Assignment Report –

Description: - Used the Image data by accessing the imageurl and the converting the image to 512 vectors using Pythorch and resnet model. Than to categorize the duplicate images I used cosine-distance between images and with productid I stored in the dictionary.

Step-1:

Unzipped the Data File and found it to be a 5.2GB dump file. As mentioned in the assignment I performed the operation on the **Subcategory "TOPS"**. I did it using a simple windows command **"FINDSTR",** basically it returns all the lines if the given string is formed. I discovered that TOPS category has a particular format" Apparels>Women>Western Wear>Shirts, Tops & Tunics>Tops" so I used the key word, **Tops & Tunics>Tops** to get only the tops section from all the dump**.** The complete command to get the exact TOPS category is
findstr /r /c:"string_to_search" file_to_search >> output_file

- **findstr /r /c:" Shirts, Tops & Tunics>Tops" "D:\p_data\2oq-c1r.csv">>tops_1.csv** The

5.2GB data turned less than 1GB with 500,000 rows

Step-2:

Used Pandas to load the data. As the file was large, I performed operations on sample data of 1000 images.
Using **Pytorch** and pretrained **Resnet model** I extracted the values of last layer after which the output is the image to a 512vetor array and stored w.r.t the image column. This is done by the **img_to_vec.py** code.

Step-3:

With **Scikit-Learn ML library** I found the **cosine-similarity** between the images. And with a product id as key I appended all the duplicate values into the values list.

Step-4:

Converted the dictionary into Json format and outputted in file **data_dict_output. Json**

References :

1.Vijay's talk: Taking Fashion and Lifestyle Commerce Towards SKUs Using Deep Image and Text Parsing helped me a lot to understand the ecommerce and gave me steps to tackle the problem as I had no idea about things behind ecommerce.

2. Imge_to_vec I took online help and found code on github -
https://github.com/christiansafka/img2vec/blob/master/img_to_vec.py