National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Sureshkumar Durairaj
Student ID: x21178933

School of Computing
National College of Ireland

Supervisor: Qurrat Ul Ain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sureshkumar Durairaj
Student ID:	x21178933
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Qurrat Ul Ain
Submission Due Date:	14/08/2023
Project Title:	Configuration Manual
Word Count:	2892
Page Count:	57

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th August 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sureshkumar Durairaj
x21178933

1 Introduction

The configuration manual illustrates the sequential step-by-step guide to execute the modules associated in the research project and the steps to evaluate the same. The steps provided comprising of various requisites starting from software installation to model building process. This project comprises of two different stages such as identify the product similarity using image data sets using Image based CNN algorithm and concatenating the same with text-based vectorisation methods for matching the corresponding product label description. The individual code snippets to perform the same is provided in the upcoming sections.

2 System Configuration

2.1 Software Requirements

The research project was developed using the open-source IDE called Jupyter Notebook as well as using Google colab an open-source framework for AI/ML projects in google ecosystem. This environment works based on python module. All these packages need to be installed before building the project.

2.2 Hardware specifications

- System Name: DESKTOP-SM51BMP
- Processor: Intel(R) Core (TM) i7-6500U CPU @ 2.50GHz, 2601 Mhz, 2 Core(s), 4 Logical Processor(s)
- Installed RAM: 16.00 GB
- Storage Size: 1TB SSD (109,951,162,7776 bytes)
- OS type: 64-bit operating system, x64-based processor

3 Installation and Environment Setup

- **Python**

Python module was used in this project. Since, it has many in-build libraries which support most of the Deep Learning and Machine Learning Projects. It ease the model building and analyse with various plots. The first requirement is to install the latest version python in the system. Based on the operating system, the package installer can be downloaded from the website [1] through browser. After successful installation of python from the website as shown below figure 1, type 'python -version' in the command prompt to verify it.

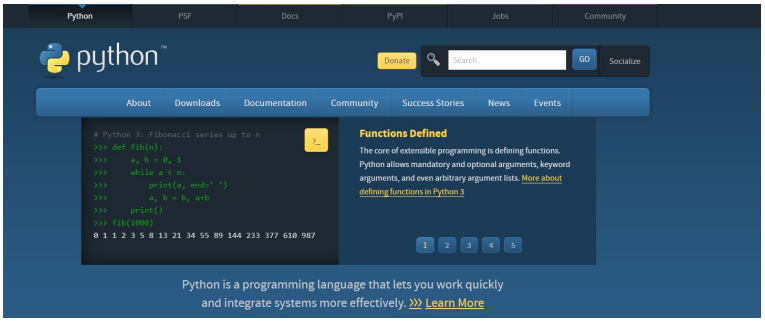


Figure 1: Python Official page

- **Anaconda**

Anaconda package comprises of several IDE which will be useful for developing the code and for analysing the outputs through the python package. This package can be downloaded and installed from the website [2] as shown in the Figure 2 . From the anaconda navigator, Jupyter notebook and it's tasks are launched in the browser tabs. Initially python notebook is created and saved as .ipynb format.

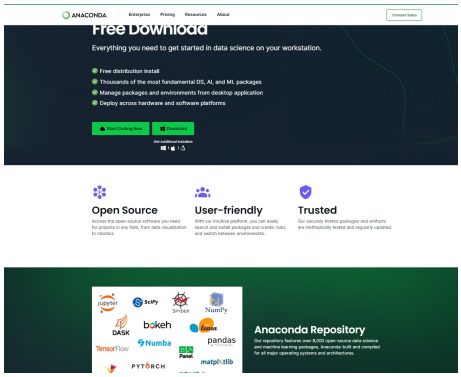


Figure 2: Anaconda Downloads Page

- **Jupyter Notebook**

The python libraries are installed during the implementation of code using pip command. The required libraries for this project are numpy, pandas, tensorflow, matplotlib, seaborn and plotly. There were lot of different IDE available in this navigator. In this project, Jupyter Notebook is used for building the model.

Command: pip install 'LibraryName'

[1] https://www.python.org/downloads/
[2] https://www.anaconda.com/products/individual

4 Data Collection

There were two datasets used for this project which is taken from kaggle [3]. Following sections are divided into two sections with one containing the data sets of product images and other containing a csv file with label description of the products title both are being contained into a variables for preprocessing as shown in the Figure 4. These are used in the respective image and text processing models , which is concatenated at the end yield an output used to satisfy the research objectives.

5 Implementation

5.1 Importing Libraries

The implementation of the project using the python is described in the below sections.Please run them step by step as described. Before we begin the implementation the first step is to perform the preprocessing for the given data. the below figure 4 shows the libraries needed for the startup ,

1 EDA

```
[2]: from google.colab import drive
     import pandas as pd
     import matplotlib.pyplot as plt
     import numpy as np
     import cv2
     import textwrap as wrap

     zip_path = '/content/drive/MyDrive/shopee-product-matching.zip'
     extract_path = '/content/drive/MyDrive/path/shopee-product-matching'
     PATH_TO_IMG ='/content/drive/MyDrive/path/shopee-product-matching/train_images/'
     PATH_TO_TEST = '/content/drive/MyDrive/path/shopee-product-matching/test_images/
      ↪'


     train=pd.read_csv(extract_path + '/train.csv')
     train.head()
```

[2]:

	posting_id	image	image_phash	title	label_group
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	249114794
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO…	2937985045
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	2395904891
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp…	4093212188
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	3648931069

Figure 3: Importing Required Libraries

5.2 Splitting of Train and Test Data

The given data set comprises of image and text data , while the image dataset are not considered in the pre-processing now , let us consider the csv fie with the label description as shown in the Figure below ,

[3] https://www.kaggle.com/competitions/shopee-product-matching/data

```
[5]: train = pd.read_csv(extract_path+'/train.csv')
     #val = pd.read_csv(extract_path+'/val.csv')
     test = pd.read_csv(extract_path+'/test.csv')
```

1

```
[6]: train.head()
```

[6]:

	posting_id	image	image_phash	title	label_group
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	249114794
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO…	2937985045
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	2395904891
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp…	4093212188
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	3648931069

```
[7]: train.shape
```

[7]: (34250, 5)

```
[8]: unique_labels = train['label_group'].unique()
     len(unique_labels)
```

[8]: 11014

Figure 4: Train / Test Split

5.3 Data Analysis and Data augmentation

5.3.1 Data Analysis

The analysis on the given data containing the csv file is performed as shown in the figure 5 below ,

```
[9]: # how many images in total
     train["image"].nunique()
```

[9]: 32412

```
[10]: train.groupby('label_group').count().sort_values(by='posting_id',␣
       ↪ascending=False)
```

[10]:

label_group	posting_id	image	image_phash	title
1163569239	51	51	51	51
159351600	51	51	51	51
994676122	51	51	51	51
3113678103	51	51	51	51
3627744656	51	51	51	51
…	…	…	…	…
2357508171	2	2	2	2
2357372960	2	2	2	2
2357221297	2	2	2	2
2355878351	2	2	2	2
2141883596	2	2	2	2

2

[11014 rows x 4 columns]

```
[11]: # check title length range by word
      train['title'].str.split().apply(lambda x: len(x)).describe()
```

[11]:
```
count    34250.000000
mean         9.414861
std          4.446719
min          1.000000
25%          6.000000
50%          9.000000
```

Figure 5: Text Data Analysis

```
[12]: title_unique = train['title'].nunique()
      title_unique

[12]: 33117

[13]: # check the id count
      posting_unique = train['posting_id'].nunique()
      posting_unique

[13]: 34250

[14]: title_unique/posting_unique

[14]: 0.9669197080291971

[15]: len(train['posting_id'])

[15]: 34250

[16]: train["image_phash"].nunique() / len(train["image_phash"])

[16]: 0.838978102189781

[17]: # check the distribution of the number of images in each group
      group_img_dist = train.groupby('label_group').count().
        ↪sort_values(by='posting_id', ascending=False)['posting_id'].value_counts()

[18]: # change group_img_dist to dataframe
      group_img_dist = pd.DataFrame(group_img_dist)
      group_img_dist.head()
```

Figure 6: Images Distribution

[18]:
```
   posting_id
2        6979
3        1779
4         862
5         468
6         282
```

[27]:
```
# for group_img_dist, combine 1 to 3, and sum the posting_id count
img_count_1_3 = group_img_dist.loc[2:3].sum()
img_count_4_10 = group_img_dist.loc[4:5].sum()
img_count_11_more = group_img_dist.loc[6:].sum()
```

[28]:
```
img_dist = [img_count_1_3, img_count_4_10, img_count_11_more]
```

[29]:
```
img_dist = pd.DataFrame(img_dist)
```

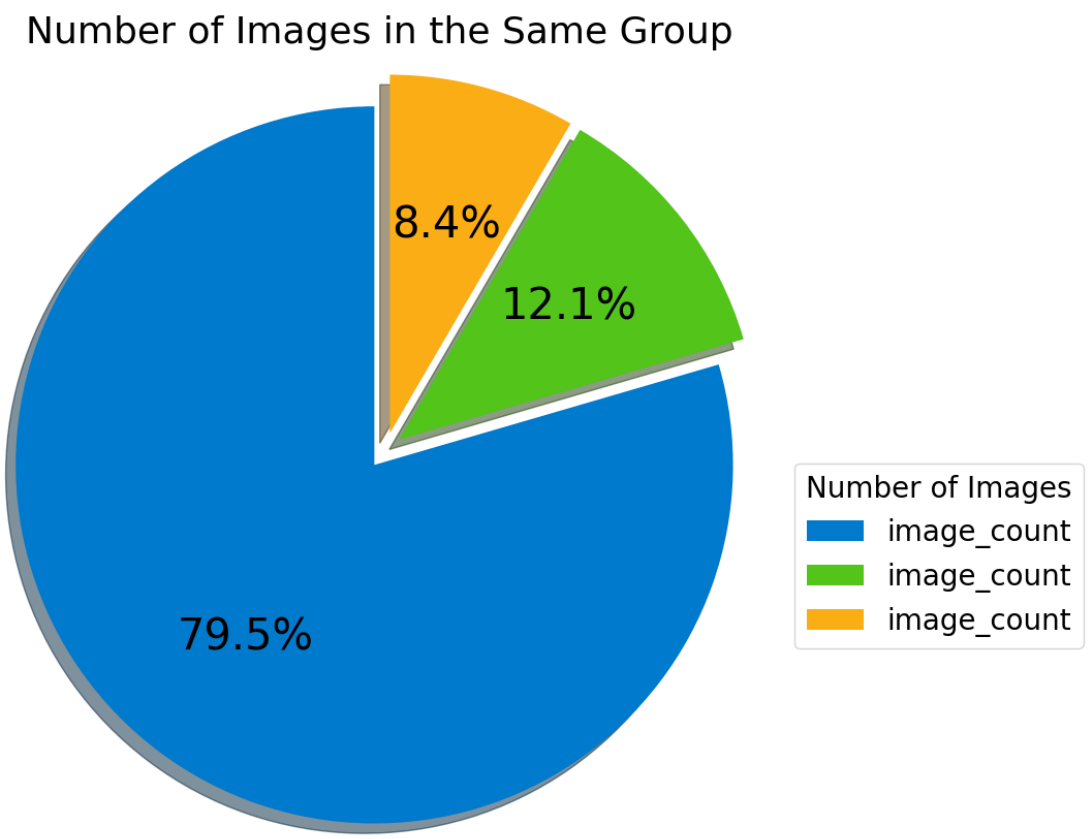
[34]:
```
# insert first column to img_dist as "image_count" and set the value to "Less␣
 ↪than 3 images", "4 to 10 images", "11 and more images"
img_dist.insert(0, "image_count", ["Less than 3 images", "4 to 5 images", "6␣
 ↪and more images"], True)

# rename the column name
img_dist.rename(columns={0: "count"}, inplace=True)

# reset the index
img_dist.reset_index(drop=True, inplace=True)
```

[45]:
```
img_dist
```

[45]:

	image_count	image_count	image_count	posting_id
0	Less than 3 images	Less than 3 images	Less than 3 images	8758
1	4 to 5 images	4 to 5 images	4 to 10 images	1330
2	6 and more images	6 and more images	11 and more images	926

