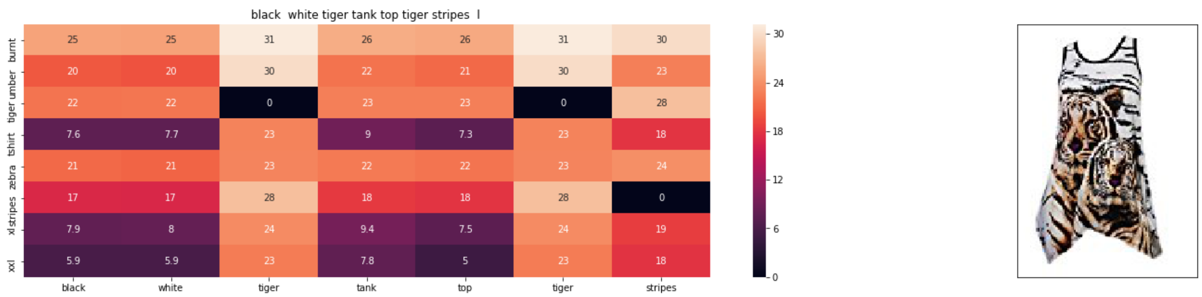

```
ASIN : B00JXQASS6
Brand : Si Row
euclidean distance from input : 1.65509310669
===========================================================================
==================================================
```



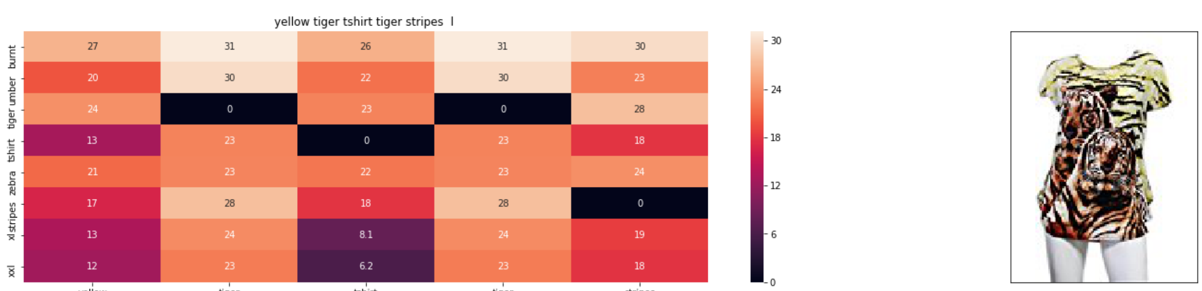

```
ASIN : B00JXQAFZ2
Brand : Si Row
euclidean distance from input : 1.77293604103
===========================================================================
==================================================
```



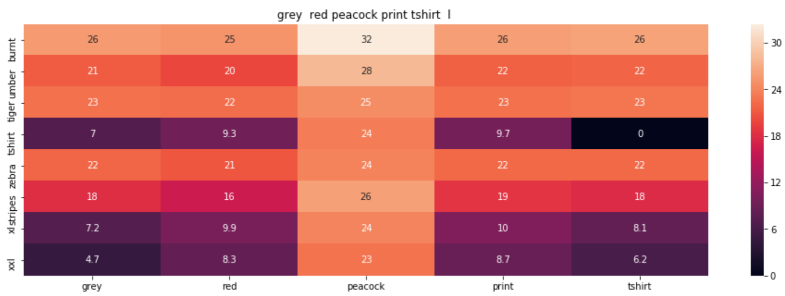

```
ASIN : B00JXQAUWA
Brand : Si Row
euclidean distance from input : 1.80287808538
============================================================================
==================================================
```



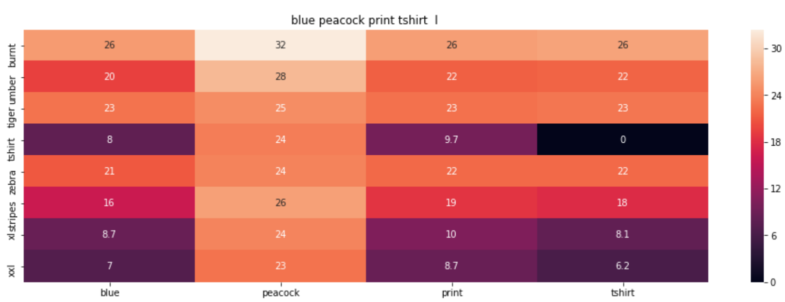

```
ASIN : B00JXQAO94
Brand : Si Row
euclidean distance from input : 1.80319609241
============================================================================
==================================================
```



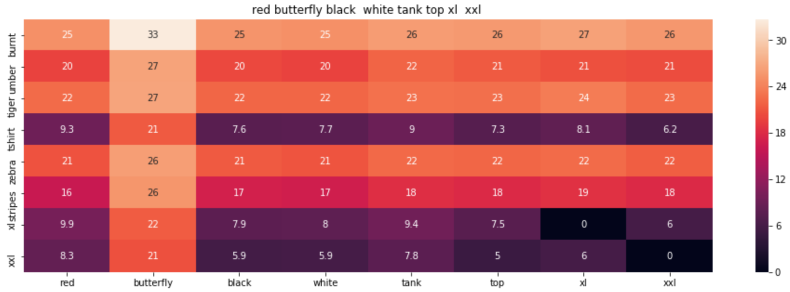

```
ASIN : B00JXQCUIC
Brand : Si Row
euclidean distance from input : 1.82141619628
============================================================================
==================================================
```



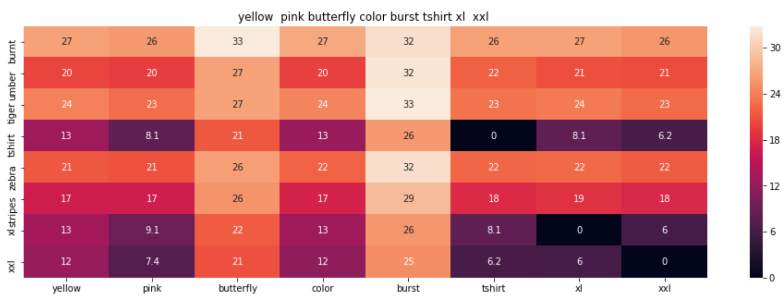

```
ASIN : B00JXQCFRS
Brand : Si Row
euclidean distance from input : 1.90777685025
===========================================================================
==================================================
```



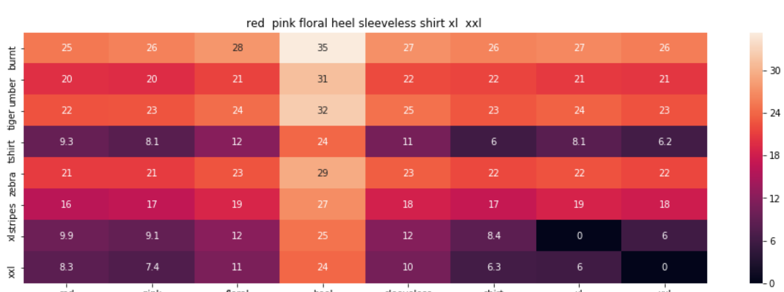

```
ASIN : B00JXQC8L6
Brand : Si Row
euclidean distance from input : 1.92142937433
===========================================================================
==================================================
```



