**1. Describe how you would calculate the similarity among each of the 3 combination pairs of Product A, B and C; Which ones are most similar: A-B or A-C or B-C?**

* I will take each feature into account as a set and identify the number of common traits between each product based on features and then the sum of overall common features. **Example**: features of product A and C are most similar to each other such as:

1. Type (Top)
2. Size (M)
3. Number of common category IDs (**1, 2, 4, 5**)
4. Number of common Keyword IDs (**10**, **22, 31, 120, 665**)
5. Color (blue).

So, overall common features are 5. Also, product A and C are of same type i.e., Top which users use the most to filter the content by while purchasing.

* Product **A-C** are most similar to each other.

**2. Now consider a dataset of 100K products; what technique(s)/algorithm(s) can you propose to efficiently calculate this similarity between this huge dataset every pair? Let’s assume we don’t have any limitations on the CPU processors or memory.**

* Technique I will use for computing the similarity between each pair of items is Item-Item collaborative filtering (CF). **Algorithm to be used for Item-Item CF is Cosine Similarity.**
* If we have no limitation of CPU or memory, we should use Cosine similarity instead of Jaccard because Jaccard similarity takes only **unique set of words** for each product while cosine similarity takes **total length of the vectors**.
* It has been proven that results of cosine similarity has the highest value in comparison with Jaccard similarity and the joint between Cosine and Jaccard similarity [1].

**3. Write a pseudo-code for your solution(s) and define your preferred data structures.**

* First, assume that “X” and “Y” denote two products and “S” denotes the similarity between them.
* Second, form distinct set of products and count number of common features occurring in each set to form vector using LCS (longest common sequence) and then apply Cosine similarly.

**LCS** (X, Y) =

**=**

* **Pseudo Code**:

Function ItemSimilarity(X, Y):

Y = convert words into vectors for Type, Size, Color using Word2Vec

commonSequences = LCS (X, Y)

similarity = CosineSimilarity(X, Y)

LCS (X, Y):

If (X == null or Y == null)

Return 0

Else (X.value == Y.value)

Return LCS (sum(X\*Y for X, Y in zip(X, Y))

CosineSimilarity (LCS(X, Y)):

dot\_product = numpy.dot(X, Y)  
 denominator = math.sqrt(X.dot (X)) \* math.sqrt(Y.dot (Y))  
 Return dot\_product/denominator

**References**:

[1]. Zahrotun, Lisna. (2016). Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering with Shared Nearest Neighbor Method. Computer Engineering and Applications Journal. 5. 11-18. 10.18495/comengapp.v5i1.160.