Recommendation System on HKTVmall Based on Image-based processing and Natural Language Processing

Group 7
Yuen Ho Man
Chan Chak Yan Esmond

Abstract : This project aims to give an algorithm for the recommendation system of HKTVmall online shopping network. We will focus on two aspects: Image-based processing of different products' photos listed on shop and Natural Language Processing of products' descriptions and names. We are aimed at providing top 10 personalized recommendations in improving customer experience.

1. Introduction

1.1 Background

Online shopping has been a skyrocketing trend these years. It has developed quickly and diversified, while HKTVmall is a great example demonstrating how online shopping has developed these days. According to YouGov's ranking(2022), HKTVMall became Hong Kong's most talked about brand in 2021. Ranked above Octopus Card and ViuTV. With $22 million daily average gross merchandise volume, more than 1450000 rapid users and 4900000 apps download volume per month.

1.2 Motivation

With the impact of COVID-19, HKTVmall has become one of the popular online shopping sites in a short time. As people are not willing to get out and risk the chances of getting sick, online shopping becomes their first choice. There are different kinds of categories offered on HKTVmall, such as personal care, supermarket and housewares etc. People can order all kinds of things through their mobile phones. HKTVmall then will prepare their ordered products, pack them into boxes and directly deliver them to their apartment or the store chosen by the customer to wait for pick up. As it has a variety of products for customers, they may find it difficult to look for products that really suit them. Taking hair shampoo as an example, there may be a lot of selling points for different products. Some may provide a good smell, claim to have better cleaning effects or have a larger capacity. Different customers may be concerned about different things, so a recommendation system is therefore important, it helps suggest them goods based on their browsing history.

In this situation, attracting customers to order becomes one of the most important things an E-Commerce company should do. One of the methods is to provide a personalized browsing experience for each user. HKTVmall has a block named "FEATURED FOR YOU" based on each category, yet it is actually not based on customers' preference but a paid advertisement provided by clients. It may not be what customers are looking for or suitable for them, it thus may not have a great impact in boosting the sales. As customers are easily attracted by products' shape and description, we therefore will be focusing on designing an algorithm with Image-based processing and Natural Language Processing to provide top 10 personalized recommendations in improving customer experience in HKTVmall.

1.3 Data Extraction

With the help of HKTVmall api, we can obtain various data, including product title, description, review, image for further analysis.

name_chi	brand_name_chi	summary_chi	description_chi	image_urls
✿Breezy智能換氣	Petkit	<p>有三款顏	<p style="vertical	https://cdn-mms.hktv
✿Breezy智能換氣	Petkit	<p>有三款顏	<p style="vertical	https://cdn-mms.hktv
✿Breezy智能換氣	Petkit	<p>有三款顏	<p style="vertical	https://cdn-mms.hktv
✿Breezy智能換氣	Petkit	<p>有三款顏	<p style="vertical	https://cdn-mms.hktv
(Hello Kitty/背囊)	SANRIO	日本直送 	尺寸: 背囊	https://images.hktv-i
(My Melody/背囊	SANRIO	日本直送 	尺寸: 背囊	https://images.hktv-i
Viola Tatami 午夜		<p>為追求功	<p>Viola Tatami	https://cdn-mms.hktv
Viola Tatami 玫瑰		<p><b id="doc	<p><b id="docs-i	https://cdn-mms.hktv
小王子限定版迷你	FX Creations	<div style="for		https://cdn-mms.hktv
[小王子限定版 x I	FX Creations	<p class="ec-ir		https://cdn-mms.hktv

2. Method

We have selected two models in generating a 10-items recommendation list. The models are Image-based Processing Model and Natural Language Processing Model. Both of them will be covered below. Based on different types of users, we will use a distinct approach to determine their target products for further analysis. For users who have not completed any transaction, we will select the highest visit-frequency item as target. For users who have completed a transaction, we will select the recently purchased product as target.

2.1 Model 1: Image-based Processing Model

For Image-based Processing Model, we will focus on the image similarity on different products within the same category. As stated above, a target product is selected based on their attribute. We will obtain the product's main image as the target image. While considering the excessive amount of product. The designed product comparing algorithm is below:

1. Remove purchased item from compare list
2.Randomly select x% of the total product for comparison.
(The value of x will be determined in model evaluation)
3.Calculate the image similarity of target and selected image.
4.Take the best image and insert it into the recommendation list.
5.Restart from step 2 if the product is inserted.