[36]:
```
# plot the pie chart
plt.figure(figsize=(10,10))
explode = [0.05]*len(img_dist) # add a slight separation between pie slices
colors = ['#007acc', '#52c41a', '#faad14', '#f5222d'] # custom color palette
plt.pie(img_dist['posting_id'], labels=None, autopct='%.
 ↪1f%%',textprops={'fontsize': 30}, shadow=True, startangle=90,␣
 ↪explode=explode, colors=colors)
plt.legend(labels=img_dist['image_count'], title='Number of Images',␣
 ↪title_fontsize=20, loc="best", bbox_to_anchor=(1, 0.5), fontsize=20)
plt.title('Number of Images in the Same Group', fontsize=26)
plt.axis('equal')
plt.show()
```



[37]:
```
# make a pie chart for the distribution of the number of posting in each group
group_posting_dist = train.groupby('label_group').count().
  ↪sort_values(by='posting_id', ascending=False)['posting_id'].value_counts()

# change group_posting_dist to dataframe
group_posting_dist = pd.DataFrame(group_posting_dist)
group_posting_dist.head()
```

[37]:

	posting_id
2	6979
3	1779
4	862
5	468
6	282

[39]:
```
posting_count_1_3 = group_posting_dist.loc[2:3].sum()
posting_count_4_10 = group_posting_dist.loc[4:5].sum()
posting_count_11_more = group_posting_dist.loc[6:].sum()
```

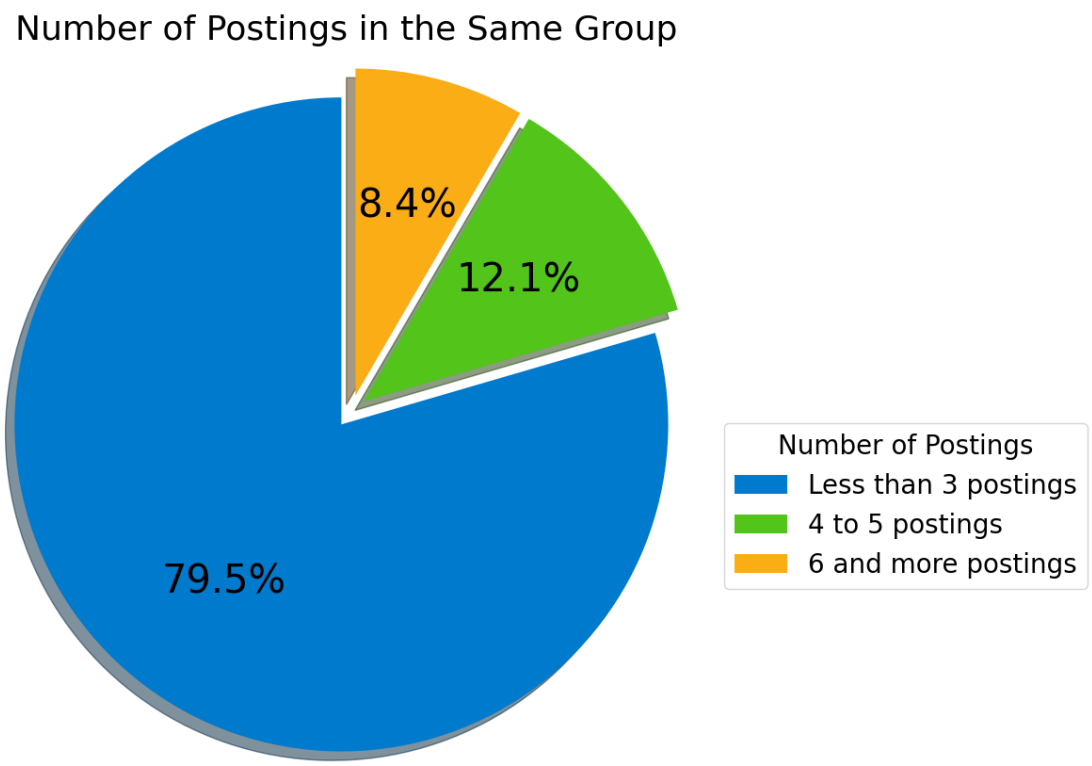
[43]:
```
posting_dist = [posting_count_1_3, posting_count_4_10, posting_count_11_more]
posting_dist = pd.DataFrame(posting_dist)

# insert first column to posting_dist as "posting_count" and set the value to␣
 ↪"Less than 3 posting", "4 to 10 posting", "11 and more posting"
posting_dist.insert(0, "posting_count", ["Less than 3 postings", "4 to 5␣
 ↪postings", "6 and more postings"], True)
```

[44]:
```
posting_dist
```

[44]:

	posting_count	posting_id
0	Less than 3 postings	8758
1	4 to 5 postings	1330
2	6 and more postings	926

[46]:
```
# plot the pie chart
plt.figure(figsize=(10,10))
explode = [0.05]*len(posting_dist) # add a slight separation between pie slices
colors = ['#007acc', '#52c41a', '#faad14', '#f5222d'] # custom color palette
plt.pie(posting_dist['posting_id'], labels=None, autopct='%.
 ↪1f%%',textprops={'fontsize': 30}, shadow=True, startangle=90,␣
 ↪explode=explode, colors=colors)
plt.legend(labels=posting_dist['posting_count'], title='Number of Postings',␣
 ↪title_fontsize=20, loc="best", bbox_to_anchor=(1, 0.5), fontsize=20)
plt.title('Number of Postings in the Same Group', fontsize=26)
plt.axis('equal')
plt.show()
```



[47]:
```
def show_same_img():
    # choose randomly two instances per each class
    labels_to_show = np.random.choice(train.label_group.unique(),
                                      replace=True, size=27)
    img_to_show = []
    for label in labels_to_show:
        rows = train[train.label_group==label].copy()
        pair = np.random.choice([i for i in range(len(rows))],
                                    replace=True, size=2)
        img_pair = rows.iloc[pair][['image', 'title']].values

        img_to_show += list(img_pair)

    fig, axes = plt.subplots(figsize = (18, 12), nrows=2,ncols=2)
    for imp, ax in zip(img_to_show, axes.ravel()):
        img = cv2.imread(PATH_TO_IMG + imp[0])
        title = '\n'.join(wrap(imp[1], 20))
        ax.set_title(title)
        ax.imshow(img)
        ax.axis('off')

    fig.tight_layout()
```

[48]:
```
import os
import numpy as np
import pandas as pd
import cv2
import matplotlib.pyplot as plt
from textwrap import wrap
# from wordcloud import WordCloud
```

[49]:
```
num_imgs = len(os.listdir(PATH_TO_IMG))
print("Number of images in train set: ", num_imgs)
```

```
Number of images in train set:  32442
```

[52]:
```
test_df = pd.read_csv(extract_path + "/test.csv")
compute_cv = len(test_df) <= 3

if compute_cv:
    train_df = pd.read_csv(extract_path + "/train.csv")
    target_dict = train_df.groupby("label_group")["posting_id"].agg("unique").
 ↪to_dict()
    train_df["target"] = train_df["label_group"].map(target_dict)
    data = train_df
else:
    data = test_df
```

[53]:
```
num_unique_label=len(data["label_group"].unique())
print("Number of unique label groups: ", num_unique_label)
```

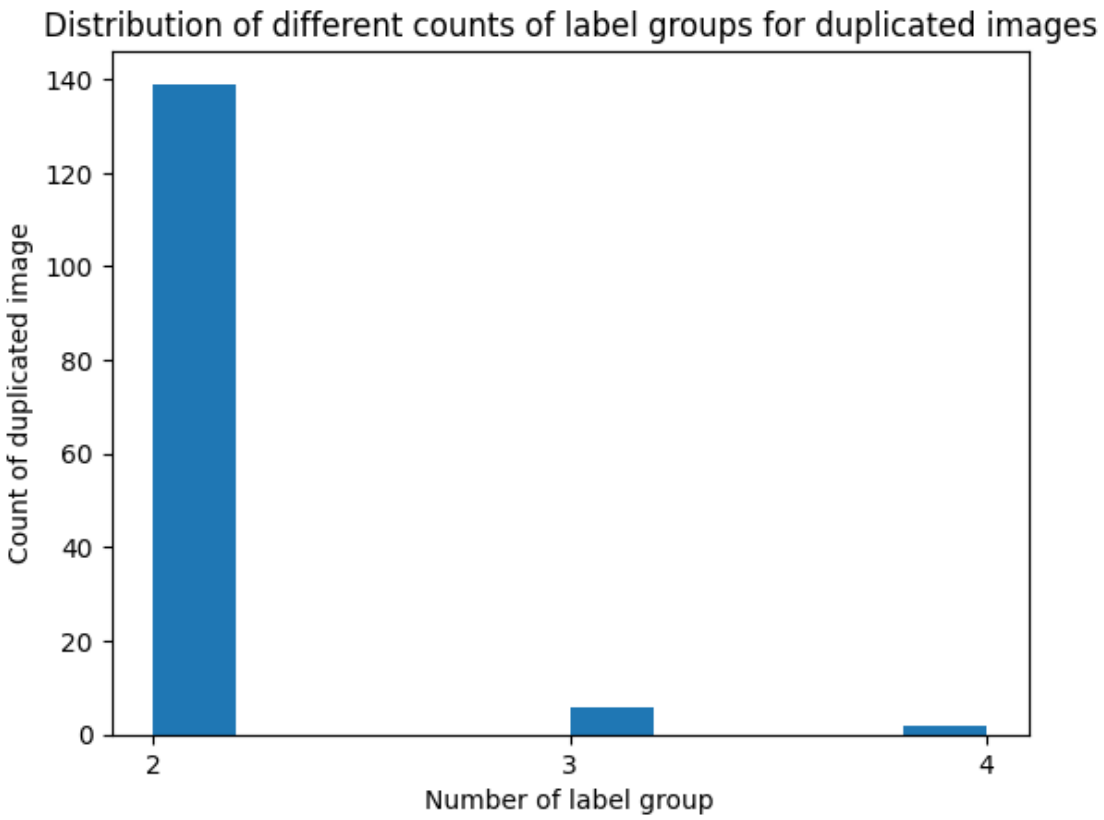
```
Number of unique label groups:  11014
```

[54]:
```
unique_hash=len(data['image_phash'].unique())
print("Number of unique image phash: ", unique_hash)
```

```
Number of unique image phash:  28735
```