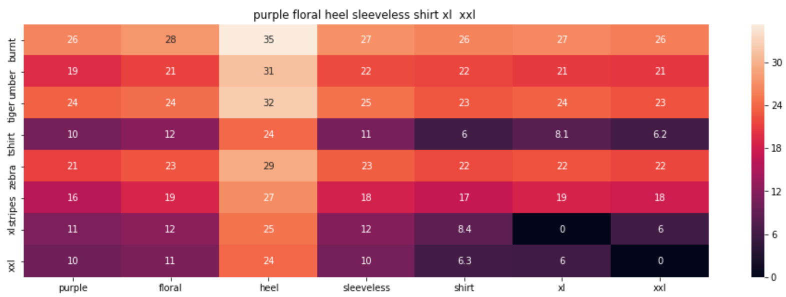

```
ASIN : B00JV63CW2
Brand : Si Row
euclidean distance from input : 1.93646323497
===========================================================================
==================================================
```



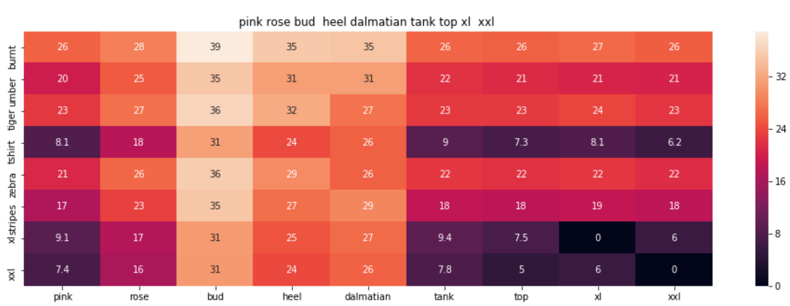

```
ASIN : B00JXQBBMI
Brand : Si Row
euclidean distance from input : 1.95670381059
===========================================================================
==================================================
```



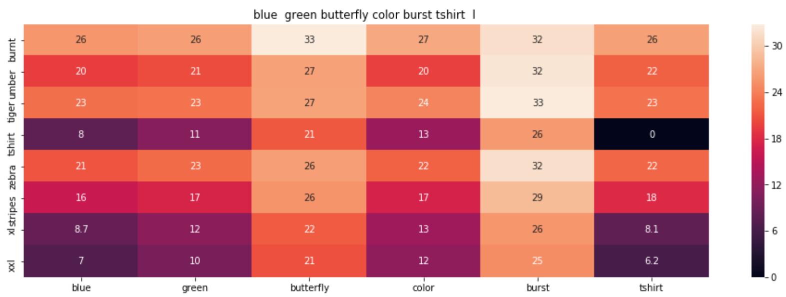

```
ASIN : B00JV63QQE
Brand : Si Row
euclidean distance from input : 1.9806066343
================================================================================
=====================================================
```



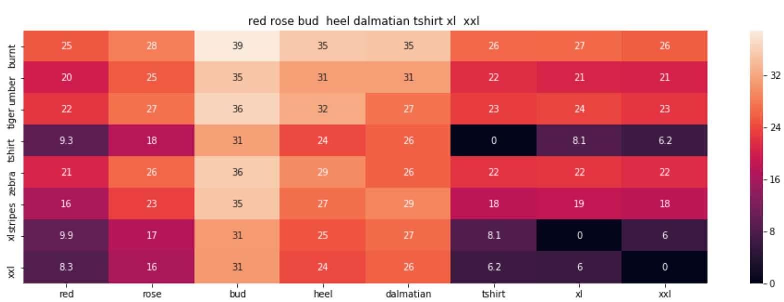

```
ASIN : B00JV63VC8
Brand : Si Row
euclidean distance from input : 2.01218559992
================================================================================
=====================================================
```



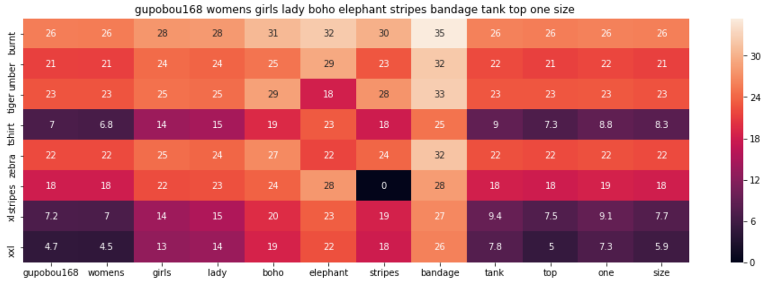

```
ASIN : B00JXQAX2C
Brand : Si Row
euclidean distance from input : 2.01335178755
================================================================================
=====================================================
```



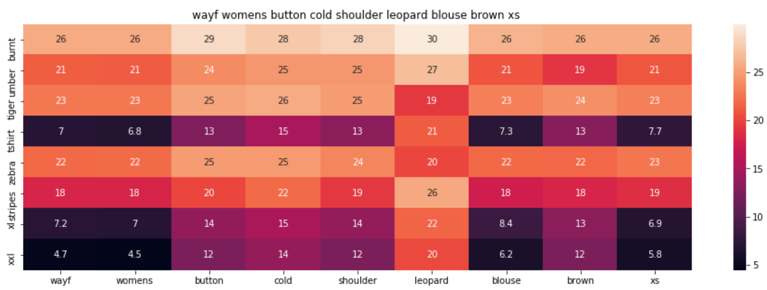

```
ASIN : B00JXQC0C8
Brand : Si Row
euclidean distance from input : 2.01388334827
================================================================================
=====================================================
```



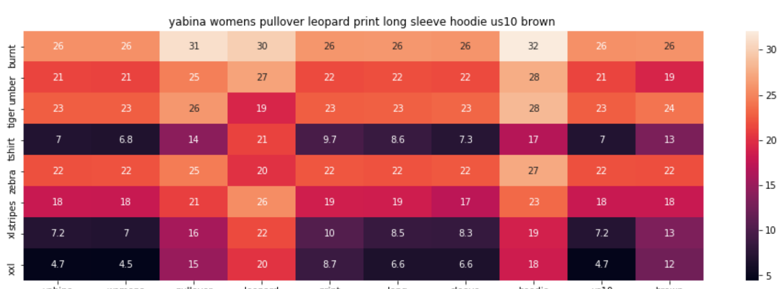

```
ASIN : B00JXQABB0
Brand : Si Row
euclidean distance from input : 2.0367257555
================================================================================
=====================================================
```



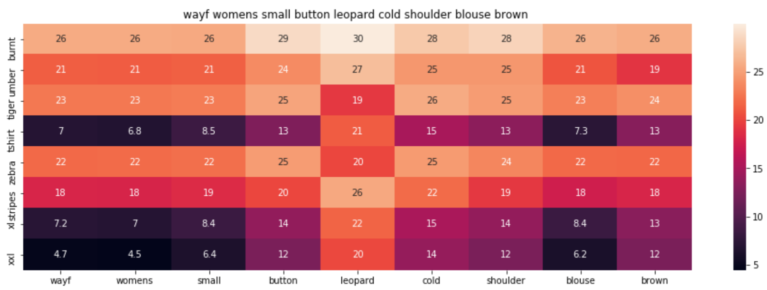

```
ASIN : B01ER184O6
Brand : GuPoBoU168
euclidean distance from input : 2.65620416778
=============================================================================
===================================================
```




```
ASIN : B01LZ7BQ4H
Brand : WAYF
euclidean distance from input : 2.6849067823
=============================================================================
===================================================
```




```
ASIN : B01KJUM6JI
Brand : YABINA
euclidean distance from input : 2.68583819266
===========================================================================
==================================================
```




```
ASIN : B01M06V4X1
Brand : WAYF
euclidean distance from input : 2.69476194865
===========================================================================
==================================================
```