According to the algorithm, we have chosen 5 Similarity Index and each will be contributing a

product to the recommendation list. The chosen Similarity Indexes is below:

Similarity Index of Image-based Processing Model

1.Structural Similarity based on scikit-image (SSIM)

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Where, μ = the pixel sample mean
σ^2 = the variance
σ_{xy} = the covariance of x and y
$c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$ = two variables to stabilize the division, with L = the dynamic range of the pixel-values
k_1 = 0.01
k_2 = 0.03

2.Cosine similarity

$$\cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Where, A and B are non-zero vector

3.Bray-Curtis Distance

$$\sum |u_i - v_i| / \sum |u_i + v_i|$$

Where, u and v are nonzero vector

4.Euclidean Distance

$$d = |\mathbf{x} - \mathbf{y}| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}\,.$$

Where, x and y are two vectors

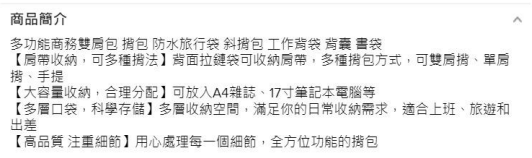
5.Jensen-Shannon Distance

$$\sqrt{\frac{D(p \parallel m) + D(q \parallel m)}{2}}$$

Where, **p** and **q** are two vectors
m is the pointwise mean of two vector
D is the kullback-leibler divergence

2.2 Model 2: Natural Language Processing Model

For Natural Language Processing Model, we will focus on the text similarity on different products. We will extract the text feature given by product name, description and review. Only Chinese text will be considered as major customers are from Hong Kong.


商品簡介

多功能商務雙層包 揹包 防水旅行袋 斜揹包 工作背袋 背囊 書袋
【層帶收納，可多種揹法】背面拉鏈袋可收納層帶，多種揹包方式，可雙層揹、單層揹、手提
【大容量收納，合理分配】可放入A4雜誌、17寸筆記本電腦等
【多層口袋，科學存儲】多層收納空間，滿足你的日常收納需求，適合上班、旅遊和出差
【高品質 注重細節】用心處理每一個細節，全方位功能的揹包

Figure 1. Example of Product Description

In the feature extracting process, we choose 'jieba', a python library for performing word splitting to Chinese text. The cut_all mode is off for better accuracy.

```
import re
import jieba
result=[]
text=re.sub(r'[^\w]', ' ', t).replace(' ','')
result.append(' '.join(jieba.lcut(text,cut_all=False)))
print(result)
```

['多功能 商務 雙層 包 揹 包 防水 旅行袋 斜 揹 包 工作 背袋 背囊 書袋 層帶 收納 可 多種 揹 法 背面 拉 鏈袋 可 收納 層帶 多 種 揹 包 方式 可雙 層 揹 單 層 揹 手提 大容量 收納 合理 分配 可 放入 A4 雜誌 17 寸筆 記本 電腦 等 多層 口袋 科學存 儲多層 收納 空間 滿足 你 的 日常 收納 需求 適合 上班 旅遊 和 出差 高品質 注重 細節 用心 處理 每 一個 細節 全方位 功能 的 揹 包']

Figure 2. Example of Product Description after Word Splitting with Jieba

From the example above, the word splitting process can divide words with high accuracy. With the results shown above, the designed product comparing algorithm is below:

1. Remove purchased item from compare list
2.Randomly select x% of the total product for comparison.
(The value of x will be determined in model evaluation)
3.Calculate the image similarity of target and selected image.
4.Take the best image and insert it into the recommendation list.
5.Restart from step 2 if the product is inserted.

According to the algorithm, We selected two methods in determining the similarity between two texts. Because of the computational time, TF-IDF will contribute 3 items to the list, while Word2vec contributes 2 items. The methods detail is below:

Method of Natural Language Processing Model

1.Term Frequency–Inverse Document Frequency (TF-IDF)

$$tfidf(t,d,D) = tf(t,d) * idf(t,D)$$

Where,
TF (Term Frequency)

$$tf_{t,d} = n_{t,d} / \sum_k n_{t,d}$$

Where,
IDF (Inverse Document Frequency)

$$idf(t,D) = log\frac{N}{|\{d \in D : t \in d\}|}$$

2.Word2vec

The method of Word2vec will be used in performing word embeddings learning with neural networks. It can change words to vectors in predicting the probability that words will appear.

