Problem: **Duplicate Detection of Product listings**

Solution – **Step-by-Step**:

1. Extracted "**Tops**" category from the "**preferred**" dataset file.
2. Definition of a **DUPLICATE PRODUCT**:
   - Duplicate product listings are those product listings with same **PRODUCT URL ID**. Example: http://dl.flipkart.com/dl/one-femme-checkered-women-s-tunic/p/**itmekuyw3cnagc7p**?pid=TUNEKUYWFGUPJB3T .
   - Here **itmekuyw3cnagc7p** is the **PRODUCT URL ID**.
   - The image associated with that **PRODUCT URL ID** will also be same for product IDs with same **PRODUCT URL ID.**

3. This problem of duplicate product detection can be solved by measuring the similarity between images of two product listings.

4. Pipeline for Duplicate Product detection:
   - **Clothing Segmentation/Detection**: Find the bounding box co-ordinates of the clothing present in the image and segment the clothing part.Please note that this step is proposed only as part of this writeup, could not be implemented due to time constraints.
   - **Clothing Similarity using Siamese Neural Networks(One shot learning):** Given two product images, measure how similar both the images are using Siamese Neural Network.The lesser the distance between the images, the more similar hence duplicates.

5. **Clothing Similarity using Siamese Neural Networks(One shot learning):**
   - **Dataset:** There are totally 350799 product IDs in TOPS category(Track Tops, Thermal Tops and Tops).Out of them, 100915 are unique i.e. with unique **PRODUCT URL ID.** 200X200 resolution image URLs were chosen, downloaded and resized to 100x100. 80732 images for training and 20183 images for testing were used. For training, every product image is rotated randomly between -15 to +15 degrees to produce a positive sample each for every product. The final array shape is 80732 x 2 x 100 x 100 x 3 (axis 1 representing one original image and the other augmented).

   - **Siamese neural network**
     - Architecture: 2 Inputs(100x100x3) --> Conv(64) --> BatchNorm --> MaxPool --> Conv(128) --> BatchNorm --> MaxPool --> Conv(128) --> BatchNorm --> Dropout --> MaxPool --> Conv(256) --> BatchNorm --> Dropout --> FC(1024) --> Distance measure --> FC(1) sigmoid with binary cross entropy loss. Note: 512 sized embedding vectors were experimented too due to overfitting concerns.
     - The network takes two images(either positive or negative class) as input and tries to minimise/maximise the **distance between the embeddings** of both inputs.Note:- the network is trained from scratch.
     - Loss function: Binary cross entropy.

- Distance measures experimented: **Manhattan/Euclidean** distance.
- Optimizers experimented: SGD/Adam.
- Learning rate: Started with lr=0.01 and reduced it upto 2e-6.
- Batch size – 128
- Total number of epochs – 100 approx.
- Minimum loss obtained: 0.00394

- **Why Siamese neural network for Duplicate detection?**
  - The problem demands for finding similarity/distance between two inputs(pairs) and there are very few examples per class(only two in my case – One shot learning), hence Siamese neural network seems more appropriate for the task.

- **Evaluation:** Two products are considered duplicates if the sum of squared difference between their embeddings is less than or equal to a **threshold value of 2**(chosen by trial and error method, threshold with less false positives and for better generalization).
  - The embedding vector for every product(row) is computed and stored in memory.
  - Every product(row) is compared with every other product to measure its similarity.
  - When there is a new product listing, the embedding vector for the new product is compared with every other product embedding and their distances are measured.
  - Distance metric used for testing is Euclidean distance.

Distance Threshold	True Positive Rate	False Positive Rate	Accuracy
0	1.0	0.0	1.0
0.5	1.0	0.0007	0.9
1.0	1.0	0.001	0.998
2.0	1.0	0.004	0.9962
3.0	1.0	0.008	0.992
4.5	1.0	0.028	0.974
7.5	1.0	0.127	0.885
9.5	1.0	0.23	0.792

*The above accuracy is taken for randomly sampled 10000 test image pairs. Accuracy drops with slightest increase in distance threshold. The lower the distance, the lower the false positives are and higher the accuracy is.

The test.json file contains product Id as key and list of tuples containing duplicate product Ids and the probability. The test is taken for 3492 Product Ids randomly sampled from test set.

- **Observations:**
  - There are two main problems existing in this current approach:
    - Model posing and background has an impact on accuracy and can be solved using clothing segmentation/detection.
    - The siamese neural net seems to have overfit the data, hence more positive samples per class and proper hard negative samples can boost the accuracy.

6. Citations:

   - https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf
   - https://github.com/sorenbouma/keras-oneshot – source for Siamese network implementation.
   - https://github.com/davidsandberg/facenet/blob/096ed770f163957c1e56efa7feeb194773920f6e/src/facenet.py#L457