REPORT OF SUBMISSION

This is a problem of Duplicate Detection. So, according to me duplicate is something similar that posses the same features as original like color, texture etc.

FILES:

The submission folder consists of 1 folder and 3 files:

1. tops_1.csv: It contains the dataset of about 314k entries.
2. out.json: The desired output of the solution. Which is a dictionary with a product id as key and list of tuple(s) of duplicate product id
3. Hashing.py: The code for finding the duplicate in data.

CBIR Folder:

It contains the other approach using computer vision techniques to find the similarity between 2 images.

1. CBIR_Color.py: Colour Histogram based technique.
2. colordescriptor.py: It containes class will encapsulate all the necessary logic to extract our 3D HSV color histogram from our images.
3. CBIR_VGG.py: Deep learning based technique.
4. VGG.py: Feature extraction using VGG functions.
5. 1.jpeg and 2.jpeg: Images for testing.

TECHNIQUES:

1. Hashing based:

First, I named the dataset columns. The column “ID” can be used as primary key and “image” can be used to detect duplication.

Since, there are N images the time required to compute is: O(N).

2. Computer Vision Based:

First Download the required image to compare. You can download any two image from “image” column and compare the images for similarity.

- CBIR_Color:

This is a color histogram based technique to extract the features and then compare those extracted features using chi-square distance method.

We use the colordescriptor.py file to define our image descriptor. Which is a a 3D color histogram in the HSV color space.

- CBIR_VGG:

To compare images with better extracted features, I used VGG. Since, this pre-trained network can provide better results for comparison of image and finding duplicates.

* There are comments given in the code for better understanding.

* Many other approaches can be used for computer based techniques:

Like in CBIR we can also use:
- Texture based: Gabor filter
- Shape based: Edge histogram
- Resnet