In the Word2vec algorithm, there are two models including Skip Gram and Common Bag Of Words (CBOW). According to T.Mikolov (2013), CBOW

is able to run faster than Skip Gram and has a better representation of more frequent words. Referring to the excessive amount of data in HKTVmall, we decided to use CBOW as the method under Word2vec.

2.3 Algorithm Formation

The algorithm is designed based on a common shopping habit. For the randomness part in both algorithms. According to L.McAlister (1982), there is a customer behavior named as variety seeking behavior. For customers with this behavior, they will seek for other products different from the one they bought. The reason behind this behavior does not mean they are not satisfied with the previous product, it is that they prefer to seek variety. However, refer to C.Wang (2010), there still a large number of customers under Behavioral brand loyalty which tend to buy the products in the same brand.

Therefore, we want to simulate customers behavior based on randomness, where the customers are usually choosing random products different from the product they bought, products which are in the same brand or products based on their preference which may be varied from time to time.

For both algorithms, the value of x in the randomness part will be evaluated in the next section.

3. Result

With the two models above, we evaluate the model in optimizing between the computational time and accuracy. The Index of determining the accuracy is the lowest mean-square error (MSE). For the Evaluation part below, we select all products under the category of All products<Women's Fashion<Bags & Wallets<Backpacks as demonstration. In the category, there are a total of 2367 items.

For the following conditions, it is given that a customer bought THE LITTLE PRINCE SPECIAL EDITION BACKPACK.

3.1 Model Evaluation : Image-based Processing Model

To evaluate the performance of an Image-based Processing Model, computational time is an important concern. In the category of women's backpacks, there are a total of 2367 items. For the computational time, we will test all the integer values from 0 to 100 of x.

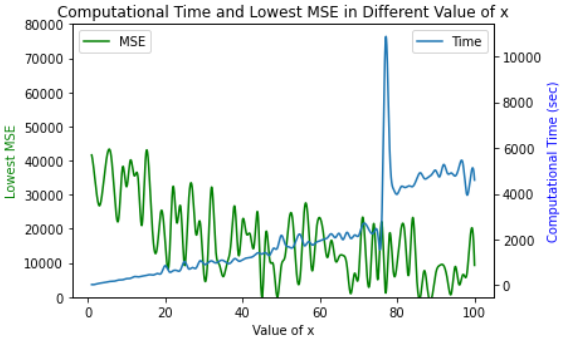


Figure 3. Computational Time and Lowest MSE in Different Value of x

From Figure 1, it illustrates the computational time to different values of x. With calculating the correlation between the time and lowest MSE:

```
PearsonRResult(statistic=-0.5634493282448062,
pvalue=1.0405885016074257e-09)
```

We can clearly see that there is a negative relationship, where the higher computational time refers to lower MSE. The P-value is statistically significant.

During the evaluation on the value of x, the lowest MSE decreases when the value of x is around 25. Therefore, for the Image-based Processing Model, we decided the range of x should be between 25-50% for better accuracy and lower computational time.

3.2 Results : Image-based Processing Model

We set x=30% for the example shown below. As for the result of the algorithm, it will generate a table with similarity index, index value, product ID, product name, brand and image url.

Similarity Index	Value	Product ID	Product Name	Brand	Image URL
SSIM	0.66820	H7847001_S	Tait Mini 時尚休閒日常兩用小背囊	Moral	https://images.hktv-
Euclidean Distance	47664.59375	H8450001_S	男士皮質時尚潮流背包(黑色)#N12	10 Acres	https://images.hktv-
Cosine Similarity	0.99939	H9458001_S	Pick&Pack Tractor Backpack S -Blue	Pick & Pack	https://cdn-mms.hkt
Bray-Curtis Distance	0.08677	H8450001_S	帆布斜挎包 灰色（長寬高：22*13*	十歇地	https://images.hktv-
Jensen-Shannon Distance	0.12370	H8450001_S	時尚商務雙肩包(迷彩三 寬28*高40	十歇地	https://images.hktv-

Figure 4. Table generated from image-based processing model

The following figures are the products from Figure 2 by order.

Figure 4.1

Figure 4.2

Figure 4.3

Figure 4.4

Figure 4.5

From the figures suggested, it is shown that they have a similar shape with the purchased product except Figure 4.4. It can be explained by randomness that there is a chance the list may include unrelated products.