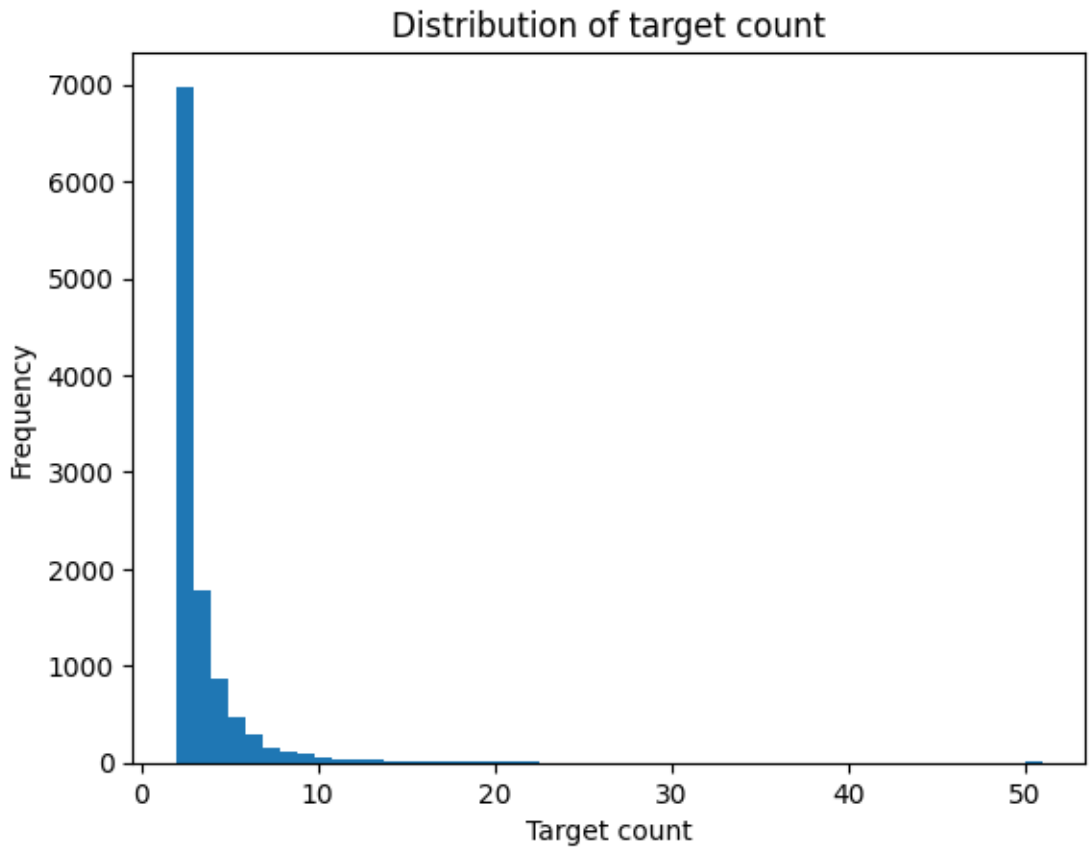
[55]:
```
duplicates = data[data.duplicated(subset='image_phash', keep=False)]

# Group duplicates by image_phash, and count the number of unique label_groups␣
 ↪for each image_phash
counts = duplicates.groupby('image_phash')['label_group'].nunique()

# Filter the counts to show only the image_phash values that have more than one␣
 ↪label_group
counts = counts[counts > 1]

# Plot the histogram
counts_int = counts.values.astype(int)
```

```
plt.hist(counts_int, bins=10)
plt.xlabel('Number of label group')
plt.ylabel('Count of duplicated image')
plt.title('Distribution of different counts of label groups for duplicated␣
 ↪images')

# Set X axis tick labels to integers only
plt.xticks(np.arange(counts_int.min(), counts_int.max()+1, 1.0))


plt.show()
```



[56]:
```
target_counts = data['label_group'].value_counts()

# Plot the distribution of target counts
fig, ax = plt.subplots()
ax.hist(target_counts, bins=50)
ax.set_xlabel('Target count')
ax.set_ylabel('Frequency')
ax.set_title('Distribution of target count')
```

```
plt.show()
```



[58]:
```
def show_random_images():
    n_rows, n_cols = 1, 2
    n_images = n_rows * n_cols
    label_groups = np.random.choice(data["label_group"].unique(),␣
 ↪size=n_images, replace=False)

    fig, axes = plt.subplots(figsize=(20, 10), nrows=n_rows, ncols=n_cols)
    img_to_show = []
    for label_group in label_groups:
        rows = data[data["label_group"] == label_group].sample(n=2,␣
 ↪replace=False)
        img_to_show += rows[["image", "title"]].values.tolist()

    if len(img_to_show) > n_images:
        img_to_show = img_to_show[:n_images]

    for i, (img_path, title) in enumerate(img_to_show):
```

```
        img = cv2.imread(os.path.join(PATH_TO_IMG, img_path))
        title = "\n".join(wrap(title, 44))
        ax = axes.flat[i]
        ax.set_title(title, fontsize=30)
        ax.imshow(img)
        ax.axis("off")

    fig.tight_layout()
show_random_images()
```

Outer/kardigan lilit bali

FRINGE OUTER JUMPUT

RAPIDS cuDF

Open in Colab

Environment Setup

Check Version

Python Version

In [ ]:
```
# Check Python Version
!python --version
```

```
Python 3.10.12
```

Ubuntu Version

In [ ]:
```
# Check Ubuntu Version
!lsb_release -a
```

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
```

Check CUDA Version

In [ ]:
```
# Check CUDA/cuDNN Version
!nvcc -V && which nvcc
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
/usr/local/cuda/bin/nvcc
```

Check GPU Version

In [ ]:
```
# Check GPU
!nvidia-smi
```

```
Wed Aug  9 11:14:02 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   67C    P8    14W /  70W |      0MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Setup:

This set up script:

1. Checks to make sure that the GPU is RAPIDS compatible
2. Installs the **current stable version** of RAPIDSAI's core libraries using pip, which are:
    A. cuDF
    B. cuML
    C. cuGraph
    D. xgboost

This will complete in about 3-4 minutes

Please use the RAPIDS Conda Colab Template notebook if you need to install any of RAPIDS Extended libraries, such as:

- cuSpatial
- cuSignal
- cuxFilter
- cuCIM

OR

- nightly versions of any library

In [ ]:
```
# This get the RAPIDS-Colab install files and test check your GPU.  Run this and th
# Please read the output of this cell.  If your Colab Instance is not RAPIDS compat
!git clone https://github.com/rapidsai/rapidsai-csp-utils.git
!python rapidsai-csp-utils/colab/pip-install.py
```

```
Cloning into 'rapidsai-csp-utils'...
remote: Enumerating objects: 390, done.
remote: Counting objects: 100% (121/121), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 390 (delta 89), reused 51 (delta 51), pack-reused 269
Receiving objects: 100% (390/390), 107.11 KiB | 2.06 MiB/s, done.
Resolving deltas: 100% (191/191), done.
Collecting pynvml
  Downloading pynvml-11.5.0-py3-none-any.whl (53 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.1/53.1 kB 991.5 kB/s eta 0:00:00
Installing collected packages: pynvml
Successfully installed pynvml-11.5.0
***********************************************************************
Woo! Your instance has the right kind of GPU, a Tesla T4!
We will now install RAPIDS cuDF, cuML, and cuGraph via pip!
Please stand by, should be quick...
***********************************************************************

Looking in indexes: https://pypi.org/simple, https://pypi.nvidia.com
Collecting cudf-cu11
  Downloading https://pypi.nvidia.com/cudf-cu11/cudf_cu11-23.6.1-cp310-cp310-manyl
inux_2_17_x86_64.manylinux2014_x86_64.whl (489.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 489.3/489.3 MB 2.8 MB/s eta 0:00:00
Collecting cuml-cu11
  Downloading https://pypi.nvidia.com/cuml-cu11/cuml_cu11-23.6.0-cp310-cp310-manyl
inux_2_17_x86_64.manylinux2014_x86_64.whl (1079.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 GB 1.0 MB/s eta 0:00:00
Collecting cugraph-cu11
  Downloading https://pypi.nvidia.com/cugraph-cu11/cugraph_cu11-23.6.2-cp310-cp310
-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1160.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 GB 968.6 kB/s eta 0:00:00
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages
(3.8.5)
Requirement already satisfied: cachetools in /usr/local/lib/python3.10/dist-packag
es (from cudf-cu11) (5.3.1)
Collecting cubinlinker-cu11 (from cudf-cu11)
  Downloading https://pypi.nvidia.com/cubinlinker-cu11/cubinlinker_cu11-0.3.0.post
1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.8/8.8 MB 105.0 MB/s eta 0:00:00
Collecting cuda-python<12.0,>=11.7.1 (from cudf-cu11)
  Downloading cuda_python-11.8.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl (16.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.5/16.5 MB 41.8 MB/s eta 0:00:00
Collecting cupy-cuda11x>=12.0.0 (from cudf-cu11)
  Downloading cupy_cuda11x-12.2.0-cp310-cp310-manylinux2014_x86_64.whl (89.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 89.6/89.6 MB 9.4 MB/s eta 0:00:00
Requirement already satisfied: fsspec>=0.6.0 in /usr/local/lib/python3.10/dist-pac
kages (from cudf-cu11) (2023.6.0)
Collecting numba>=0.57 (from cudf-cu11)
  Downloading numba-0.57.1-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.
whl (3.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.6/3.6 MB 76.8 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.21 in /usr/local/lib/python3.10/dist-packa
ges (from cudf-cu11) (1.23.5)
Collecting nvtx>=0.2.1 (from cudf-cu11)
  Downloading nvtx-0.2.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.wh
l (553 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 553.1/553.1 kB 51.3 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-package
s (from cudf-cu11) (23.1)
Requirement already satisfied: pandas<1.6.0dev0,>=1.3 in /usr/local/lib/python3.1
0/dist-packages (from cudf-cu11) (1.5.3)
Collecting protobuf<4.22,>=4.21.6 (from cudf-cu11)
  Downloading protobuf-4.21.12-cp37-abi3-manylinux2014_x86_64.whl (409 kB)
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 409.8/409.8 kB 41.5 MB/s eta 0:00:00
Collecting ptxcompiler-cu11 (from cudf-cu11)
  Downloading https://pypi.nvidia.com/ptxcompiler-cu11/ptxcompiler_cu11-0.7.0.post
1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.8/8.8 MB 92.0 MB/s eta 0:00:00
Collecting pyarrow==11.* (from cudf-cu11)
  Downloading pyarrow-11.0.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (34.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.9/34.9 MB 19.4 MB/s eta 0:00:00
Collecting rmm-cu11==23.6.* (from cudf-cu11)
  Downloading https://pypi.nvidia.com/rmm-cu11/rmm_cu11-23.6.0-cp310-cp310-manylin
ux_2_17_x86_64.manylinux2014_x86_64.whl (1.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 93.1 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.
10/dist-packages (from cudf-cu11) (4.7.1)
Collecting dask-cuda==23.6.* (from cuml-cu11)
  Downloading dask_cuda-23.6.0-py3-none-any.whl (125 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 125.2/125.2 kB 16.4 MB/s eta 0:00:00
Collecting dask-cudf-cu11==23.6.* (from cuml-cu11)
  Downloading https://pypi.nvidia.com/dask-cudf-cu11/dask_cudf_cu11-23.6.0-py3-non
e-any.whl (79 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.6/79.6 kB 10.2 MB/s eta 0:00:00
Collecting dask==2023.3.2 (from cuml-cu11)
  Downloading dask-2023.3.2-py3-none-any.whl (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 85.5 MB/s eta 0:00:00
Collecting distributed==2023.3.2.1 (from cuml-cu11)
  Downloading distributed-2023.3.2.1-py3-none-any.whl (957 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 957.1/957.1 kB 75.5 MB/s eta 0:00:00
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.10/dist-pack
ages (from cuml-cu11) (1.3.1)
Collecting raft-dask-cu11==23.6.* (from cuml-cu11)
  Downloading https://pypi.nvidia.com/raft-dask-cu11/raft_dask_cu11-23.6.2-cp310-c
p310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (214.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 214.7/214.7 MB 5.7 MB/s eta 0:00:00
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (f
rom cuml-cu11) (1.10.1)
Collecting treelite==3.2.0 (from cuml-cu11)
  Downloading treelite-3.2.0-py3-none-manylinux2014_x86_64.whl (1.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 74.6 MB/s eta 0:00:00
Collecting treelite-runtime==3.2.0 (from cuml-cu11)
  Downloading treelite_runtime-3.2.0-py3-none-manylinux2014_x86_64.whl (198 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 198.2/198.2 kB 22.6 MB/s eta 0:00:00
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.10/dist-packag
es (from dask==2023.3.2->cuml-cu11) (8.1.6)
Requirement already satisfied: cloudpickle>=1.1.1 in /usr/local/lib/python3.10/dis
t-packages (from dask==2023.3.2->cuml-cu11) (2.2.1)
Requirement already satisfied: partd>=1.2.0 in /usr/local/lib/python3.10/dist-pack
ages (from dask==2023.3.2->cuml-cu11) (1.4.0)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-pac
kages (from dask==2023.3.2->cuml-cu11) (6.0.1)
Requirement already satisfied: toolz>=0.8.2 in /usr/local/lib/python3.10/dist-pack
ages (from dask==2023.3.2->cuml-cu11) (0.12.0)
Collecting importlib-metadata>=4.13.0 (from dask==2023.3.2->cuml-cu11)
  Downloading importlib_metadata-6.8.0-py3-none-any.whl (22 kB)
Collecting pynvml<11.5,>=11.0.0 (from dask-cuda==23.6.*->cuml-cu11)
  Downloading pynvml-11.4.1-py3-none-any.whl (46 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 47.0/47.0 kB 6.3 MB/s eta 0:00:00
Requirement already satisfied: zict>=2.0.0 in /usr/local/lib/python3.10/dist-packa
ges (from dask-cuda==23.6.*->cuml-cu11) (3.0.0)
Requirement already satisfied: jinja2>=2.10.3 in /usr/local/lib/python3.10/dist-pa
ckages (from distributed==2023.3.2.1->cuml-cu11) (3.1.2)
Requirement already satisfied: locket>=1.0.0 in /usr/local/lib/python3.10/dist-pac
kages (from distributed==2023.3.2.1->cuml-cu11) (1.0.0)
Requirement already satisfied: msgpack>=1.0.0 in /usr/local/lib/python3.10/dist-pa
```

```
ckages (from distributed==2023.3.2.1->cuml-cu11) (1.0.5)
Requirement already satisfied: psutil>=5.7.0 in /usr/local/lib/python3.10/dist-pac
kages (from distributed==2023.3.2.1->cuml-cu11) (5.9.5)
Requirement already satisfied: sortedcontainers>=2.0.5 in /usr/local/lib/python3.1
0/dist-packages (from distributed==2023.3.2.1->cuml-cu11) (2.4.0)
Requirement already satisfied: tblib>=1.6.0 in /usr/local/lib/python3.10/dist-pack
ages (from distributed==2023.3.2.1->cuml-cu11) (2.0.0)
Requirement already satisfied: tornado>=6.0.3 in /usr/local/lib/python3.10/dist-pa
ckages (from distributed==2023.3.2.1->cuml-cu11) (6.3.1)
Requirement already satisfied: urllib3>=1.24.3 in /usr/local/lib/python3.10/dist-p
ackages (from distributed==2023.3.2.1->cuml-cu11) (1.26.16)
Collecting pylibraft-cu11==23.6.* (from raft-dask-cu11==23.6.*->cuml-cu11)
  Downloading https://pypi.nvidia.com/pylibraft-cu11/pylibraft_cu11-23.6.2-cp310-c
p310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (471.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 471.7/471.7 MB 2.7 MB/s eta 0:00:00
Collecting ucx-py-cu11==0.32.* (from raft-dask-cu11==23.6.*->cuml-cu11)
  Downloading https://pypi.nvidia.com/ucx-py-cu11/ucx_py_cu11-0.32.0-cp310-cp310-m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.9/7.9 MB 107.3 MB/s eta 0:00:00
Collecting pylibcugraph-cu11==23.6.* (from cugraph-cu11)
  Downloading https://pypi.nvidia.com/pylibcugraph-cu11/pylibcugraph_cu11-23.6.2-c
p310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1159.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 GB 1.3 MB/s eta 0:00:00
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-pac
kages (from aiohttp) (23.1.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in /usr/local/lib/pyth
on3.10/dist-packages (from aiohttp) (3.2.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/di
st-packages (from aiohttp) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/pytho
n3.10/dist-packages (from aiohttp) (4.0.2)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-pa
ckages (from aiohttp) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist
-packages (from aiohttp) (1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-
packages (from aiohttp) (1.3.1)
Requirement already satisfied: cython in /usr/local/lib/python3.10/dist-packages
(from cuda-python<12.0,>=11.7.1->cudf-cu11) (0.29.36)
Requirement already satisfied: fastrlock>=0.5 in /usr/local/lib/python3.10/dist-pa
ckages (from cupy-cuda11x>=12.0.0->cudf-cu11) (0.8.1)
Collecting llvmlite<0.41,>=0.40.0dev0 (from numba>=0.57->cudf-cu11)
  Downloading llvmlite-0.40.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_
64.whl (42.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.1/42.1 MB 16.8 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.1
0/dist-packages (from pandas<1.6.0dev0,>=1.3->cudf-cu11) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-pack
ages (from pandas<1.6.0dev0,>=1.3->cudf-cu11) (2022.7.1)
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.10/dist-package
s (from yarl<2.0,>=1.0->aiohttp) (3.4)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-package
s (from importlib-metadata>=4.13.0->dask==2023.3.2->cuml-cu11) (3.16.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-p
ackages (from jinja2>=2.10.3->distributed==2023.3.2.1->cuml-cu11) (2.1.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages
(from python-dateutil>=2.8.1->pandas<1.6.0dev0,>=1.3->cudf-cu11) (1.16.0)
Installing collected packages: ptxcompiler-cu11, nvtx, cubinlinker-cu11, pynvml, p
yarrow, protobuf, llvmlite, importlib-metadata, cupy-cuda11x, cuda-python, ucx-py-
cu11, treelite-runtime, treelite, numba, dask, rmm-cu11, distributed, pylibraft-cu
11, dask-cuda, cudf-cu11, raft-dask-cu11, pylibcugraph-cu11, dask-cudf-cu11, cuml-
cu11, cugraph-cu11
  Attempting uninstall: pynvml
    Found existing installation: pynvml 11.5.0
```

```
    Uninstalling pynvml-11.5.0:
      Successfully uninstalled pynvml-11.5.0
  Attempting uninstall: pyarrow
    Found existing installation: pyarrow 9.0.0
    Uninstalling pyarrow-9.0.0:
      Successfully uninstalled pyarrow-9.0.0
  Attempting uninstall: protobuf
    Found existing installation: protobuf 3.20.3
    Uninstalling protobuf-3.20.3:
      Successfully uninstalled protobuf-3.20.3
  Attempting uninstall: llvmlite
    Found existing installation: llvmlite 0.39.1
    Uninstalling llvmlite-0.39.1:
      Successfully uninstalled llvmlite-0.39.1
  Attempting uninstall: importlib-metadata
    Found existing installation: importlib-metadata 4.6.4
    Uninstalling importlib-metadata-4.6.4:
      Successfully uninstalled importlib-metadata-4.6.4
  Attempting uninstall: cupy-cuda11x
    Found existing installation: cupy-cuda11x 11.0.0
    Uninstalling cupy-cuda11x-11.0.0:
      Successfully uninstalled cupy-cuda11x-11.0.0
  Attempting uninstall: numba
    Found existing installation: numba 0.56.4
    Uninstalling numba-0.56.4:
      Successfully uninstalled numba-0.56.4
  Attempting uninstall: dask
    Found existing installation: dask 2022.12.1
    Uninstalling dask-2022.12.1:
      Successfully uninstalled dask-2022.12.1
  Attempting uninstall: distributed
    Found existing installation: distributed 2022.12.1
    Uninstalling distributed-2022.12.1:
      Successfully uninstalled distributed-2022.12.1
ERROR: pip's dependency resolver does not currently take into account all the pack
ages that are installed. This behaviour is the source of the following dependency
conflicts.
pandas-gbq 0.17.9 requires pyarrow<10.0dev,>=3.0.0, but you have pyarrow 11.0.0 wh
ich is incompatible.
Successfully installed cubinlinker-cu11-0.3.0.post1 cuda-python-11.8.2 cudf-cu11-2
3.6.1 cugraph-cu11-23.6.2 cuml-cu11-23.6.0 cupy-cuda11x-12.2.0 dask-2023.3.2 dask-
cuda-23.6.0 dask-cudf-cu11-23.6.0 distributed-2023.3.2.1 importlib-metadata-6.8.0
llvmlite-0.40.1 numba-0.57.1 nvtx-0.2.6 protobuf-4.21.12 ptxcompiler-cu11-0.7.0.po
st1 pyarrow-11.0.0 pylibcugraph-cu11-23.6.2 pylibraft-cu11-23.6.2 pynvml-11.4.1 ra
ft-dask-cu11-23.6.2 rmm-cu11-23.6.0 treelite-3.2.0 treelite-runtime-3.2.0 ucx-py-c
u11-0.32.0
Requirement already satisfied: cupy-cuda11x in /usr/local/lib/python3.10/dist-pack
ages (12.2.0)
Requirement already satisfied: numpy<1.27,>=1.20 in /usr/local/lib/python3.10/dist
-packages (from cupy-cuda11x) (1.23.5)
Requirement already satisfied: fastrlock>=0.5 in /usr/local/lib/python3.10/dist-pa
ckages (from cupy-cuda11x) (0.8.1)

          *******************************************************************
          The pip install of RAPIDS is complete.

          Please do not run any further installation from the conda based installa
tion methods, as they may cause issues!

          Please ensure that you're pulling from the git repo to remain updated wi
th the latest working install scripts.
r
          Troubleshooting:
             - If there is an installation failure, please check back on RAPIDSAI
```

```
owned templates/notebooks to see how to update your personal files.
            - If an installation failure persists when using the latest script, p
lease make an issue on https://github.com/rapidsai-community/rapidsai-csp-utils
         ********************************************************************
```

In [ ]:
```
pip install transformers
```

```
Collecting transformers
  Downloading transformers-4.31.0-py3-none-any.whl (7.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.4/7.4 MB 19.8 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages
(from transformers) (3.12.2)
Collecting huggingface-hub<1.0,>=0.14.1 (from transformers)
  Downloading huggingface_hub-0.16.4-py3-none-any.whl (268 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 268.8/268.8 kB 27.8 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packa
ges (from transformers) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-p
ackages (from transformers) (23.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packa
ges (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist
-packages (from transformers) (2022.10.31)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages
(from transformers) (2.31.0)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1 (from transformers)
  Downloading tokenizers-0.13.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl (7.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 45.8 MB/s eta 0:00:00
Collecting safetensors>=0.3.1 (from transformers)
  Downloading safetensors-0.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl (1.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 47.5 MB/s eta 0:00:00
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packag
es (from transformers) (4.65.0)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from huggingface-hub<1.0,>=0.14.1->transformers) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python
3.10/dist-packages (from huggingface-hub<1.0,>=0.14.1->transformers) (4.7.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.
10/dist-packages (from requests->transformers) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-pack
ages (from requests->transformers) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dis
t-packages (from requests->transformers) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dis
t-packages (from requests->transformers) (2023.7.22)
Installing collected packages: tokenizers, safetensors, huggingface-hub, transform
ers
Successfully installed huggingface-hub-0.16.4 safetensors-0.3.2 tokenizers-0.13.3
transformers-4.31.0
```

Critical Imports

In [ ]:
```
# Critical imports
import cudf
import cuml
import os
import numpy as np
import pandas as pd
# adding new imports
import numpy as np
```

```
import cupy, cudf
import gc
import pandas as pd
from tqdm import tqdm
tqdm.pandas()
import random
import torch
import torchvision
from torchvision import  models, transforms
from transformers import BertTokenizer, BertModel
from cuml.feature_extraction.text import TfidfVectorizer
from cuml.neighbors import NearestNeighbors
import torch.nn as nn
import torch.nn.functional as F
import os
import glob
from PIL import Image
import seaborn as sns
import cv2, matplotlib.pyplot as plt
import matplotlib.image as mpimg
from textwrap import wrap
```

Data import

In [ ]:
```
device = 'cuda'if torch.cuda.is_available() else 'cpu'
device
```

Out[ ]:
```
'cuda'
```

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:
```
import zipfile

zip_path = '/content/drive/MyDrive/shopee-product-matching.zip'
extract_path = '/content/drive/MyDrive/path/shopee-product-matching'
PATH_TO_IMG ='/content/drive/MyDrive/path/shopee-product-matching/train_images/'
PATH_TO_TEST = '/content/drive/MyDrive/path/shopee-product-matching/test_images/'
#with zipfile.ZipFile(zip_path, 'r') as zip_ref:
 #   zip_ref.extractall(extract_path)
```

In [ ]:
```
print(extract_path)
```

```
/content/drive/MyDrive/path/shopee-product-matching
```

In [ ]:
```
def list_sp_files(path_list):
    for path in path_list:
        for filename in os.listdir(path):
            print(path, filename)
```

In [ ]:
```
res = []
for path in os.listdir(extract_path):
    # check if current path is a file
    if os.path.isdir(os.path.join(extract_path, path)):
        res.append(path)
print(res)
```

```
['test_images', 'train_images', 'sentence-transformer']
```