# [10.2] Keras and Tensorflow to extract features

In [0]:

```python
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dropout, Flatten, Dense
from keras import applications
from sklearn.metrics import pairwise_distances
import matplotlib.pyplot as plt
import requests
from PIL import Image
import pandas as pd
import pickle
```

```
Using TensorFlow backend.
```

In [0]:
```
# https://gist.github.com/fchollet/f35fbc80e066a49d65f1688a7e99f069
# Code reference: https://blog.keras.io/building-powerful-image-classificat
ion-models-using-very-little-data.html



# 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 generat
e 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((16042,25
088))

    np.save(open('16k_data_cnn_features.npy', 'wb'), bottleneck_features_tr
ain)
```

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


save_bottlebeck_features()

'''
```

# [10.3] Visual features based product similarity.

In [0]:
```
#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, bottlenec
k_features_train[doc_id].reshape(1,-1))

    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[i
ndices[i]]]
        for indx, 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.amzon.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: 0.0
Amazon Url: www.amzon.com/dp/B00JXQB5FQ
```

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

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

```
Product Title:  brown  white tiger tshirt tiger stripes xl  xxl
Euclidean Distance from input image: 44.0002
Amazon Url: www.amzon.com/dp/B00JXQCWTO
```

Product Title:  kawaii pastel tops tees pink flower design
Euclidean Distance from input image: 47.3825
Amazon Url: www.amzon.com/dp/B071FCWD97

Product Title:  womens thin style tops tees pastel watermelon print
Euclidean Distance from input image: 47.7184
Amazon Url: www.amzon.com/dp/B01JUNHBRM

Product Title:  kawaii pastel tops tees baby blue flower design
Euclidean Distance from input image: 47.9021
Amazon Url: www.amzon.com/dp/B071SBCY9W

Product Title:  edv cheetah run purple multi xl
Euclidean Distance from input image: 48.0465
Amazon Url: www.amzon.com/dp/B01CUPYBM0

Product Title:  danskin womens vneck loose performance tee xsmall pink ombr
e
Euclidean Distance from input image: 48.1019
Amazon Url: www.amzon.com/dp/B01F7PHXY8

Product Title:  summer alpaca 3d pastel casual loose tops tee design
Euclidean Distance from input image: 48.1189
Amazon Url: www.amzon.com/dp/B01I80A93G


Product Title:  miss chievous juniors striped peplum tank top medium shadow
peach
Euclidean Distance from input image: 48.1313
Amazon Url: www.amzon.com/dp/B0177DM70S

Product Title:  red  pink floral heel sleeveless shirt xl  xxl
Euclidean Distance from input image: 48.1695
Amazon Url: www.amzon.com/dp/B00JV63QQE


```
Product Title:  moana logo adults hot v neck shirt black xxl
Euclidean Distance from input image: 48.2568
Amazon Url: www.amzon.com/dp/B01LX6H43D
```

```
Product Title:  abaday multicolor cartoon cat print short sleeve longline s
hirt large
Euclidean Distance from input image: 48.2657
Amazon Url: www.amzon.com/dp/B01CR57YY0
```

```
Product Title:  kawaii cotton pastel tops tees peach pink cactus design
Euclidean Distance from input image: 48.3626
Amazon Url: www.amzon.com/dp/B071WYLBZS
```


```
Product Title:  chicago chicago 18 shirt women pink
Euclidean Distance from input image: 48.3836
Amazon Url: www.amzon.com/dp/B01GXAZTRY
```

Product Title:  yichun womens tiger printed summer tshirts tops
Euclidean Distance from input image: 48.4493
Amazon Url: www.amzon.com/dp/B010NN9RXO

Product Title:  nancy lopez whimsy short sleeve  whiteblacklemon drop  xs
Euclidean Distance from input image: 48.4788
Amazon Url: www.amzon.com/dp/B01MPX6IDX

Product Title:  womens tops tees pastel peach ice cream cone print
Euclidean Distance from input image: 48.558
Amazon Url: www.amzon.com/dp/B0734GRKZL

Product Title:  uswomens mary j blige without tshirts shirt
Euclidean Distance from input image: 48.6144
Amazon Url: www.amzon.com/dp/B01M0XXFKK

# Assignment - Apparel Recommendation

## Apply Weighted Euclidean Distance measure on the feature set

#### The feature set should include Text, Brand, color and Image

#### Text Features should be - IDF weighted Word2Vec

#### Brand and Color should be - one-hot encoded

#### Image constructed - VGG16

#### Give weights Wt,Wb,Wc and Wi to the above features

## Try to build a Weighted Euclidean Distance based similarity and display the recommendations

## Further you can also try multiple values for the four different weights.

## Display and note down how the Recommendations change

# Utility functions

In [1]:
```
#from gensim.models import Word2Vec
#from gensim.models import KeyedVectors
import pickle

# 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 representat
ion 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.vocabula
ry_:
                vec.append(idf_title_features[doc_id, idf_title_vectorizer.
vocabulary_[i]] * model[i])
            elif m_name == 'avg':
                vec.append(model[i])
        else:
            # if the word in our courpus is not there in the google word2ve
c corpus, we are just ignoring it
            vec.append(np.zeros(shape=(300,)))
    # we will return a numpy array of shape (#number of words in title * 30
0 ) 300 = len(w2v_model[word])
    # each row represents the word2vec representation of each word (weighte
d/avg) in given sentance
    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 corresponds to each word in give title
    # vec2 = np.array(#number_of_words_title2 * 300), each row is a vector
 of length 300 corresponds to each word in give title

    final_dist = []
    # for each vector in vec1 we caluclate the distance(euclidean) to all v
ectors in vec2
    for i in vec1:
        dist = []
        for j in vec2:
            # np.linalg.norm(i-j) will result the euclidean distance betwee
n 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 i
n 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 length 300 corresponds to each word in give title
    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 length 300 corresponds to each word in give title
    s2_vec = get_word_vec(sentence2, doc_id2, model)

    # s1_s2_dist = np.array(#number of words in title1 * #number of words i
n 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 d
isplays image of apparel
    gs = gridspec.GridSpec(2, 2, width_ratios=[4,1],height_ratios=[2,1])
    fig = plt.figure(figsize=(15,15))

    ax = plt.subplot(gs[0])
    # ploting 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()