3.3 Model Evaluation : Natural Language Processing Model

Same as Image-based Processing, our target is to get a balance between computational time and the accuracy (measured by the lowest MSE). For the computational time, we will test all the integer values from 0 to 100 of x.

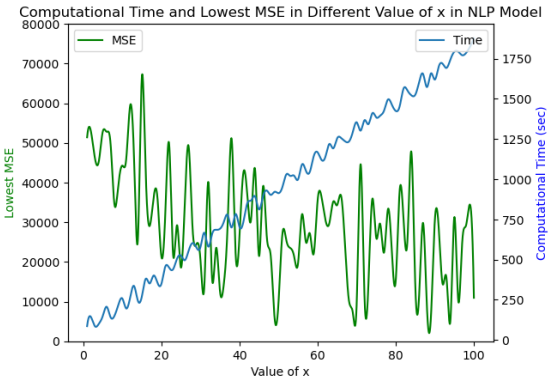


Figure 5. Computational Time and Lowest MSE in Different Value of x in NLP Model

From Figure 2, it illustrates the computational time to different values of x. With calculating the correlation between the time and lowest MSE:

```
PearsonRResult(statistic=-0.6802745774047039,
pvalue=7.003945436918876e-15)
```

We can clearly see that there is a negative relationship, which is the same as the Image-based Processing Model. The P-value is also statistically significant. For the value of x, the lowest MSE decreases when the value of x is around 30. Therefore, for the Natural Language Processing Model, we decided the range of x should be between 30-55% for better accuracy and lower computational time.

3.4 Results : Natural Language Processing Model

We set x=50% for the example shown below. As for the result of the algorithm, it will generate a table with similarity index, product ID, product name, summary and review. The following figures are the products from Figure 3 by order.

Similarity Index	Product ID	Product Name	Summary	Review
TF-IDF	H0147001_S_LPP	[小王子限定版 x FX Creations聯乘系	<p>[FX Creations	比較細吃啲！;Good
TF-IDF	H0116003_S_ATC	FX Creations - 小王子限定版迷你背	<p><span style="	外觀漂亮，質料好
TF-IDF	H0116003_S_SS-S	SOU SOU 海外限定聯名系列 迷你多	<p>anello x SOU	good quality.
Word2vec	H8881001_S_Bac	(藍色) 16L雙層防水背囊,戶外防水袋	<p>16L雙層防水	細細地，做運動最
Word2vec	H0116003_S_JSO	Jansport 迷你 書包 背囊 後背包 熱賣	<p dir="ltr"><spa	love the bag

Figure 6. Table generated from natural-language processing model

The following figures are the products from Figure 2 by order.

Figure 6.1

Figure 6.2

Figure 6.3

Figure 6.4

Figure 6.5

From the above figures, Figures 6.1 and 6.2 have the same brand as the purchased one. While Figure 6.3 is also a special edition and Figure 6.4 is also blue, these features are mentioned in the product name of Figure 6. As for Figure 6.5, it is an unrelated product like Figure 4.4 above.

4. Discussion

The recommendation list can show some important features of the target image. With the Image-based Processing Model, the list can suggest the product based on the shape. With the Natural Language Processing Model, the list obtains the product based on the unique selling proposition. However, for both models, there are some unrelated products like Figure 4.4 from image-based processing model and Figure 6.5 from natural-language processing model. It is expected as the algorithm is based on randomness.

4.1 Limitation

For limitation, the amount of products is way too big that it may cost too much time to run the process. We try to tackle it by extracting x% of the total amount of products in the category and run one of the methods. However, it still requires a long time to process.

Another limit is that during the image processing, they will be changed from BGR to grayscale. So we will not be able to consider the color difference as a reference.

4.2 Recommendations for Future Research

We are not able to figure out patterns of customers' habits based on the data obtained, which affects the accuracy a lot. Therefore, the suggestions may not match their preference. It is recommended to perform predictions on customers' purchasing habits into consideration in future research.

4.3 Conclusion

Recommendation system is an important component as it should be able to predict the desire of a user and suggest products matching the user's preference. Based on image-based processing and natural language processing, we hope the system can help customers to find suitable products and stimulate sales. We believe a good recommendation system will be able to help both customers and HKTVmall.