In [ ]:

```
import os

# Specify the path to the subfolder
subfolder_path = '/content/drive/MyDrive/path/shopee-product-matching/train_images

# List the contents of the subfolder
contents = os.listdir(subfolder_path)

# Print the contents
for item in contents:
    print(item)
```

```
Streaming output truncated to the last 5000 lines.
27e1e22db5b72a2ff80af5c33a86ad5f.jpg
27f8cd710447ccdc4be680330bd6425e.jpg
27d66999236ae5ffadd0125d68eb90e3.jpg
27dcd98558dc619910d93fc5d7211c25.jpg
27f2b0e6119889111c59a10696d291cb.jpg
27e50e14b6d8e9c039f6a4b4f1ffa7dc.jpg
27db63d24546367e2f86077fd05b268a.jpg
27facd9afa37f3fcc91b3b93fccd4b29.jpg
28056181fa2ce14fd1c4b0c47c535097.jpg
27c7d6082dcbdc494295853c7cdf99fc.jpg
27d8e2b88c717a93633022eb43e50116.jpg
2820f1192344b6c6cf52d57dd620a44f.jpg
2818f62e52cfdfc1dbe10fd8bdc14908.jpg
281c76b578e4a473d6021f711f0794d1.jpg
281235e589dc4a0b6ceabcd64a53af4e.jpg
282480dba1296d7b3f3d9dabe59921be.jpg
282559d15e204c70b65b4021b855173f.jpg
2820ae904a947b8176fbff4633bb6f1a.jpg
280a59b854a8dd1542e5ac4b12ec5c2f.jpg
28239d2df8eedee688826e2d55a5d210.jpg
281a23efc786b3de196ffca5fdbaf876.jpg
280fce37634fb6ee3b9996cd8baa3b4a.jpg
28065c1a1d4f66c2644d72600747b7de.jpg
2820d2301c83b7770d2653f2f53d70a5.jpg
280b33d16ba86d113727e21b8bd32756.jpg
28093c77e12623f4ba449c3397b0b8ec.jpg
280b1f800dbee30473b55c3dd4bb9d52.jpg
28167561778e2f44eccc6acf2ceca330.jpg
2814c9c9549f62bfd6ed00f08e9d7d2f.jpg
280b5998bc49cc35d0d5e0d48633fbc9.jpg
2812faba1e0d58c6a662638a97219c3d.jpg
28162c73c9cb5a413921df4e29d4ded8.jpg
281f120c64376d45c1aed97c9fbc3171.jpg
280a534e684d93655a2ce71bdae5bea6.jpg
28399084914ca4aa9d4a98823b293e69.jpg
28461bf2358effc84977d8ef5428c6c8.jpg
283140c972494f6e83904e082ef61421.jpg
2826339257579760a00d6e0a0065a89e.jpg
28540714be9f019041dd7f5d02c344d1.jpg
2854307b190a3d9da2b824651cc1c5b8.jpg
28347927145e3db04609685a5b208523.jpg
2832ed41b8e32112eb069a0a67959239.jpg
2836cccbdc6fdc733358619a1cf6ca3c.jpg
28400d9c7817b84eed634e37b39cec6a.jpg
2833579f35684118327e9cf04879246e.jpg
284f9582dedef50f1a32b692411705b8.jpg
2836d94df6253a117dc497686ad9012e.jpg
283a30e139b4ee2d062a1856ee2bc2fe.jpg
28459ad0efc782f937fad07cd8b9751a.jpg
2829d11d9939b6a89ce093844213cfb4.jpg
2850a7d0c2238047ad84190b90ec34bd.jpg
284fd69594d9e1cae652b336848b6d62.jpg
282f75dccfb50e1bc86818f430ab1a76.jpg
28534d6fd47524968eaf605851059f03.jpg
282dbc8d999de380cc7722c4c06846c0.jpg
2841b2abfc013d9772ee4d76a5b5145f.jpg
28407e3b0f509e02537354850c9141f2.jpg
2842155bec3df934718b1a47b3d08369.jpg
2858a721e5e3763e904596320c85d49b.jpg
2879f482574a0d837663ba11b86cfc92.jpg
2864365735411042a4f9b403ef9c469f.jpg
286ede927b58286f5eabe98f6f3df922.jpg
2864d5dc8463ffa650565241bfecfd1f.jpg
```

```
03a0244b0abc5fd23dba296c4cfcd070.jpg
039b7194bfc5be44d0e7b4c092ff5961.jpg
038cf1d2ffaa2bdfec0802f21bfb9819.jpg
0393e58027891808f86a4ea9db8c4168.jpg
0396f5762aedc849678a585b43352850.jpg
03952dc36e0e61eb6f25442414bc806d.jpg
038024158016c18e27637a6cc8b7fdfc.jpg
038b836a073982e014a1f52ecfcf3151.jpg
0370e0820821ef97edbc138f5663dfbc.jpg
03a9841aa70fbcdd588b19e6622b3f36.jpg
0393d455e6b0f90e9754969e088c760d.jpg
037b91143e8b57a8ea94bf5a27432eec.jpg
03a282f0dabd1c6ab169695af7d54831.jpg
036ed3ec877143184ae9b7ea4b091600.jpg
03c167867de0a6834ef8769baaf36e1a.jpg
03c1a0502eaf756ebcf6a35a61d604e0.jpg
03de765fd1059b0cb7472631e4a0579c.jpg
03b6c99cdc8e0bb3324a11290a43ae83.jpg
03b86209a6c8c0c0a5719d308605fdca.jpg
03c9aa4d005cf31d145434f4095ac5f9.jpg
03b4341b88dd97bbe7f7b44fabfab30b.jpg
03c994ef1e7d3f5449e3975d1f5ed231.jpg
03ba8b8f7a3781acc61788dc557af1d3.jpg
03c4f8e9669d31b6e129806dae1a7402.jpg
03cee0099893beece6986a183bdc3342.jpg
03bf33a0bba3db4c15e0d561464460dc.jpg
03b60b788c6f02295d43cd0d36606185.jpg
03d42a05a9bf2bd027ddea39e0120d8d.jpg
03dacaebb7f13423c30911c4f01bde7d.jpg
03ae04bd14113279190085fee52c37d0.jpg
03d976855b24305a41fa0f2bd0240001.jpg
03c9c3973cd4b4f3142af9cb486cfdd0.jpg
03d7dd4a957a473ac67438d17fe264cb.jpg
03b82e7438b9c5547fdc9ab220852c8e.jpg
03d3c97610540792cb5e53889160cd29.jpg
03ba78002ef0260e2d88a627152fbec4.jpg
03bb7b2cad26bee8998341d4ce2e4824.jpg
03ea776cf40f6fe06aafae43b1bcaccb.jpg
03fa79210b60eff815326ef31325adea.jpg
03fd0a5413c03a05e43aa8b78d759c66.jpg
03e5678583225519d424edb59bc92f67.jpg
03eb9a2871862b37dfa375d3f830655a.jpg
03f0c235b2e56a460e23587c4d4c3ad0.jpg
03f44ae0d5f1938ceb30ac6a2f609c1d.jpg
040481c9a5c57a613fab2710882ddc8f.jpg
03ffd2c162cdd40ed3816e3a8a0e8b72.jpg
03eddc8a882614f8aa7a0f212f98c72e.jpg
03e027656171420583db686600b59cf8.jpg
03f94cf522101e71933eb4047c5091fe.jpg
03ef8d1cd5804c0a1ebc7632f058d7f2.jpg
04056b0b275ab9774c4a2d4a77c4c982.jpg
0404c41acf973bb9c4d19ef1dd569658.jpg
040717829d61e53c6274b476bc367b28.jpg
03f3b26cbe5419d68cb4eb8b06938741.jpg
03f0c92e2f79f3054547aab785bfb185.jpg
03e1c0f4227b523cfdfb9205f240ca39.jpg
03ed1de3688c83c139aef39088aad083.jpg
0403ae0b674415bfc675483e7e5a1e0b.jpg
03ebb65008cdbe2b553e97e1992cf502.jpg
03fe7e408acc2ef838ac3899897c2387.jpg
040cdc5eb85306d99f25ba70cb7b0f9a.jpg
042a301f7a13f1476692926360663fb5.jpg
042bd9e1c544b0bf5657b4d47bd2eede.jpg
0418de923fec638a4f8f2033959963e5.jpg
```

```
0426c8d10556e748192b4df3a223c0fe.jpg
0426f873aad38961ecbd4d1da27f6b28.jpg
04164dcca25716b0a3d0dc2d44e4b900.jpg
04325d767af190517379650dacd02493.jpg
04198ddfc9091f7e8f467a75db3eab29.jpg
041eb2a38f0c7062237cedbc8e4565cd.jpg
0433a8e9bee270cdf66f8dfa8c4ba977.jpg
0411784efaf3e9fa1d97a1fd07fbe79c.jpg
041c4ed0dfca106e6f3c76ce705db152.jpg
```

In [ ]:
```
COMPUTE_CV = True
if len(pd.read_csv(extract_path + '/test.csv')) > 3: COMPUTE_CV = False
```

In [ ]:
```
if COMPUTE_CV:
    dataset = pd.read_csv(extract_path + '/train.csv')
    tmp = dataset.groupby('label_group').posting_id.agg('unique').to_dict()
    dataset['target'] = dataset.label_group.map(tmp)
else:
    dataset = pd.read_csv(extract_path + '/test.csv')
```

In [ ]:
```
dataset.head()
```

Out[ ]:

	posting_id	image	image_phash	title	label_g
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	24911
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO...	293798
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	239590
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp...	409321
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	364893

In [ ]:
```
def show_random_img():
    # choose randomly two instances per each class
    labels_to_show = np.random.choice(dataset.label_group.unique(),
                                      replace=False, size=24)
    #PATH_TO_IMG=extract_path+"/train_images"
    img_to_show = []
    for label in labels_to_show:
        rows = dataset[dataset.label_group==label].copy()
        pair = np.random.choice([i for i in range(len(rows))],
                                    replace=False, size=2)
        img_pair = rows.iloc[pair][['image', 'title']].values
```

```
        img_to_show += list(img_pair)

    fig, axes = plt.subplots(figsize = (18, 12), nrows=4,ncols=6)
    for imp, ax in zip(img_to_show, axes.ravel()):
        img = cv2.imread("/content/drive/MyDrive/path/shopee-product-matching/trai
        if img is not None:
          #print(img)
          title = '\n'.join(wrap(imp[1], 20))
          ax.set_title(title)
          ax.imshow(img)
          ax.axis('off')
    fig.tight_layout()
```

In [ ]:
```
if COMPUTE_CV:
    show_random_img()
```

ResNet50 Block

In [ ]:
```
class ResNetEmbedder(nn.Module):

    def __init__(self, device='cpu'):
        super(ResNetEmbedder, self).__init__()
        self.model = models.resnet50(pretrained=False)
        self.device = device
        path = '/content/drive/MyDrive/path/shopee-product-matching/resnet50-19c8e
        self.model.load_state_dict(torch.load(path))
#         to freeze weights
        for param in self.model.parameters():
                param.requires_grad = False
        self.model.to(device)


    def transform(self, img):
        image_transform = torchvision.transforms.Compose(
```

```
            [
                torchvision.transforms.Resize(256),
                transforms.CenterCrop(224),
                torchvision.transforms.ToTensor(),
                torchvision.transforms.Normalize(
                    mean=(0.485, 0.456, 0.406),
                    std=(0.229, 0.224, 0.225)
                ),
            ]
        )
        return image_transform(img)

    def forward(self, img):
        img_tr = self.transform(img).unsqueeze(0)
        img_tr = img_tr.to(self.device)
        features = self.model(img_tr).squeeze()
        return features
```

In [ ]:
```
model_img = ResNetEmbedder(device)
```

```
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:208: UserWarn
ing: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
future, please use 'weights' instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:223: UserWarn
ing: Arguments other than a weight enum or `None` for 'weights' are deprecated sin
ce 0.13 and may be removed in the future. The current behavior is equivalent to pa
ssing `weights=None`.
  warnings.warn(msg)
```

In [ ]:
```
model_img
```

Out[ ]:

```
ResNetEmbedder(
  (model): ResNet(
    (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias
=False)
    (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_sta
ts=True)
    (relu): ReLU(inplace=True)
    (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode
=False)
    (layer1): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer2): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
```

```
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer3): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
```

```
          (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (4): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (5): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
```

```
g_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer4): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
    (fc): Linear(in_features=2048, out_features=1000, bias=True)
  )
)
```