# this function will add the vectors of each word and returns the avg vecto
r 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 representat
ion 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 intialize a vector of size 300 with all zeros
    # we add each word2vec(wordi) to this fetureVec
    nwords = 0

    for word in sentence.split():
        nwords += 1
        if word in vocab:
            if m_name == 'weighted' and word in  idf_title_vectorizer.vocab
ulary_:
                featureVec = np.add(featureVec, idf_title_features[doc_id,
idf_title_vectorizer.vocabulary_[word]] * model[word])
            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
```

## Text feature - idf weighted Word2Vec

In [3]:
```
fname ='E:/appliedaiacourse/assignments/asign-24-apparelrcmnd/'
data = pd.read_pickle(fname+'16k_apperal_data_preprocessed')
data.head()
```

Out[3]:

| | asin | brand | color | medium_image_url | product_type_name | |
|---|---|---|---|---|---|---|
| **4** | B004GSI2OS | FeatherLite | Onyx Black/ Stone | https://images-na.ssl-images-amazon.com/images... | SHIRT | featherli ladies lo sleeve s resistan |
| **6** | B012YX2ZPI | HX-Kingdom Fashion T-shirts | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | womens unique ' cotton special olympics wor... |
| **15** | B003BSRPB0 | FeatherLite | White | https://images-na.ssl-images-amazon.com/images... | SHIRT | featherli ladies moisture free mes sport sh |
| **27** | B014ICEJ1Q | FNC7C | Purple | https://images-na.ssl-images-amazon.com/images... | SHIRT | superna chibis sa dean ca neck tsh |
| **46** | B01NACPBG2 | Fifth Degree | Black | https://images-na.ssl-images-amazon.com/images... | SHIRT | fifth deg womens gold foil graphic jun... |

In [4]:
```
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_points * #words_in_corpus
# idf_title_features[doc_id, index_of_word_in_corpus] = number of times the
word occured in that doc
```

In [5]:
```
def n_containing(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)))
```

In [6]:
```
# 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 w
ith the idf values of the word
    # idf_title_features[:, idf_title_vectorizer.vocabulary_[i]].nonzero()
[0] will return all documents in which the word i present
    for j in idf_title_features[:, idf_title_vectorizer.vocabulary_[i]].non
zero()[0]:

        # we replace the count values of word i in document j with  idf_val
ue of word i
        # idf_title_features[doc_id, index_of_word_in_courpus] = idf value
 of word
        idf_title_features[j,idf_title_vectorizer.vocabulary_[i]] = idf_val
```

In [7]:
```
doc_id = 0
w2v_title_weight = []

with open(fname+'word2vec_model', 'rb') as handle:
    model = pickle.load(handle)

vocab = model.keys()

# 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 correspo
nds to a doc
w2v_title_weight = np.array(w2v_title_weight)
```

In [59]:
```
def get_simil_items(doc_id, w1, w2, w3, w4, 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[d
oc_id].reshape(1,-1))
    brand_feat_dist = pairwise_distances(brand_features, brand_features[doc
_id].reshape(1,-1))
    color_feat_dist = pairwise_distances(color_features, color_features[doc
_id].reshape(1,-1))
    img_feat_dist = pairwise_distances(img_feat, img_feat[doc_id].reshape(1
,-1))
    pairwise_dist   = (w1 * idf_w2v_dist +  w2 * brand_feat_dist+ w3 * colo
r_feat_dist + w4 * img_feat_dist )/float(w1 + w2 + w3 + w4)

    # 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])

    print('Queried Data point')
    qry_asin = asins.index(df_asins[doc_id])
    qry_row = data[['medium_image_url','title','brand','color','asin']].loc
[data['asin']== asins[qry_asin]]
    print("-"*50)
    for indx, row in qry_row.iterrows():
        print('Product Title: ', row['title'])
        print('Brand :',row['brand'])
        print('Color :',row['color'])
        print('Amazon Url: www.amzon.com/dp/'+ row['asin'])
        display(Image(url=row['medium_image_url'], embed=True))

    print("-"*50)

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

## Brand,Type, Color - One-hot Encoded

In [18]:
```
# 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 hypen
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)).to
csr()
```

## Image - VGG16

In [71]:
```
#load the features and corresponding ASINS info.
fname ='E:/appliedaiacourse/assignments/asign-24-apparelrcmnd/'
img_feat = np.load(fname+'16k_data_cnn_features.npy')
asins = np.load(fname+'16k_data_cnn_feature_asins.npy')
asins = list(asins)

# load the original 16K dataset
df_asins = list(data['asin'])
```

In [19]:
```
print(type(w2v_title_weight))
print(type(brand_features))
print(type(color_features))
print(type(img_feat))

print(w2v_title_weight.shape)
print(brand_features.shape)
print(color_features.shape)
print(img_feat.shape)
brand_features = brand_features.toarray()
color_features = color_features.toarray()
print(type(w2v_title_weight))
print(type(brand_features))
print(type(color_features))
print(type(img_feat))
```

```
<class 'numpy.ndarray'>
<class 'scipy.sparse.csr.csr_matrix'>
<class 'scipy.sparse.csr.csr_matrix'>
<class 'numpy.ndarray'>
(16042, 300)
(16042, 3835)
(16042, 1845)
(16042, 25088)
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

In [20]:
```
joblib.dump(w2v_title_weight,fname+'w2v_title_wt.pkl')
joblib.dump(brand_features,fname+'brand_features.pkl')
joblib.dump(color_features,fname+'color_features.pkl')
```

Out[20]:
```
['E:/appliedaiacourse/assignments/asign-24-apparelrcmnd/color_features.pk
l']
```

In [22]:
```
fname ='E:/appliedaiacourse/assignments/asign-24-apparelrcmnd/'
w2v_title_weight = joblib.load(fname+'w2v_title_wt.pkl')
brand_features = joblib.load(fname+'brand_features.pkl')
color_features = joblib.load(fname+'color_features.pkl')
#load the features and corresponding ASINS info.
fname ='E:/appliedaiacourse/assignments/asign-24-apparelrcmnd/'
img_feat = np.load(fname+'16k_data_cnn_features.npy')
asins = np.load(fname+'16k_data_cnn_feature_asins.npy')
asins = list(asins)

# load the original 16K dataset
df_asins = list(data['asin'])
```

In [23]:
```
X_train = np.hstack((w2v_title_weight, brand_features,color_features, img_f
eat))
```

In [24]:
```
print(X_train.shape)
```

```
(16042, 31068)
```

# Weigthed Euclidean Distance Similarity

## Example 1

### Providing equal weightage to all features

In [60]:
```
get_simil_items(12566, 5, 5,5,5, 20)
```

```
Queried Data point
--------------------------------------------------
Product Title:  burnt umber tiger tshirt zebra stripes xl  xxl
Brand : Si Row
Color : Brown
Amazon Url: www.amzon.com/dp/B00JXQB5FQ
```

```
--------------------------------------------------
```

```
Product Title:  foxcroft nyc womens pinpoint oxford shirt noniron stretch p
oplin blouse xlarge white
Brand : Foxcroft NYC
Color : White
Euclidean Distance from input image: 0.012025105953216552
Amazon Url: www.amzon.com/dp/B072277HVB
```

```
Product Title:  kiind longsleeve swing top white
Brand : KIIND OF
Color : White
Euclidean Distance from input image: 12.145862058447523
Amazon Url: www.amzon.com/dp/B0142Q3C6G
```

```
Product Title:  free people scoopneck highlow thermal tunic oatmeal heather
Brand : Free People
Color : Oatmeal Heather
Euclidean Distance from input image: 12.611821225927994
Amazon Url: www.amzon.com/dp/B00ZH2X45E
```

```
Product Title:  karen scott wmen short sleeve henley top plus size 0x brigh
t white
Brand : Karen Scott
Color : Bright White
Euclidean Distance from input image: 12.861254886616171
Amazon Url: www.amzon.com/dp/B0196H7P5A
```