5. Appendix

5.1 Code example for Image-based Processing Model

```
#Data Cleaning
img_df=df['image_urls']
img_list=img_df.tolist()
img_list_clean=[]
img_list_clean1=[]
for mm in img_list:
    try:
        mm1=mm.split(' | ')[0]
        img_list_clean.append(mm1)
    except:
        img_list_clean.append(mm)
for oo in img_list_clean:
    try:
        oo1=oo.split('//')[1]
        oo2='https://'+str(oo1)
        img_list_clean1.append(oo2)
    except:
        img_list_clean1.append(oo)
img_df_clean=pd.DataFrame(img_list_clean1)
df_img=df[['sku_id','name_chi','brand_name_chi']]
df_img['df_img']=img_df_clean
df_img
```

Image Final (SSIM)

```
In [11]: from skimage import io
         import numpy as np
         from PIL import Image
         import requests
         from io import BytesIO
         import warnings
         import matplotlib.pyplot as plt
         from skimage import data, img_as_float
         from skimage.metrics import structural_similarity as ssim
         from skimage.metrics import mean_squared_error
         import time
         import random
         num_list=[]
         compare_list=[]
         compare_list_name=[]
         compare_list_id=[]
         for l in range(50):
             nn=random.randint(0,2115)
             num_list.append(nn)
         for u in num_list:
             pp=df_img['df_img'][u]
             compare_list.append(pp)
             yy=df_img['name_chi'][u]
             compare_list_name.append(yy)
             tt=df_img['sku_id'][u]
             compare_list_id.append(tt)
         target='https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wkJLCKrWjQ20220826125727_1200.jpg'
         ssim1=[]
         mse=[]
         for link in compare_list:
             compare=link
             response = requests.get(target)
             t_img = Image.open(BytesIO(response.content))
             response2 = requests.get(compare)
             c_img = Image.open(BytesIO(response2.content))
             if c_img.size>t_img.size:
                 cimg = c_img.resize(t_img.size)
                 c_img_np = img_as_float(cimg)
                 t_img_np = io.imread(target)
             if c_img.size<t_img.size:
                 timg = t_img.resize(c_img.size)
                 t_img_np = img_as_float(timg)
                 c_img_np = io.imread(compare)
             if c_img.size==t_img.size:
                 c_img_np = io.imread(compare)
                 t_img_np = io.imread(target)
             ssim_none = ssim(c_img_np, t_img_np,multichannel=True)
             mse_none = mean_squared_error(c_img_np, t_img_np)
             ssim1.append(ssim_none)
             mse.append(mse_none)
         number_df=pd.DataFrame(num_list)
         compare_df=pd.DataFrame(compare_list)
         compare_id_df=pd.DataFrame(compare_list_id)
         compare_name_df=pd.DataFrame(compare_list_name)
         ssim_df=pd.DataFrame(ssim1)
         mse_df=pd.DataFrame(mse)
         final = pd.concat([number_df, compare_id_df,compare_name_df,compare_df,ssim_df,mse_df], axis=1)
         final.columns =['Number','Compare Item ID','Compare Item Link','Compare Item Img',"SSIM","MSE"]
         final.to_excel('img_final_df.xlsx')
         final
```

Final for Euclidean Distance with random loop

```
[ ]: import numpy as np
     import urllib
     import cv2
     from scipy import spatial
     from scipy.spatial import distance
     def url_to_image(url):
         resp = urllib.request.urlopen(url)
         image = np.asarray(bytearray(resp.read()), dtype="uint8")
         image = cv2.imdecode(image, cv2.IMREAD_COLOR)
         return image
     target='https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wkJLCKrWjQ20220826125727_1200.jpg'
     test1=url_to_image('https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wkJLCKrWjQ20220826125727_1200.jpg')
     gray_image = cv2.cvtColor(test1, cv2.COLOR_BGR2GRAY)
     histogram = cv2.calcHist([gray_image], [0],None, [256], [0, 256])
     ed=[]
     num_list=[]
     compare_list=[]
     compare_list_name=[]
     compare_list_id=[]
     for l in range(50):
         nn=random.randint(0,2115)
         num_list.append(nn)
     for u in num_list:
         pp=df_img['df_img'][u]
         compare_list.append(pp)
         yy=df_img['name_chi'][u]
         compare_list_name.append(yy)
         tt=df_img['sku_id'][u]
         compare_list_id.append(tt)
     for link in compare_list:
         test2 = url_to_image(link)
         gray_image1 = cv2.cvtColor(test2, cv2.COLOR_BGR2GRAY)
         histogram1 = cv2.calcHist([gray_image1], [0],None, [256], [0, 256])
         c1, c2 = 0, 0
         i = 0
         while i<len(histogram) and i<len(histogram1):
             c1+=(histogram[i]-histogram1[i])**2
             i+= 1
         c1 = c1**(1 / 2)
         ed.append(c1)
```