In [ ]:

```python
def vectorize_img(img_path):
    img = Image.open(img_path).convert('RGB')
    model_img.eval()
    with torch.no_grad():
```

```
        output = model_img(img).cpu().numpy()
    return output
```

In [ ]:
```
#pip install pretrainedmodels
```

In [ ]:
```
#import pretrainedmodels
#print(pretrainedmodels.model_names)
#print(pretrainedmodels.pretrained_settings['resnet50'])
#resnet50(weights=ResNet50_Weights.IMAGENET1K_V2)
```

In [ ]:
```
%%time
if COMPUTE_CV:
    dataset['resnet_v'] = dataset['image'].progress_apply(lambda x: vectorize_img(
else:
    dataset['resnet_v'] = dataset['image'].progress_apply(lambda x: vectorize_img(
```

```
100%|██████████| 34250/34250 [1:57:02<00:00,  4.88it/s]
CPU times: user 13min 28s, sys: 23.2 s, total: 13min 51s
Wall time: 1h 57min 2s
```

In [ ]:
```
del model_img
```

In [ ]:
```
vectors = np.stack(dataset.resnet_v)
vectors = torch.Tensor(vectors).to(device)
vectors = F.normalize(vectors)
```

In [ ]:
```
preds = []
CHUNK = 1024

print('Finding similar titles...')
CTS = len(dataset)//CHUNK
if len(dataset)%CHUNK!=0: CTS += 1
for j in range( CTS ):

    a = j*CHUNK
    b = (j+1)*CHUNK
    b = min(b,len(dataset))
    print('chunk',a,'to',b)

    # COSINE SIMILARITY DISTANCE
    cts = torch.matmul( vectors, vectors[a:b].T).T
    cts = cts.cpu().numpy()

    for k in range(b-a):
        IDX = np.where(cts[k,]>0.9)[0]
        o = dataset.iloc[IDX].posting_id.values
        preds.append(o)

del vectors, cts, IDX, o
_ = gc.collect()
```

```
Finding similar titles...
chunk 0 to 1024
chunk 1024 to 2048
chunk 2048 to 3072
chunk 3072 to 4096
chunk 4096 to 5120
chunk 5120 to 6144
chunk 6144 to 7168
chunk 7168 to 8192
chunk 8192 to 9216
chunk 9216 to 10240
chunk 10240 to 11264
chunk 11264 to 12288
chunk 12288 to 13312
chunk 13312 to 14336
chunk 14336 to 15360
chunk 15360 to 16384
chunk 16384 to 17408
chunk 17408 to 18432
chunk 18432 to 19456
chunk 19456 to 20480
chunk 20480 to 21504
chunk 21504 to 22528
chunk 22528 to 23552
chunk 23552 to 24576
chunk 24576 to 25600
chunk 25600 to 26624
chunk 26624 to 27648
chunk 27648 to 28672
chunk 28672 to 29696
chunk 29696 to 30720
chunk 30720 to 31744
chunk 31744 to 32768
chunk 32768 to 33792
chunk 33792 to 34250
```

In [ ]:

```
dataset['preds_resnet'] = preds
dataset.head()
```

Out[ ]:

	posting_id	image	image_phash	title	label_g
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	24911
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO...	293798
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	239590
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp...	409321
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	364893

In [ ]:
```
def getMetric(col):
    def f1score(row):
        n = len( np.intersect1d(row.target,row[col]) )
        return 2*n / (len(row.target)+len(row[col]))
    return f1score
```

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_resnet'] = dataset.apply(getMetric('preds_resnet'), axis=1)
    print('CV score for baseline =', dataset.f1_resnet.mean())
```

```
CV score for baseline = 0.6251503234406631
```

In [ ]:
```
training_dataset= extract_path + '/train.csv'
```

E-Bert for e-commerce

In [ ]:
```
class BERTEmbedder(nn.Module):

    def __init__(self, device='cpu'):
        super(BERTEmbedder, self).__init__()
        self.bert_path = "/content/drive/MyDrive/path/shopee-product-matching/sente
        self.model = BertModel.from_pretrained(self.bert_path)
#         to freeze weights
        for param in self.model.parameters():
                param.requires_grad = False
        self.model.to(device)

    def transform(self, txt):
        tokenizer = BertTokenizer.from_pretrained(self.bert_path)
```

```
        encoded_input  = tokenizer.encode_plus( txt,
                                                 truncation=True,
                                                 max_length=128,
                                                 add_special_tokens=True,
                                                 padding=True,
                                                 return_tensors='pt').values()

        return encoded_input

    def mean_pooling(self, model_output, attention_mask):
        token_embeddings = model_output[0]
        input_mask_expanded = attention_mask.unsqueeze(-1).expand(token_embeddings
        sum_embeddings = torch.sum(token_embeddings * input_mask_expanded, 1)
        sum_mask = torch.clamp(input_mask_expanded.sum(1), min=1e-9)
        return sum_embeddings / sum_mask

    def forward(self, txt):
        inputs_ids, token_type_ids, attention_mask = self.transform(txt)
        inputs_ids, token_type_ids, attention_mask = inputs_ids.to(device), \
                                                 token_type_ids.to(device), attentic
        with torch.no_grad():
            encoded_layers = self.model(inputs_ids,
                                        attention_mask=attention_mask,
                                        token_type_ids=token_type_ids)
        features = self.mean_pooling(encoded_layers, attention_mask)
        return features
```

In [ ]:
```
model_txt = BERTEmbedder(device)
```

In [ ]:
```
model_txt
```

Out[ ]:
```
BERTEmbedder(
  (model): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
)
```

In [ ]:
```
def vectorize_txt(txt):
    model_txt.eval()
    with torch.no_grad():
        output = model_txt(txt).cpu().numpy()
    return output
```

In [ ]:
```
%%time
dataset['sbert_v'] = dataset['title'].progress_apply(lambda x: vectorize_txt(x))
```

```
100%|██████████| 34250/34250 [38:25<00:00, 14.86it/s]
CPU times: user 32min 33s, sys: 47.1 s, total: 33min 20s
Wall time: 38min 25s
```

In [ ]:
```
#bert_path = "/content/drive/MyDrive/path/shopee-product-matching/sentence-transfor
#tokenizer = BertTokenizer.from_pretrained(bert_path)
#token_lens = []
#for txt in training_dataset.proc_title:
          #tokens = tokenizer.encode(txt, max_length=512)
```

```
        #token_lens.append(len(tokens))
#sns.distplot(token_lens)
```

In [ ]:
```
del model_txt
```

In [ ]:
```
vectors = np.stack(dataset.sbert_v).squeeze(1)
vectors = torch.Tensor(vectors).to(device)
vectors = F.normalize(vectors)
```

In [ ]:
```
preds = []
CHUNK = 1024

print('Finding similar titles...')
CTS = len(dataset)//CHUNK
if len(dataset)%CHUNK!=0: CTS += 1
for j in range( CTS ):

    a = j*CHUNK
    b = (j+1)*CHUNK
    b = min(b,len(dataset))
    print('chunk',a,'to',b)

    # COSINE SIMILARITY DISTANCE
    cts = torch.matmul( vectors, vectors[a:b].T).T
    cts = cts.cpu().numpy()

    for k in range(b-a):
        IDX = np.where(cts[k,]>0.95)[0]
        o = dataset.iloc[IDX].posting_id.values
        preds.append(o)

del vectors, cts, IDX, o
_ = gc.collect()
```

```
Finding similar titles...
chunk 0 to 1024
chunk 1024 to 2048
chunk 2048 to 3072
chunk 3072 to 4096
chunk 4096 to 5120
chunk 5120 to 6144
chunk 6144 to 7168
chunk 7168 to 8192
chunk 8192 to 9216
chunk 9216 to 10240
chunk 10240 to 11264
chunk 11264 to 12288
chunk 12288 to 13312
chunk 13312 to 14336
chunk 14336 to 15360
chunk 15360 to 16384
chunk 16384 to 17408
chunk 17408 to 18432
chunk 18432 to 19456
chunk 19456 to 20480
chunk 20480 to 21504
chunk 21504 to 22528
chunk 22528 to 23552
chunk 23552 to 24576
chunk 24576 to 25600
chunk 25600 to 26624
chunk 26624 to 27648
chunk 27648 to 28672
chunk 28672 to 29696
chunk 29696 to 30720
chunk 30720 to 31744
chunk 31744 to 32768
chunk 32768 to 33792
chunk 33792 to 34250
```

In [ ]:

```
dataset['preds_sbert'] = preds
dataset.head()
```

Out[ ]:

	posting_id	image	image_phash	title	label_g
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	24911
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO...	293798
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	239590
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp...	409321
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	364893

In [ ]:
```
del preds
```

Evaluating the accuracy of BERT using Cross Validation(CV) Score

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_sbert'] = dataset.apply(getMetric('preds_sbert'), axis=1)
    print('CV score for baseline =', dataset.f1_sbert.mean())
```

```
CV score for baseline = 0.5458147145334143
```

Concatenation - ResNet 50 + BERT

In [ ]:
```
def concat():
    def cat(row):
        comm = np.concatenate([row.resnet_v,row.sbert_v.squeeze()])
        return comm
    return cat
```

In [ ]:
```
dataset['concat_v'] = dataset.progress_apply(concat(), axis=1)
```

```
100%|██████████| 34250/34250 [00:01<00:00, 29039.62it/s]
```

In [ ]:
```
vectors = np.stack(dataset.concat_v)
```

In [ ]:
```
KNN = 50
model = NearestNeighbors(n_neighbors=KNN)
model.fit(vectors)
```

Out[ ]:
```
NearestNeighbors()
```

In [ ]:
```
preds = []
CHUNK = 1024*4

print('Finding similar images...')
CTS = len(vectors)//CHUNK
if len(vectors)%CHUNK!=0: CTS += 1
for j in range( CTS ):

    a = j*CHUNK
    b = (j+1)*CHUNK
    b = min(b,len(vectors))
    print('chunk',a,'to',b)
    distances, indices = model.kneighbors(vectors[a:b,])

    for k in range(b-a):
        IDX = np.where(distances[k,]<35.0)[0]
        IDS = indices[k,IDX]
        o = dataset.iloc[IDS].posting_id.values
        preds.append(o)

del model, distances, indices, vectors, IDX, o, IDS
_ = gc.collect()
```

```
Finding similar images...
chunk 0 to 4096
chunk 4096 to 8192
chunk 8192 to 12288
chunk 12288 to 16384
chunk 16384 to 20480
chunk 20480 to 24576
chunk 24576 to 28672
chunk 28672 to 32768
chunk 32768 to 34250
```

In [ ]:
```
dataset['preds_concat'] = preds
dataset.head()
```

Out[ ]:

	posting_id	image	image_phash	title	label_g
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	24911
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO...	293798
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	239590
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp...	409321
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	364893

In [ ]:
```
del preds
```

Evaluating the accuracy of concatenated models (ResNet50 + e- BERT) using Cross Validation(CV) Score

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_concat'] = dataset.apply(getMetric('preds_concat'), axis=1)
    print('CV score for baseline =', dataset.f1_concat.mean())
```

```
CV score for baseline = 0.6486006610184477
```

We could observe that the combination has increased the CV score by 10 % from BERT and 2 % improvement from Standalone ResNet 50

Phash block

In [ ]:
```
tmp = dataset.groupby('image_phash').posting_id.agg('unique').to_dict()
dataset['preds_phash'] = dataset.image_phash.map(tmp)
dataset.head()
```

Out[ ]:

	posting_id	image	image_phash	title	label_g
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	24911
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO...	293798
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	239590
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp...	409321
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	364893

In [ ]:
```
del tmp
```

TF-IDF block for comparision against effectiveness with BERT

In [ ]:
```
dataset_gf = cudf.DataFrame(dataset[['posting_id', 'title']])
```

In [ ]:
```
model = TfidfVectorizer(stop_words='english', binary=True, max_features=25_000)
text_embeddings = model.fit_transform(dataset_gf.title)
```

In [ ]:
```
model
```

Out[ ]:
```
<cuml.feature_extraction._tfidf_vectorizer.TfidfVectorizer at 0x7b266e60f430>
```

In [ ]:
```
del model
```

In [ ]:
```
preds = []
CHUNK = 1024

print('Finding similar titles...')
CTS = len(dataset)//CHUNK
if len(dataset)%CHUNK!=0: CTS += 1
for j in range( CTS ):

    a = j*CHUNK
    b = (j+1)*CHUNK
    b = min(b,len(dataset))
```

```
    print('chunk',a,'to',b)