```
Product Title:  karen scott womens plus short sleeve solid henley top white
1x
Brand : Karen Scott
Color : Bright White
Euclidean Distance from input image: 12.864219856352552
Amazon Url: www.amzon.com/dp/B0196H3Z2W
```

Product Title:  masion jules longsleeve paneled sweatshirt xxl grey
Brand : Maison Jules
Color : Grey
Euclidean Distance from input image: 12.92604751586914
Amazon Url: www.amzon.com/dp/B06X16WTFX

Product Title:  jessica simpson womens frida peasant top antiquewhite x sma
ll
Brand : Fancy Jessica Simpson
Color : Antique/White
Euclidean Distance from input image: 12.946235777849262
Amazon Url: www.amzon.com/dp/B01KW297J0

Product Title:  vitamina usa soft vneckline belted blouse st6356 wht
Brand : Vitamina USA
Color : White
Euclidean Distance from input image: 13.170262718200684
Amazon Url: www.amzon.com/dp/B0155ICSDS

Product Title:  dg2 diane gilman turtleneck fleece pullover pockets small gray
Brand : DG 2 Diane Gilman
Color : Grey
Euclidean Distance from input image: 13.288906599992817
Amazon Url: www.amzon.com/dp/B01MSP0KAM

Product Title:  bb dakota womens plus size tereza top optic white 2x
Brand : BB Dakota
Color : Optic White
Euclidean Distance from input image: 13.314244772905415
Amazon Url: www.amzon.com/dp/B00JG6EBEA

Product Title:  american living womens dipdye peplum peplum top red l
Brand : American Home Living
Color : Red Multi
Euclidean Distance from input image: 13.319847869963391
Amazon Url: www.amzon.com/dp/B01LYFIRIL

```
Product Title:  jenni kayne womens crepe tee white small
Brand : Jenni Kayne
Color : White
Euclidean Distance from input image: 13.322205664629047
Amazon Url: www.amzon.com/dp/B07531DL6X
```

```
Product Title:  view walter baker long slv knit top a263076 white
Brand : Walter Baker
Color : White
Euclidean Distance from input image: 13.377500394629163
Amazon Url: www.amzon.com/dp/B074Q572HW
```

```
Product Title:  red house  ladies nailhead noniron buttondown shirt3xl whit
e
Brand : Red House
Color : Red
Euclidean Distance from input image: 13.395344473833148
Amazon Url: www.amzon.com/dp/B008LOSLYO
```

```
Product Title:  slinky 34slv ruffle scoopneck tee 532020 white l
Brand : Slinky
Color : White
Euclidean Distance from input image: 13.400631192969007
Amazon Url: www.amzon.com/dp/B0759W1F8F
```

```
Product Title:  robert graham womens blouse trish lavender small
Brand : Robert Graham
Color : Lavender
Euclidean Distance from input image: 13.404460074419086
Amazon Url: www.amzon.com/dp/B071NQ52LM
```

```
Product Title:  size comfortblend shirred crewneck longsleeve tshirt white
2x
Brand : Just My Size
Color : White
Euclidean Distance from input image: 13.407314612382953
Amazon Url: www.amzon.com/dp/B01N5YHV9R
```

```
Product Title:  isaac mizrahi woven tunic multicolored soutache a223828 pin
k purple
Brand : Isaac Mizrahi Live!
Color : Pink Purple
Euclidean Distance from input image: 13.444131400012335
Amazon Url: www.amzon.com/dp/B074HFNCWK
```

```
Product Title:  elizabeth james nevada tunic women vanilla size xs xsmall
Brand : Elizabeth and James
Color : Vanilla
Euclidean Distance from input image: 13.46192550668214
Amazon Url: www.amzon.com/dp/B071NP9WJL
```

```
Product Title:  jm collection womens bright white top size xl
Brand : JM Collection
Color : Bright White
Euclidean Distance from input image: 13.490855147356097
Amazon Url: www.amzon.com/dp/B01N4NYF25
```

# Example 2

### Only Brand has the highest weight and all the other features Text, Color and Image have been excluded

In [69]:
```
get_simil_items(12566, 0, 90,0,0, 20)
```

```
Queried Data point
--------------------------------------------------
Product Title:  burnt umber tiger tshirt zebra stripes xl  xxl
Brand : Si Row
Color : Brown
Amazon Url: www.amzon.com/dp/B00JXQB5FQ
```

```
--------------------------------------------------
```

```
Product Title:  fylo ladies size medium 14 zip vneck blouse pinkgrey animal
Brand : f
Color : Pink/Grey Animal
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B073X71H8K
```

```
Product Title:  romeo  juliet navy womens medium jeweled knit top green
Brand : Romeo & Juliet Couture
Color : Green
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B071P49MCK
```

Product Title:  blouseuploter cute cartoon vest summer pajamas
Brand : UPLOTER
Color : White
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B01LPJRYI0

Product Title:  kirkland signature ladies striped active yoga tee large gre
en
Brand : Kirkland Signature
Color : Green
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B01LXO8SGX

Product Title:  maison margiela mm6 beige 34 sleeve womens blouse top us 42
Brand : Maison Margiela MM6
Color : Beige
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B01N7OM7H4

```
Product Title:  kirkland signature ladies size medium cotton vneck tee whit
e
Brand : KS
Color : White
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B073ZRD5FL
```

```
Product Title:  ideology womens small crewneck solid tee tshirt black
Brand : Ideology
Color : Black
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B0721T22DL
```

```
Product Title:  material girl womens juniors cutout swing shift caviar blac
k combo xxl
Brand : Material Girl
Color : Caviar Black Combo
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B072L5CGR3
```

```
Product Title:  miss chievous womens short sleeve knit top wsharkbite hem c
harcoal 1x
Brand : Miss Chievous
Color : Charcoal
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B06XR9NTVL
```

```
Product Title:  foxcroft nyc womens pinpoint oxford shirt noniron stretch p
oplin blouse xlarge white
Brand : Foxcroft NYC
Color : White
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B072277HVB
```


```
Product Title:  divas love snow henley long sleeve tshirt 4xlarge violettur
quoise
Brand : Divas SnowGear
Color : Violet/Turquoise
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B071NHMRB2
```

```
Product Title:  socialite teal women medium crochet offshoulder blouse blue
Brand : Socialite
Color : Blue
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B01M0DVT93
```

```
Product Title:  tahari asl ruffledfront polkadot top blackwhite xl
Brand : Tahari ASL
Color : Black
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B074MZTZS5
```

```
Product Title:  exotic india yellow gray jamawar wrap fauxfur collar
Brand : Exotic India
Color : Gray
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B073ZHRBV8
```

Product Title:  hippie rose juniors marled dolmansleeve tunic top size
Brand : Hippie Rose
Color : Russian Teal Combo
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B072HXX3W7


Product Title:  womans wicked musical poster shirts red xxl
Brand : vcord
Color : Red
Euclidean Distance from input image: 0.0
Amazon Url: www.amzon.com/dp/B01KUN4VWE

Product Title:  beige black  white striped lightweight poncho top
Brand : Luxury Divas
Color : Multicolor
Euclidean Distance from input image: 1.4142135623730951
Amazon Url: www.amzon.com/dp/B01EESHW3G

Product Title:  anna kaci sm fit classic high collared front zip sheer over all top
Brand : Anna-Kaci
Color : Grey
Euclidean Distance from input image: 1.4142135623730951
Amazon Url: www.amzon.com/dp/B018J0515G


```
Product Title:  vansty love dad f2 100 cotton shirt women pink size xl
Brand : Vansty
Color : Pink
Euclidean Distance from input image: 1.4142135623730951
Amazon Url: www.amzon.com/dp/B01EJS5AIK
```

```
Product Title:  foi indian cotton vintage pattern hippie top blusa dress bl
ouse women ehs
Brand : Fashion of India
Color : Multi
Euclidean Distance from input image: 1.4142135623730951
Amazon Url: www.amzon.com/dp/B073C9B6T5
```

## Example 3

#### Color has the highest weight , Text and Image feature have equal weights and Brand feature excluded

In [67]:
```
get_simil_items(12566, 5, 0,90,5, 20)
```

Queried Data point
---------------------------------------------------
Product Title:  burnt umber tiger tshirt zebra stripes xl  xxl
Brand : Si Row
Color : Brown
Amazon Url: www.amzon.com/dp/B00JXQB5FQ

---------------------------------------------------

Product Title:  foxcroft nyc womens pinpoint oxford shirt noniron stretch poplin blouse xlarge white
Brand : Foxcroft NYC
Color : White
Euclidean Distance from input image: 0.0024050211906433104
Amazon Url: www.amzon.com/dp/B072277HVB

Product Title:  shop flash slimming body shaping pro performance muscle womens tank top white xxlarge
Brand : Shop Flash
Color : White
Euclidean Distance from input image: 2.641324920654297
Amazon Url: www.amzon.com/dp/B00TO30Z2W

Product Title:  felina lingerie cotton stretch camisole w hint modal assorted colors medium mint green
Brand : Felina Lingerie
Color : Mint Green
Euclidean Distance from input image: 2.6922734069824217
Amazon Url: www.amzon.com/dp/B0716PDM8C

Product Title:  boden womens stitched craft top us sz 4 pearl
Brand : BODEN
Color : Pearl
Euclidean Distance from input image: 2.81744327545166
Amazon Url: www.amzon.com/dp/B01JN90OMW

Product Title:  ladies  sleeve hooded slub tee white 2xl
Brand : J. America
Color : White
Euclidean Distance from input image: 2.9034089279174804
Amazon Url: www.amzon.com/dp/B00CO8F6XW

Product Title:  drew womens tweed knit audie top sz ivorygrey 230135f
Brand : Drew Shoe
Color : Ivory/Grey
Euclidean Distance from input image: 2.9046815490722655
Amazon Url: www.amzon.com/dp/B01ETLYQ6E

Product Title:  janette plus women knit top short sleeves collared neck yellow plaid size 1xl
Brand : Janette Plus
Color : Yellow
Euclidean Distance from input image: 2.9203470611572264
Amazon Url: www.amzon.com/dp/B0756RB1JC

Product Title:  st john liquid satin tank top small
Brand : Sandra Ingrish
Color : Cream
Euclidean Distance from input image: 2.9209498977661132
Amazon Url: www.amzon.com/dp/B07237HGYL

Product Title:  black casual hollow sleeveless lace blouse bky8054
Brand : bankhunyabangyai store
Color : Black
Euclidean Distance from input image: 2.9299710845947264
Amazon Url: www.amzon.com/dp/B00UP2467G

Product Title:  ana shortsleeve boyfriend tee  plus size 2 extra large colo
r white
Brand : A.N.A Shirt
Color : White
Euclidean Distance from input image: 2.9393921661376954
Amazon Url: www.amzon.com/dp/B01MSN4CTE

Product Title:  devon  jones ladies crown collection whtgrphlt grph xxxlarg
e
Brand : Devon & Jones
Color : Wht/Grph/Lt Grph
Euclidean Distance from input image: 2.9409006881713866
Amazon Url: www.amzon.com/dp/B01GGAEPRG

Product Title:  baishitop hollow crochet loose cover blouse tops shirt biki
ni swimwear women
Brand : Baishitop
Color : White
Euclidean Distance from input image: 2.9464166259765623
Amazon Url: www.amzon.com/dp/B01G89AKD8

```
Product Title:  chfashion womens button back stand collar spots chiffon blo
use white xlarge
Brand : Chfashion
Color : White
Euclidean Distance from input image: 2.962702445983887
Amazon Url: www.amzon.com/dp/B00UBU5Y5A
```

```
Product Title:  dkny womens small highsplithem button shirt white
Brand : DKNY
Color : White
Euclidean Distance from input image: 2.972809257507324
Amazon Url: www.amzon.com/dp/B073V3QWD5
```

```
Product Title:  generation love womens mona leopard print lace back top bla
ck small
Brand : Generation Love
Color : Black
Euclidean Distance from input image: 2.981822814941406
Amazon Url: www.amzon.com/dp/B075BDRD7R
```

Product Title:  azkara womens fashion tshirt casual vneck cross front tops unique tees shirts white large
Brand : Azkara
Color : White
Euclidean Distance from input image: 2.9848342895507813
Amazon Url: www.amzon.com/dp/B06WW8FVN9

Product Title:  san soleil jade cooling long sleeve mock neck upf 50 polo golf shirtme
Brand : San Soleil
Color : Palm Beach Paisley Bright
Euclidean Distance from input image: 2.9873234939575197
Amazon Url: www.amzon.com/dp/B06XKM62R1


Product Title:  official tangerinetane infinities art  small white tshirt women
Brand : Head Case Designs
Color : Some Infinities on White
Euclidean Distance from input image: 2.9974985122680664
Amazon Url: www.amzon.com/dp/B06XPH6KS6

Product Title:  roberta roller rabbit womens floral print wide sleeve tunic xsmall white multi
Brand : Roberta Roller Rabbit
Color : White Multi
Euclidean Distance from input image: 3.000602264404297
Amazon Url: www.amzon.com/dp/B01KW5ZAHU

Product Title:  exotic india yellow gray jamawar wrap fauxfur collar
Brand : Exotic India
Color : Gray
Euclidean Distance from input image: 3.0159403038024903
Amazon Url: www.amzon.com/dp/B073ZHRBV8

## Example 4

#### Image feature has the highest weight followed by Brand and Color having equal weights and Text Feture having the lowest weight

In [66]:
```
get_simil_items(12566, 5, 50,50,90, 20)
```

```
Queried Data point
--------------------------------------------------
Product Title:  burnt umber tiger tshirt zebra stripes xl  xxl
Brand : Si Row
Color : Brown
Amazon Url: www.amzon.com/dp/B00JXQB5FQ
```

```
--------------------------------------------------
```

```
Product Title:  foxcroft nyc womens pinpoint oxford shirt noniron stretch p
oplin blouse xlarge white
Brand : Foxcroft NYC
Color : White
Euclidean Distance from input image: 0.020497471246963894
Amazon Url: www.amzon.com/dp/B072277HVB
```

```
Product Title:  kiind longsleeve swing top white
Brand : KIIND OF
Color : White
Euclidean Distance from input image: 18.255204629945677
Amazon Url: www.amzon.com/dp/B0142Q3C6G
```

Product Title:  size comfortblend shirred crewneck longsleeve tshirt white 2x
Brand : Just My Size
Color : White
Euclidean Distance from input image: 18.801785167786104
Amazon Url: www.amzon.com/dp/B01N5YHV9R

Product Title:  karen scott womens plus short sleeve solid henley top white 1x
Brand : Karen Scott
Color : Bright White
Euclidean Distance from input image: 19.26604972261745
Amazon Url: www.amzon.com/dp/B0196H3Z2W

Product Title:  karen scott wmen short sleeve henley top plus size 0x bright white
Brand : Karen Scott
Color : Bright White
Euclidean Distance from input image: 19.320222848734183
Amazon Url: www.amzon.com/dp/B0196H7P5A

```
Product Title:  free people scoopneck highlow thermal tunic oatmeal heather
Brand : Free People
Color : Oatmeal Heather
Euclidean Distance from input image: 19.337894668382177
Amazon Url: www.amzon.com/dp/B00ZH2X45E
```

```
Product Title:  masion jules longsleeve paneled sweatshirt xxl grey
Brand : Maison Jules
Color : Grey
Euclidean Distance from input image: 19.506226446689702
Amazon Url: www.amzon.com/dp/B06X16WTFX
```

```
Product Title:  vitamina usa soft vneckline belted blouse st6356 wht
Brand : Vitamina USA
Color : White
Euclidean Distance from input image: 19.593636439396786
Amazon Url: www.amzon.com/dp/B0155ICSDS
```

```
Product Title:  dg2 diane gilman turtleneck fleece pullover pockets small g
ray
Brand : DG 2 Diane Gilman
Color : Grey
Euclidean Distance from input image: 19.842087561772733
Amazon Url: www.amzon.com/dp/B01MSP0KAM
```

```
Product Title:  jessica simpson womens frida peasant top antiquewhite x sma
ll
Brand : Fancy Jessica Simpson
Color : Antique/White
Euclidean Distance from input image: 19.941589460122593
Amazon Url: www.amzon.com/dp/B01KW297J0
```

```
Product Title:  red house  ladies nailhead noniron buttondown shirt3xl whit
e
Brand : Red House
Color : Red
Euclidean Distance from input image: 20.227423885046218
Amazon Url: www.amzon.com/dp/B008LOSLYO
```

```
Product Title:  american living womens dipdye peplum peplum top red l
Brand : American Home Living
Color : Red Multi
Euclidean Distance from input image: 20.326642041421287
Amazon Url: www.amzon.com/dp/B01LYFIRIL
```

```
Product Title:  jenni kayne womens crepe tee white small
Brand : Jenni Kayne
Color : White
Euclidean Distance from input image: 20.421741394624043
Amazon Url: www.amzon.com/dp/B07531DL6X
```

```
Product Title:  view walter baker long slv knit top a263076 white
Brand : Walter Baker
Color : White
Euclidean Distance from input image: 20.456099890487543
Amazon Url: www.amzon.com/dp/B074Q572HW
```

Product Title:  isaac mizrahi woven tunic multicolored soutache a223828 pin
k purple
Brand : Isaac Mizrahi Live!
Color : Pink Purple
Euclidean Distance from input image: 20.491487128330462
Amazon Url: www.amzon.com/dp/B074HFNCWK

Product Title:  bb dakota womens plus size tereza top optic white 2x
Brand : BB Dakota
Color : Optic White
Euclidean Distance from input image: 20.56340864816721
Amazon Url: www.amzon.com/dp/B00JG6EBEA

Product Title:  robert graham womens blouse trish lavender small
Brand : Robert Graham
Color : Lavender
Euclidean Distance from input image: 20.58540712698332
Amazon Url: www.amzon.com/dp/B071NQ52LM

Product Title:  jm collection womens bright white top size xl
Brand : JM Collection
Color : Bright White
Euclidean Distance from input image: 20.593637932991683
Amazon Url: www.amzon.com/dp/B01N4NYF25

```
Product Title:  hengsong women casual half sleeve sexy v lace splicing croc
het tops blouse shirts
Brand : HENGSONG
Color : White
Euclidean Distance from input image: 20.627490567031014
Amazon Url: www.amzon.com/dp/B01DXPBBLK
```

```
Product Title:  como vintage womens scoop neck bell sleeve blouse top sleev
e tie detail rose pink white  white xlarge
Brand : Como vintage
Color : White
Euclidean Distance from input image: 20.718525777859522
Amazon Url: www.amzon.com/dp/B074TWPJJP
```

## Example 5

#### Text feature has the highest weight, Brand has a lower weigth with color and Image having equal weight

In [70]:
```
get_simil_items(12566, 65, 25,5,5, 20)
```

```
Queried Data point
--------------------------------------------------
Product Title:  burnt umber tiger tshirt zebra stripes xl  xxl
Brand : Si Row
Color : Brown
Amazon Url: www.amzon.com/dp/B00JXQB5FQ
```

```
--------------------------------------------------
```

```
Product Title:  foxcroft nyc womens pinpoint oxford shirt noniron stretch p
oplin blouse xlarge white
Brand : Foxcroft NYC
Color : White
Euclidean Distance from input image: 0.004748771190643311
Amazon Url: www.amzon.com/dp/B072277HVB
```

```
Product Title:  kirkland signature ladies size medium cotton vneck tee whit
e
Brand : KS
Color : White
Euclidean Distance from input image: 5.137732772845218
Amazon Url: www.amzon.com/dp/B073ZRD5FL
```

```
Product Title:  exotic india yellow gray jamawar wrap fauxfur collar
Brand : Exotic India
Color : Gray
Euclidean Distance from input image: 5.878505554199219
Amazon Url: www.amzon.com/dp/B073ZHRBV8
```

```
Product Title:  miss chievous womens short sleeve knit top wsharkbite hem c
harcoal 1x
Brand : Miss Chievous
Color : Charcoal
Euclidean Distance from input image: 6.251144332903811
Amazon Url: www.amzon.com/dp/B06XR9NTVL
```

```
Product Title:  blouseuploter cute cartoon vest summer pajamas
Brand : UPLOTER
Color : White
Euclidean Distance from input image: 6.442845077532718
Amazon Url: www.amzon.com/dp/B01LPJRYI0
```

```
Product Title:  tahari asl ruffledfront polkadot top blackwhite xl
Brand : Tahari ASL
Color : Black
Euclidean Distance from input image: 6.5102986908139675
Amazon Url: www.amzon.com/dp/B074MZTZS5
```

```
Product Title:  kirkland signature ladies striped active yoga tee large gre
en
Brand : Kirkland Signature
Color : Green
Euclidean Distance from input image: 6.532643051165531
Amazon Url: www.amzon.com/dp/B01LXO8SGX
```

```
Product Title:  hanro womens leonie 34 sleeve shirt 76214 dim brown xsmall
Brand : HANRO
Color : Dim Brown
Euclidean Distance from input image: 7.046723410528452
Amazon Url: www.amzon.com/dp/B06Y3YRLP1
```

```
Product Title:  free people scoopneck highlow thermal tunic oatmeal heather
Brand : Free People
Color : Oatmeal Heather
Euclidean Distance from input image: 7.321495430258945
Amazon Url: www.amzon.com/dp/B00ZH2X45E
```

```
Product Title:  womens tops tees cute cactus graphic print size
Brand : namnoishop Crop Tops
Color : As Show
Euclidean Distance from input image: 7.399799124639781
Amazon Url: www.amzon.com/dp/B01M5H7LNO
```

```
Product Title:  shop flash slimming body shaping pro performance muscle wom
ens tank top white xxlarge
Brand : Shop Flash
Color : White
Euclidean Distance from input image: 7.421155242919922
Amazon Url: www.amzon.com/dp/B00TO30Z2W
```

```
Product Title:  fylo ladies size medium 14 zip vneck blouse pinkgrey animal
Brand : f
Color : Pink/Grey Animal
Euclidean Distance from input image: 7.455065536517093
Amazon Url: www.amzon.com/dp/B073X71H8K
```