Final for Cosine Similarity with random loop

```
]: import numpy as np
   import urllib
   import cv2
   from scipy import spatial
   from scipy.spatial import distance
   def url_to_image(url):
       resp = urllib.request.urlopen(url)
       image = np.asarray(bytearray(resp.read()), dtype="uint8")
       image = cv2.imdecode(image, cv2.IMREAD_COLOR)
       return image
   target='https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wkJLCKrWjQ20220826125727_1200.jpg'
   test1=url_to_image('https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wkJLCKrWjQ20220826125727_1200.jpg')
   gray_image = cv2.cvtColor(test1, cv2.COLOR_BGR2GRAY)
   histogram = cv2.calcHist([gray_image], [0],None, [256], [0, 256])
   cos=[]
   num_list=[]
   compare_list=[]
   compare_list_name=[]
   compare_list_id=[]
   for l in range(50):
       nn=random.randint(0,2115)
       num_list.append(nn)
   for u in num_list:
       pp=df_img['df_img'][u]
       compare_list.append(pp)
       yy=df_img['name_chi'][u]
       compare_list_name.append(yy)
       tt=df_img['sku_id'][u]
       compare_list_id.append(tt)
   for link in compare_list:
       test2 = url_to_image(link)
       gray_image1 = cv2.cvtColor(test2, cv2.COLOR_BGR2GRAY)
       histogram1 = cv2.calcHist([gray_image1], [0],None, [256], [0, 256])
       cos_similarity = -1 * (spatial.distance.cosine(histogram, histogram1) - 1)
       cos.append(cos_similarity)
```

Final for Bray-Curtis distance with random loop

```
import numpy as np
import urllib
import cv2
from scipy import spatial
from scipy.spatial import distance
def url_to_image(url):
    resp = urllib.request.urlopen(url)
    image = np.asarray(bytearray(resp.read()), dtype="uint8")
    image = cv2.imdecode(image, cv2.IMREAD_COLOR)
    return image
target='https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wk3LCKrWjQ20220826125727_1200.jpg'
test1=url_to_image('https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wk3LCKrWjQ20220826125727_1200.jpg')
gray_image = cv2.cvtColor(test1, cv2.COLOR_BGR2GRAY)
histogram = cv2.calcHist([gray_image], [0],None, [256], [0, 256])
bc=[]
num_list=[]
compare_list=[]
compare_list_name=[]
compare_list_id=[]
for l in range(50):
    nn=random.randint(0,2115)
    num_list.append(nn)
for u in num_list:
    pp=df_img['df_img'][u]
    compare_list.append(pp)
    yy=df_img['name_chi'][u]
    compare_list_name.append(yy)
    tt=df_img['sku_id'][u]
    compare_list_id.append(tt)
for link in compare_list:
    test2 = url_to_image(link)
    gray_image1 = cv2.cvtColor(test2, cv2.COLOR_BGR2GRAY)
    histogram1 = cv2.calcHist([gray_image1], [0],None, [256], [0, 256])
    bc1=distance.braycurtis(histogram, histogram1)
    bc.append(bc1)
```

Final for Jensen-Shannon distance with random loop

```
import numpy as np
import urllib
import cv2
from scipy import spatial
from scipy.spatial import distance
def url_to_image(url):
    resp = urllib.request.urlopen(url)
    image = np.asarray(bytearray(resp.read()), dtype="uint8")
    image = cv2.imdecode(image, cv2.IMREAD_COLOR)
    return image
target='https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wk3LCKrWjQ20220826125727_1200.jpg'
test1=url_to_image('https://cdn-mms.hktvmall.com/hktv/mms/uploadProductImage/79a2/51e8/268a/wk3LCKrWjQ20220826125727_1200.jpg')
gray_image = cv2.cvtColor(test1, cv2.COLOR_BGR2GRAY)
histogram = cv2.calcHist([gray_image], [0],None, [256], [0, 256])
js=[]
num_list=[]
compare_list=[]
compare_list_name=[]
compare_list_id=[]
for l in range(50):
    nn=random.randint(0,2115)
    num_list.append(nn)
for u in num_list:
    pp=df_img['df_img'][u]
    compare_list.append(pp)
    yy=df_img['name_chi'][u]
    compare_list_name.append(yy)
    tt=df_img['sku_id'][u]
    compare_list_id.append(tt)
for link in compare_list:
    test2 = url_to_image(link)
    gray_image1 = cv2.cvtColor(test2, cv2.COLOR_BGR2GRAY)
    histogram1 = cv2.calcHist([gray_image1], [0],None, [256], [0, 256])
    js1=distance.jensenshannon(histogram, histogram1)
    js.append(js1)
```

