# **Question – 2 - Approach document**

**Q)** **Implement a simple approach to cluster the listings data into similar items clusters that can be used in this way: If the items A and B are part of the same cluster it is reasonable to recommend B to a user that is viewing or has viewed A. Feel free to use the category information that sellers have used to classify their listings in any way**.

A) The following steps were followed to cluster the listings provided:

1. Data wrangling
2. Clustering

- **Data wrangling:**

- Removed the *listing\_description* column - In the interest of time and saving computing power

- *listing\_price* has 8110 missing values and these are changed to *-9999*

- **Clustering:**

- Clustering is done at 2 levels

- **Level 1 Clustering:**

* Items are clustered using *latitude*, *longitude*, *category\_l2\_name\_en* and *listing\_price* features
* The categorical feature *category\_l2\_name\_en* is one-hot encoded
* 110 clusters are made - 110 is randomly chosen to decrease the computation. Ideally, we need to construct elbow curve and choose optimal number of clusters

- **Level 2 clustering:**

* The clusters which have more than median amount of items are re-clustered using hierarchal clustering.
* The features used for clustering are the titles similarity scores
* Punctuations, stop words, numbers and white spaces are removed from the titles.
* Euclidean distances are calculated for each title against every other title
* Using these distances as features, the items are clustered and 10 final clusters are obtained.

***Note:*** The second level clustering is taking a huge time to run and therefore included the results only from first level clustering. The code for second level is added in the code file as well in the Github

**Q)** **How do you evaluate the quality of your results? How does it compare to a naive approach that takes random listings from the same category?**

A) To access the quality of clusters, we have metrics like *Silhouette score* but it might not suit our purposes here. The only way to check if the clusters are making sense, is to check them real time. Quality can be further tested by A/B testing methods and checking for user activity with and without the recommendations from this algorithm.

This method is comparatively better than naïve approach at two levels:

- This method utilizes additional features like location and prices which will add a further personalization experience compared to that of a random listing from the category.

- The hierarchal clustering further helps by breaking down huge clusters into smaller clusters based on title’s similarity which may not be the case with the random listings

**Q)** **What are possible shortcomings and extensions of your implementation? How are newly listed (unseen) listings assigned to your clusters?**

A) **Possible extensions for the implementation**:

- Due to computation limitations, the features for second level of clustering were restricted to words from titles. This could be further improved by adding features from the description of the product

- Due to the sheer size of the dataset, it was not possible to ascertain, in detail, the quality of the clusters. The following things can be done which will address this caveat:

* Take a small subset of the items which is a representative of population
* Using the item’s title and description cluster the items
* Assess the quality of clusters by defining a similarity scoring mechanism.
* Optimize the clusters and scoring functions
* Once we are confident with the clustering method, we can possibly deploy this to a larger data and test it real time.

**Short comings of the method**:

- This method may not suitable for a dynamic inventory clustering and large set of items. If a new listing are brought up, we might have to re-run the entire algorithm to obtain the clusters