    # COSINE SIMILARITY DISTANCE
    cts = text_embeddings.dot(text_embeddings[a:b].T).T.toarray()

    for k in range(b-a):
        IDX = cupy.where(cts[k,]>0.7)[0]
        o = dataset.iloc[cupy.asnumpy(IDX)].posting_id.values
        preds.append(o)

del text_embeddings, IDX, o, cts
_ = gc.collect()
```

```
Finding similar titles...
chunk 0 to 1024
chunk 1024 to 2048
chunk 2048 to 3072
chunk 3072 to 4096
chunk 4096 to 5120
chunk 5120 to 6144
chunk 6144 to 7168
chunk 7168 to 8192
chunk 8192 to 9216
chunk 9216 to 10240
chunk 10240 to 11264
chunk 11264 to 12288
chunk 12288 to 13312
chunk 13312 to 14336
chunk 14336 to 15360
chunk 15360 to 16384
chunk 16384 to 17408
chunk 17408 to 18432
chunk 18432 to 19456
chunk 19456 to 20480
chunk 20480 to 21504
chunk 21504 to 22528
chunk 22528 to 23552
chunk 23552 to 24576
chunk 24576 to 25600
chunk 25600 to 26624
chunk 26624 to 27648
chunk 27648 to 28672
chunk 28672 to 29696
chunk 29696 to 30720
chunk 30720 to 31744
chunk 31744 to 32768
chunk 32768 to 33792
chunk 33792 to 34250
```

In [ ]:
```
dataset['preds_tfidf'] = preds
```

In [ ]:
```
del preds
```

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_tfidf'] = dataset.apply(getMetric('preds_tfidf'), axis=1)
    print('CV score for baseline =', dataset.f1_tfidf.mean())
```

```
CV score for baseline = 0.6139718474362906
```

The individual performance of the TF-IDF on the given dataset is only (CV Score =

0.6139). which is considerably improved about 7% greater than e-BERT (CV Score = 0.5458)

Submission block - Comparing and performing test with TF - IDF for validation

In [ ]:
```
def combine_for_sub(row):
    x = np.concatenate([row.preds_concat,row.preds_phash, row.preds_tfidf])
    return ' '.join( np.unique(x) )

def combine_for_train(row):
    x = np.concatenate([row.preds_concat,row.preds_phash, row.preds_tfidf])
    return list(np.unique(x))
```

In [ ]:
```
if COMPUTE_CV:
    dataset['matches'] = dataset.apply(combine_for_train, axis=1)
else:
    dataset['matches'] = dataset.apply(combine_for_sub, axis=1)
```

In [ ]:
```
dataset.to_pickle('train_data.pkl')
```

In [ ]:
```
dataset[['posting_id', 'matches']].to_csv('submission.csv',index=False)
```

In [ ]:
```
subm = pd.read_csv('submission.csv')
subm.head()
```

Out[ ]:

	posting_id	matches
0	train_129225211	['train_129225211', 'train_2278313361']
1	train_3386243561	['train_1816968361', 'train_2120597446', 'trai...
2	train_2288590299	['train_2288590299']
3	train_2406599165	['train_1508100548', 'train_1744956981', 'trai...
4	train_3369186413	['train_3369186413', 'train_921438619']

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_final'] = dataset.apply(getMetric('matches'), axis=1)
    print('CV score for baseline =', dataset.f1_final.mean())
```

```
CV score for baseline = 0.7178046822803029
```

Thereby the combined usage of **ResNet50 + eBERT + TF-IDF** has increased the performance of the product match classification to over 10 % precisely 0.717804

Resnet50 V2

In [ ]:
```
import locale
locale.setlocale(locale.LC_ALL, 'en_US.utf-8')
```

Out[ ]:
```
'en_US.utf-8'
```

In [ ]:
```
#pip install scikit-plot
```

In [ ]:
```
import random
import os
import glob
import time

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import tensorflow as tf
import tensorflow_hub as hub
from tensorflow.keras import layers, Sequential
from tensorflow.keras.utils import plot_model

from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, precision_recall_fscore_support
from sklearn.metrics import accuracy_score, f1_score, matthews_corrcoef
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
#from scikitplot.metrics import plot_roc
```

In [ ]:
```
class CFG:
    EPOCHS = 10
    BATCH_SIZE = 32
    SEED = 42
    TF_SEED = 768
    HEIGHT = 224
    WIDTH = 224
    CHANNELS = 3
    IMAGE_SIZE = (224, 224, 3)
```

Dataset

In [ ]:
```
# Define paths
DATASET_PATH = "/content/drive/MyDrive/path/shopee-product-matching/"
TRAIN_PATH = '/content/drive/MyDrive/path/shopee-product-matching/train_images'
VAL_PATH = '/content/drive/MyDrive/path/shopee-product-matching/train_images'
TEST_PATH = '/content/drive/MyDrive/path/shopee-product-matching/test_images'
```

In [ ]:
```
# Generate a summary of the dataset
print('DATASET SUMMARY')
print('=======================\n')
for dirpath, dirnames, filenames in os.walk(DATASET_PATH):
    print(f"There are {len(dirnames)} directories and {len(filenames)} images in {
print('\n=======================')
```

```
DATASET SUMMARY
========================

There are 3 directories and 6 images in /content/drive/MyDrive/path/shopee-product
-matching/
There are 0 directories and 3 images in /content/drive/MyDrive/path/shopee-product
-matching/test_images
There are 0 directories and 32492 images in /content/drive/MyDrive/path/shopee-pro
duct-matching/train_images
There are 0 directories and 5 images in /content/drive/MyDrive/path/shopee-product
-matching/sentence-transformer

========================
```

In [ ]:
```
%%time
train_images = glob.glob(f"{TRAIN_PATH}**/*.jpg")
val_images = glob.glob(f"{VAL_PATH}**/*.jpg")
test_images = glob.glob(f"{TEST_PATH}**/*.jpg")
```

```
CPU times: user 128 ms, sys: 17.8 ms, total: 146 ms
Wall time: 1.23 s
```

In [ ]:
```
train_size = len(train_images)
val_size = len(val_images)
test_size = len(test_images)
total = train_size + val_size + test_size

# View the counts
print(f'train samples count:\t\t{train_size}')
print(f'dev/validation samples count:\t{val_size}')
print(f'test samples count:\t\t{test_size}')
print('=======================================')
print(f'TOTAL:\t\t\t\t{total}')
```

```
train samples count:            32492
dev/validation samples count:   32492
test samples count:             3
=======================================
TOTAL:                          64987
```

In [ ]:
```
def generate_labels(image_paths):
    labels = []
    for _ in image_paths:
        if ('PNEUMONIA' in _.replace('chest-xray-pneumonia', '')):
            labels.append('PNEUMONIA')
        else:
            labels.append('NORMAL')

    return labels

def build_df(image_paths, labels):
    df = pd.DataFrame({
        'image_path': image_paths,
        'label': generate_labels(labels)
    })
    df['label_encoded'] = df.apply(lambda row: 0 if row.label == 'NORMAL' else 1, a

    return df.sample(frac=1, random_state=CFG.SEED).reset_index()
```

In [ ]:
```
# Build the DataFrames
train_df = build_df(train_images, generate_labels(train_images))
val_df = build_df(val_images, generate_labels(val_images))
test_df = build_df(test_images, generate_labels(test_images))
```

In [ ]:
```
# Have a look at val_df
val_df
```

Out[ ]:

	index	image_path	label	label_encoded
0	9999	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
1	2231	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
2	24803	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
3	30136	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
4	4410	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
...	...	...	...	...
32487	29802	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
32488	5390	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
32489	860	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
32490	15795	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0
32491	23654	/content/drive/MyDrive/path/shopee-product-mat...	NORMAL	0

32492 rows × 4 columns

In [ ]:
```
def _load(image_path):
    # Read and decode an image file to a uint8 tensor
    image = tf.io.read_file(image_path)
    image = tf.io.decode_jpeg(image, channels=3)

    # Resize image
    image = tf.image.resize(image, [CFG.HEIGHT, CFG.WIDTH],
                            method=tf.image.ResizeMethod.LANCZOS3)

    # Convert image dtype to float32 and NORMALIZE!!!
    image = tf.cast(image, tf.float32)/255.

    # Return image
    return image

def view_sample(image, label, color_map='gray', fig_size=(8, 10)):
    plt.figure(figsize=fig_size)
    plt.imshow(image, cmap=color_map)
    plt.title(f'Label: {label}', fontsize=16)
    return
```

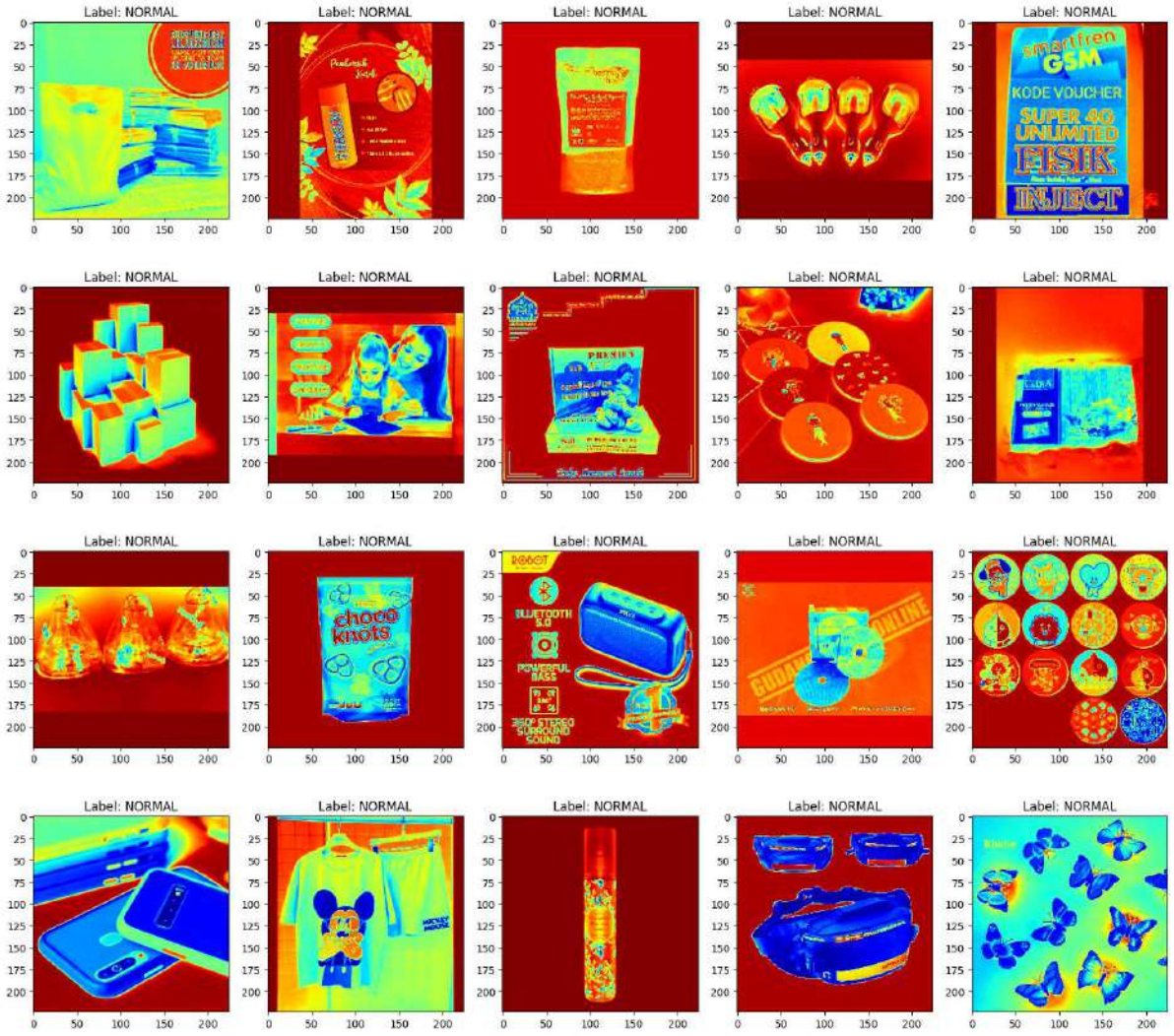
Displaying Sample Images

In [ ]:
```
# Select random sample from train_df
idx = random.sample(train_df.index.to_list(), 1)[0]

# Load the random sample and label
sample_image, sample_label = _load(train_df.image_path[idx]), train_df.label[idx]

# View the random sample
view_sample(sample_image, sample_label)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RG
B data ([0..1] for floats or [0..255] for integers).
```

In [ ]:

```
def view_mulitiple_samples(df, sample_loader, count=10, color_map='gray', fig_size=
    rows = count//5
    if count%5 > 0:
        rows +=1

    idx = random.sample(df.index.to_list(), count)
    fig = plt.figure(figsize=fig_size)

    for column, _ in enumerate(idx):
        plt.subplot(rows, 5, column+1)
        plt.title(f'Label: {df.label[_]}')
        plt.imshow(tf.image.rgb_to_grayscale(sample_loader(df.image_path[_])), cmap

    return

view_mulitiple_samples(train_df, _load,
                       count=20, color_map='jet',
                       fig_size=(20, 18))
```



In [ ]:

```
class ResNetEmbedder_1(nn.Module):

    def __init__(self, device='cpu'):
        super(ResNetEmbedder_1, self).__init__()
        self.model = models.wide_resnet50_2(pretrained=False)
        self.device = device
        path = '/content/drive/MyDrive/path/shopee-product-matching/wide_resnet50_2
        self.model.load_state_dict(torch.load(path))
#         to freeze weights
        for param in self.model.parameters():
                param.requires_grad = False
        self.model.to(device)


    def transform(self, img):
        image_transform = torchvision.transforms.Compose(
            [
                torchvision.transforms.Resize(256),
                transforms.CenterCrop(224),
                torchvision.transforms.ToTensor(),
                torchvision.transforms.Normalize(
                    mean=(0.485, 0.456, 0.406),
                    std=(0.229, 0.224, 0.225)
                ),
            ]
        )
        return image_transform(img)

    def forward(self, img):
        img_tr = self.transform(img).unsqueeze(0)
```

```
        img_tr = img_tr.to(self.device)
        features = self.model(img_tr).squeeze()
        return features
```

In [ ]:
```
model_img_r50v2 = ResNetEmbedder_1(device)
```

```
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:208: UserWarn
ing: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
future, please use 'weights' instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:223: UserWarn
ing: Arguments other than a weight enum or `None` for 'weights' are deprecated sin
ce 0.13 and may be removed in the future. The current behavior is equivalent to pa
ssing `weights=None`.
  warnings.warn(msg)
```

In [ ]:
```
model_img_r50v2
```

Out[ ]:

```
ResNetEmbedder_1(
  (model): ResNet(
    (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias
=False)
    (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_sta
ts=True)
    (relu): ReLU(inplace=True)
    (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode
=False)
    (layer1): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer2): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
```

```
        (conv3): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer3): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
```

```
          (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (4): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv3): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (5): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
g_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_runnin
```

```
g_stats=True)
        (conv3): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer4): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(1024, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (conv2): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (conv3): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(2048, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (conv2): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (conv3): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(2048, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (conv2): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (conv3): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_runni
ng_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
    (fc): Linear(in_features=2048, out_features=1000, bias=True)
  )
)
```

In [ ]:

```
def vectorize_img_r50v2(img_path):
    img = Image.open(img_path).convert('RGB')
    model_img_r50v2.eval()
    with torch.no_grad():
```

```
        output = model_img_r50v2(img).cpu().numpy()
    return output
```

In [ ]:
```
%%time
if COMPUTE_CV:
    dataset['resnet_v2'] = dataset['image'].progress_apply(lambda x: vectorize_img
else:
    dataset['resnet_v2'] = dataset['image'].progress_apply(lambda x: vectorize_img
```

```
100%|██████████| 34250/34250 [14:59<00:00, 38.06it/s]
CPU times: user 12min 18s, sys: 10.4 s, total: 12min 28s
Wall time: 14min 59s
```

In [ ]:
```
del model_img_r50v2
```

In [ ]:
```
vectors1 = np.stack(dataset.resnet_v2)
vectors1 = torch.Tensor(vectors1).to(device)
vectors1 = F.normalize(vectors1)
```

In [ ]:
```
preds_rv2 = []
CHUNK = 1024

print('Finding similar titles...')
CTS = len(dataset)//CHUNK
if len(dataset)%CHUNK!=0: CTS += 1
for j in range( CTS ):

    a = j*CHUNK
    b = (j+1)*CHUNK
    b = min(b,len(dataset))
    print('chunk',a,'to',b)

    # COSINE SIMILARITY DISTANCE
    cts = torch.matmul( vectors1, vectors1[a:b].T).T
    cts = cts.cpu().numpy()

    for k in range(b-a):
        IDX = np.where(cts[k,]>0.9)[0]
        o = dataset.iloc[IDX].posting_id.values
        preds_rv2.append(o)

del vectors1, cts, IDX, o
_ = gc.collect()
```

```
Finding similar titles...
chunk 0 to 1024
chunk 1024 to 2048
chunk 2048 to 3072
chunk 3072 to 4096
chunk 4096 to 5120
chunk 5120 to 6144
chunk 6144 to 7168
chunk 7168 to 8192
chunk 8192 to 9216
chunk 9216 to 10240
chunk 10240 to 11264
chunk 11264 to 12288
chunk 12288 to 13312
chunk 13312 to 14336
chunk 14336 to 15360
chunk 15360 to 16384
chunk 16384 to 17408
chunk 17408 to 18432
chunk 18432 to 19456
chunk 19456 to 20480
chunk 20480 to 21504
chunk 21504 to 22528
chunk 22528 to 23552
chunk 23552 to 24576
chunk 24576 to 25600
chunk 25600 to 26624
chunk 26624 to 27648
chunk 27648 to 28672
chunk 28672 to 29696
chunk 29696 to 30720
chunk 30720 to 31744
chunk 31744 to 32768
chunk 32768 to 33792
chunk 33792 to 34250
```

In [ ]:

```
dataset['preds_resnet_rv2'] = preds_rv2
dataset.head()
```

Out[ ]:

	posting_id	image	image_phash	title	label_g
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	24911
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO...	293798
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	239590
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp...	409321
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	364893

5 rows × 22 columns

In [ ]:
```
del preds_rv2
```

In [ ]:
```
def getMetric(col):
    def f1score(row):
        n = len( np.intersect1d(row.target,row[col]) )
        return 2*n / (len(row.target)+len(row[col]))
    return f1score
```

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_concatRsv2'] = dataset.apply(getMetric('preds_resnet_rv2'), axis=1
    print('CV score for baseline =', dataset.f1_concatRsv2.mean())
```

```
CV score for baseline = 0.608940409337176
```

The accuracy obatined using the f1score for Resnet 50v2 is 0.6089

Concatenating E-BERT with ResNet 50 v2

In [ ]:
```
def concat_v2():
    def cat_v2(row):
        comm = np.concatenate([row.resnet_v2,row.sbert_v.squeeze()])
        return comm
    return cat_v2
```

In [ ]:
```
dataset['concat_v2'] = dataset.progress_apply(concat_v2(), axis=1)
```

```
100%|██████████| 34250/34250 [00:01<00:00, 26770.39it/s]
```

In [ ]:
```
vectors_v2 = np.stack(dataset.concat_v2)
```

In [ ]:
```
KNN = 50
modelv2 = NearestNeighbors(n_neighbors=KNN)
modelv2.fit(vectors_v2)
```

Out[ ]:
```
NearestNeighbors()
```

In [ ]:
```
predsv2 = []
CHUNK = 1024*4

print('Finding similar images...')
CTS = len(vectors_v2)//CHUNK
if len(vectors_v2)%CHUNK!=0: CTS += 1
for j in range( CTS ):

    a = j*CHUNK
    b = (j+1)*CHUNK
    b = min(b,len(vectors_v2))
    print('chunk',a,'to',b)
    distances, indices = modelv2.kneighbors(vectors_v2[a:b,])

    for k in range(b-a):
        IDX = np.where(distances[k,]<35.0)[0]
        IDS = indices[k,IDX]
        o = dataset.iloc[IDS].posting_id.values
        predsv2.append(o)

del modelv2, distances, indices, vectors_v2, IDX, o, IDS
_ = gc.collect()
```

```
Finding similar images...
chunk 0 to 4096
chunk 4096 to 8192
chunk 8192 to 12288
chunk 12288 to 16384
chunk 16384 to 20480
chunk 20480 to 24576
chunk 24576 to 28672
chunk 28672 to 32768
chunk 32768 to 34250
```

In [ ]:
```
dataset['preds_concat_v2'] = predsv2
dataset.head()
```

Out[ ]:

	posting_id	image	image_phash	title	label_g
0	train_129225211	0000a68812bc7e98c42888dfb1c07da0.jpg	94974f937d4c2433	Paper Bag Victoria Secret	24911
1	train_3386243561	00039780dfc94d01db8676fe789ecd05.jpg	af3f9460c2838f0f	Double Tape 3M VHB 12 mm x 4,5 m ORIGINAL / DO...	293798
2	train_2288590299	000a190fdd715a2a36faed16e2c65df7.jpg	b94cb00ed3e50f78	Maling TTS Canned Pork Luncheon Meat 397 gr	239590
3	train_2406599165	00117e4fc239b1b641ff08340b429633.jpg	8514fc58eafea283	Daster Batik Lengan pendek - Motif Acak / Camp...	409321
4	train_3369186413	00136d1cf4edede0203f32f05f660588.jpg	a6f319f924ad708c	Nescafe \xc3\x89clair Latte 220ml	364893

5 rows × 25 columns

In [ ]:
```
del predsv2
```

Evaluating the accuracy of concatenated models (ResNet50 v2 + e- BERT) using Cross Validation(CV) Score

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_concatv2'] = dataset.apply(getMetric('preds_concat_v2'), axis=1)
    print('CV score for baseline =', dataset.f1_concatv2.mean())
```

```
CV score for baseline = 0.6040379772731148
```

Re-iterating the submission block in an attempt to improve the performance by coombining with Resnet50 v2 + e-BERT + TF-IDF

In [ ]:
```
def combine_for_sub_v2(row):
    x = np.concatenate([row.preds_concat_v2,row.preds_phash, row.preds_tfidf])
    return ' '.join( np.unique(x) )
```

```
def combine_for_train_v2(row):
    x = np.concatenate([row.preds_concat_v2,row.preds_phash, row.preds_tfidf])
    return list(np.unique(x))
```

In [ ]:
```
if COMPUTE_CV:
    dataset['matchesv2'] = dataset.apply(combine_for_train_v2, axis=1)
else:
    dataset['matchesv2'] = dataset.apply(combine_for_sub_v2, axis=1)
```

In [ ]:
```
dataset.to_pickle('train_data.pkl')
```

In [ ]:
```
dataset[['posting_id', 'matches']].to_csv('submission.csv',index=False)
```

In [ ]:
```
subm = pd.read_csv('submission.csv')
subm.head()
```

Out[ ]:

	posting_id	matches
0	train_129225211	['train_129225211', 'train_2278313361']
1	train_3386243561	['train_1816968361', 'train_2120597446', 'trai...
2	train_2288590299	['train_2288590299']
3	train_2406599165	['train_1508100548', 'train_1744956981', 'trai...
4	train_3369186413	['train_3369186413', 'train_921438619']

In [ ]:
```
if COMPUTE_CV:
    dataset['f1_final_v2'] = dataset.apply(getMetric('matches'), axis=1)
    print('CV score for baseline =', dataset.f1_final_v2.mean())
```

```
CV score for baseline = 0.7178046822803029
```