5.2 Code example for Natural Language Processing Model

```
#data cleaning and jieba
import re
sum_chi_list_clean=[]
sum_chi_list_clean1=[]
sum_chi_list_sep=[]
sum_chi_list=df['summary_chi'].to_list()
for x in sum_chi_list:
    t1=BeautifulSoup(str(x), 'html.parser')
    sum_chi_list_clean.append(t1.text)
for y in sum_chi_list_clean:
    y1=re.sub(r'[^\w]', ' ', y)
    y1=y1.replace(" ","")
    sum_chi_list_clean1.append(y1)
for z in sum_chi_list_clean1:
    sum_chi_list_sep.append(' '.join(jieba.lcut(z, cut_all=False, HMM=False)))
```

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\ivany\AppData\Local\Temp\jieba.cache
Loading model cost 1.015 seconds.
Prefix dict has been built successfully.
```

```
import jieba
import jieba.analyse
joined = " ".join(sum_chi_list_sep)
tags = jieba.analyse.extract_tags(joined, topK=20, withWeight=True)
df_tag=pd.DataFrame(tags)
df_tag.columns =['Terms','TF-IDF']
df_tag
df_tag.to_excel('tfidf_test_sum.xlsx')
```

```
#CBOW fit
text = ' '.join(text[:100])
model  = cbow(text=text)
model.fit()

final_vectors = {}
for k, v in model.word_index.items():
    final_vectors[k] = (model.U[v, :] + model.V[:, v])/2

print(final_vectors)
```

6. Reference

1. Baek, J. H. (n.d.). Amazon Recomzmender System. UC San Diego Library | Digital Collections. Retrieved October 16, 2022, from https://library.ucsd.edu/dc/object/bb8503744c
2. 一灯架构. (n.d.). Gensim word2vec 使用教程. 知乎专栏. Retrieved October 16, 2022, from https://zhuanlan.zhihu.com/p/28943718
3. Understanding TF-ID: A simple introduction. MonkeyLearn Blog. (2019, May 10). Retrieved October 16, 2022, from https://monkeylearn.com/blog/what-is-tf-idf/
4. Alake, R. (2021, November 3). Understanding cosine similarity and its application. Medium. Retrieved October 16, 2022, from https://towardsdatascience.com/understanding-cosine-similarity-and-its-application-fd42f585296a
5. Building a Recommender System Using Amazon Reviews. (2018, October 15). Retrieved November 3, 2022, from https://github.com/smwitkowski/Amazon-Recommender-System/blob/master/Collaborative%20Filtering%20Amazon%20Watch%20Reviews.ipynb
6. Ben Chamblee. (2022, February 7). What is Cosine Similarity? How to Compare Text and Images in Python. Retrieved November 3, 2022 from https://towardsdatascience.com/what-is-cosine-similarity-how-to-compare-text-and-images-in-python-d2bb6e411ef0
7. scipy.spatial.distance.cdist. (n.d.). Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html
8. Adrian Rosebrock. (2015, March 2). Convert URL to image with Python and OpenCV. Retrieved November 3, 2022, from https://pyimagesearch.com/2015/03/02/convert-url-to-image-with-python-and-opencv/
9. scipy.spatial.distance.jensenshannon. (n.d.). Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jensenshannon.html
10. scipy.spatial.distance.braycurtis. (n.d.). Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.braycurtis.html
11. Distributed representations of words and phrases and their compositionality. (n.d.).

Retrieved December 1, 2022, from https://arxiv.org/pdf/1310.4546.pdf

12. Karani, D. (2020, September 2). Introduction to word embedding and word2vec. Medium. Retrieved December 1, 2022, from https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa