**Objective**: To build a machine learning model that can identify which set/group of images contain the same product/s.

**Approach**:

* The concept of product matching allows an organization to offer products at rates that are competitive to the same product sold by another retailer
* When dealing with images of similar products, there is a possibility that they could either depict the same item or completely different products. Retailers strive to prevent any misrepresentations or complications that may arise from confusing unrelated items
* Hence, it is important to maintain a high degree of precision i.e., whatever images we happen to group together as similar images as per the predicted results, the actual scenario should also be true.
* However, recall is also an important metric meaning, a retailer would not want to see their product(s) not listed within a cluster of images which they identify as similar.
* **Evaluation Metric**: In order to balance both KPIs, we go ahead and use the mean F1 score as the evaluation metric
  + F1 score is defined as the harmonic mean of precision and recall i.e*., 2 \* precision \* recall / (precision + recall)* and ranges between 0 to 1.
  + Submissions will be evaluated based on the mean F1 score. The mean is calculated in a sample-wise fashion, meaning that an F1 score is calculated for every predicted row, then averaged.
  + We should strive to achieve a higher score compared to the baseline which is calculated based on similar perceptual hash IDs grouped together to represent the same product. (Baseline mean F1 score = 0.553).

**Creation of text embeddings**: Below algorithms are applied on the product titles post which a threshold is set to identify similar products:

1. Tf-idf vectorizer + KNN (unsupervised): tf-idf refers to term frequency – inverse document frequency
2. Word2Vec + KNN (unsupervised) – the Skip Gram model of word2vec applied to obtain embeddings
3. Further analysis of models that can be implemented to improve evaluation metric (if time permits):
   1. BERT
   2. BERT-indo-15g: since most of the titles are in the Indonesian language (BERT-base model pre-trained with Indonesian Wikipedia and Indonesian newspapers)
   3. LSTM

**Creation of image embeddings**:

1. ResNet (pre-trained) + KNN/Cosine Similarity
2. EfficientNet + KNN/Cosine Similarity

**Combining text and image embeddings**: The embedding output of text and images will be concatenated to check whether the results can be bettered compared to independent model runs via text embeddings and image embedding