```
Product Title:  modern dreamr womens slub pullover top blouse tank white la
rge
Brand : Dreamr
Color : White
Euclidean Distance from input image: 7.4621596527280305
Amazon Url: www.amzon.com/dp/B01BFERXHC
```

```
Product Title:  h bordeaux womens plus short sleeve hilow hem vneck shirt 1
x peony
Brand : H by Bordeaux Woman
Color : Pink
Euclidean Distance from input image: 7.469604415911624
Amazon Url: www.amzon.com/dp/B01I75WDQ8
```

```
Product Title:  elan white bell sleeve shoulder peasant blouse medium
Brand : Crazy4Bling
Color : White
Euclidean Distance from input image: 7.473423829440944
Amazon Url: www.amzon.com/dp/B00TA3TPN6
```

```
Product Title:  ramy brook womens poppy sleeveless top sz xs cognac 270064e
Brand : Ramy Brook
Color : Cognac
Euclidean Distance from input image: 7.474698097495272
Amazon Url: www.amzon.com/dp/B06XYYY256
```

```
Product Title:  focal18 paisley printed shirt mini dress blouse loose tops
long sleeve novelty
Brand : Focal18
Color : Multi
Euclidean Distance from input image: 7.489404823569491
Amazon Url: www.amzon.com/dp/B06W56V2N6
```

```
Product Title:  hippie rose juniors marled dolmansleeve tunic top size
Brand : Hippie Rose
Color : Russian Teal Combo
Euclidean Distance from input image: 7.490525741595218
Amazon Url: www.amzon.com/dp/B072HXX3W7
```

```
Product Title:  ira soleil womens ethnic viscose knit red  pink anarkali sh
rug kurta kurti large red
Brand : Ira Soleil
Color : Red
Euclidean Distance from input image: 7.495223318975718
Amazon Url: www.amzon.com/dp/B00P979OT2
```

```
Product Title:  jessica simpson womens frida peasant top antiquewhite x sma
ll
Brand : Fancy Jessica Simpson
Color : Antique/White
Euclidean Distance from input image: 7.496957938116343
Amazon Url: www.amzon.com/dp/B01KW297J0
```

## Example 6

**Text and Brand have equal weights whereas color and Image are excluded**

In [63]:
```
get_simil_items(12566, 5, 5,0,0, 20)
```

Queried Data point
---------------------------------------------------
Product Title:  burnt umber tiger tshirt zebra stripes xl  xxl
Brand : Si Row
Color : Brown
Amazon Url: www.amzon.com/dp/B00JXQB5FQ

---------------------------------------------------

Product Title:  foxcroft nyc womens pinpoint oxford shirt noniron stretch p
oplin blouse xlarge white
Brand : Foxcroft NYC
Color : White
Euclidean Distance from input image: 0.001953125
Amazon Url: www.amzon.com/dp/B072277HVB

Product Title:  kirkland signature ladies size medium cotton vneck tee whit
e
Brand : KS
Color : White
Euclidean Distance from input image: 2.0319442749023438
Amazon Url: www.amzon.com/dp/B073ZRD5FL

Product Title:  exotic india yellow gray jamawar wrap fauxfur collar
Brand : Exotic India
Color : Gray
Euclidean Distance from input image: 2.385471153259277
Amazon Url: www.amzon.com/dp/B073ZHRBV8

Product Title:  miss chievous womens short sleeve knit top wsharkbite hem c
harcoal 1x
Brand : Miss Chievous
Color : Charcoal
Euclidean Distance from input image: 2.6800804138183594
Amazon Url: www.amzon.com/dp/B06XR9NTVL

Product Title:  blouseuploter cute cartoon vest summer pajamas
Brand : UPLOTER
Color : White
Euclidean Distance from input image: 2.8447616577148436
Amazon Url: www.amzon.com/dp/B01LPJRYI0

```
Product Title:  kirkland signature ladies striped active yoga tee large gre
en
Brand : Kirkland Signature
Color : Green
Euclidean Distance from input image: 2.8465106964111326
Amazon Url: www.amzon.com/dp/B01LXO8SGX
```

```
Product Title:  tahari asl ruffledfront polkadot top blackwhite xl
Brand : Tahari ASL
Color : Black
Euclidean Distance from input image: 2.9467212677001955
Amazon Url: www.amzon.com/dp/B074MZTZS5
```

```
Product Title:  fylo ladies size medium 14 zip vneck blouse pinkgrey animal
Brand : f
Color : Pink/Grey Animal
Euclidean Distance from input image: 3.421704864501953
Amazon Url: www.amzon.com/dp/B073X71H8K
```

Product Title:  hippie rose juniors marled dolmansleeve tunic top size
Brand : Hippie Rose
Color : Russian Teal Combo
Euclidean Distance from input image: 3.4967933654785157
Amazon Url: www.amzon.com/dp/B072HXX3W7


Product Title:  divas love snow henley long sleeve tshirt 4xlarge violettur
quoise
Brand : Divas SnowGear
Color : Violet/Turquoise
Euclidean Distance from input image: 3.57947998046875
Amazon Url: www.amzon.com/dp/B071NHMRB2


Product Title:  womans wicked musical poster shirts red xxl
Brand : vcord
Color : Red
Euclidean Distance from input image: 3.6908035278320312
Amazon Url: www.amzon.com/dp/B01KUN4VWE

Product Title:  material girl womens juniors cutout swing shift caviar blac
k combo xxl
Brand : Material Girl
Color : Caviar Black Combo
Euclidean Distance from input image: 3.822268295288858
Amazon Url: www.amzon.com/dp/B072L5CGR3

Product Title:  boden womens embellished safari spot top us sz 4 brownblack white
Brand : BODEN
Color : Brown/Black/White
Euclidean Distance from input image: 3.932520345495864
Amazon Url: www.amzon.com/dp/B01H62TIZ2

Product Title:  hanro womens leonie 34 sleeve shirt 76214 dim brown xsmall
Brand : HANRO
Color : Dim Brown
Euclidean Distance from input image: 3.9943787134157858
Amazon Url: www.amzon.com/dp/B06Y3YRLP1

Product Title:  socialite teal women medium crochet offshoulder blouse blue
Brand : Socialite
Color : Blue
Euclidean Distance from input image: 3.9959529876708983
Amazon Url: www.amzon.com/dp/B01M0DVT93

Product Title:  romeo  juliet navy womens medium jeweled knit top green
Brand : Romeo & Juliet Couture
Color : Green
Euclidean Distance from input image: 4.00236701965332
Amazon Url: www.amzon.com/dp/B071P49MCK

Product Title:  ideology womens small crewneck solid tee tshirt black
Brand : Ideology
Color : Black
Euclidean Distance from input image: 4.005290603637695
Amazon Url: www.amzon.com/dp/B0721T22DL

Product Title:  cavalli womens sparkle loose style blouse top us 40
Brand : Just Cavalli
Color : Sparkle Gray / Black
Euclidean Distance from input image: 4.062127546118423
Amazon Url: www.amzon.com/dp/B01CAIM9OE

Product Title:  ira soleil womens ethnic viscose knit red  pink anarkali shrug kurta kurti large red
Brand : Ira Soleil
Color : Red
Euclidean Distance from input image: 4.07347588901393
Amazon Url: www.amzon.com/dp/B00P979OT2

Product Title:  style  co plus size shortsleeve flouncehem top 2x pink
Brand : Style & Co.
Color : Red
Euclidean Distance from input image: 4.091505674170181
Amazon Url: www.amzon.com/dp/B01N6U2KIT