Final Project Report: eBay University Machine Learning Competition.

Chu-An Tsai and John L. Parrotte

CSCI 57300 Data Mining

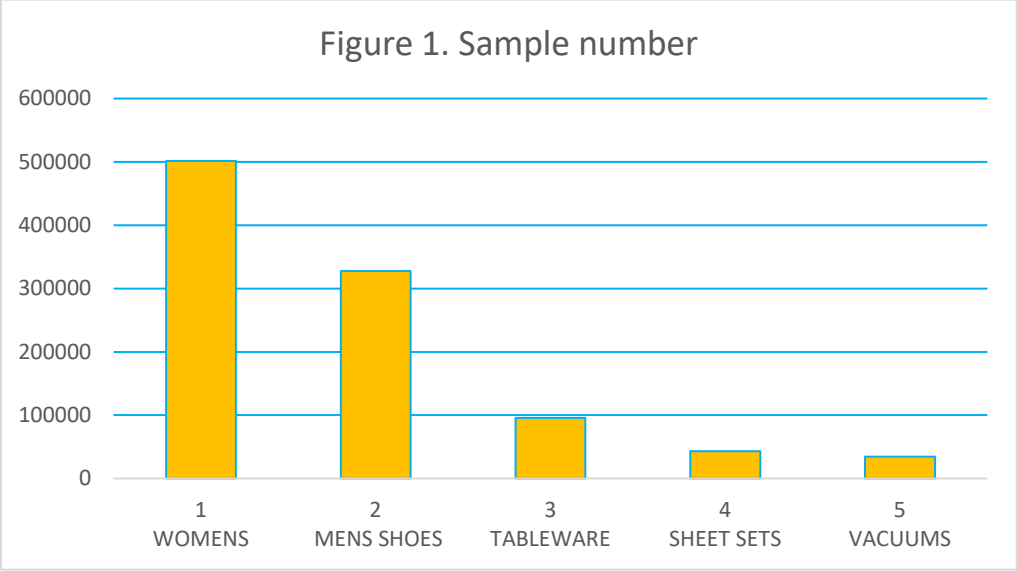
Dr. Mohammad Hasan

Introduction

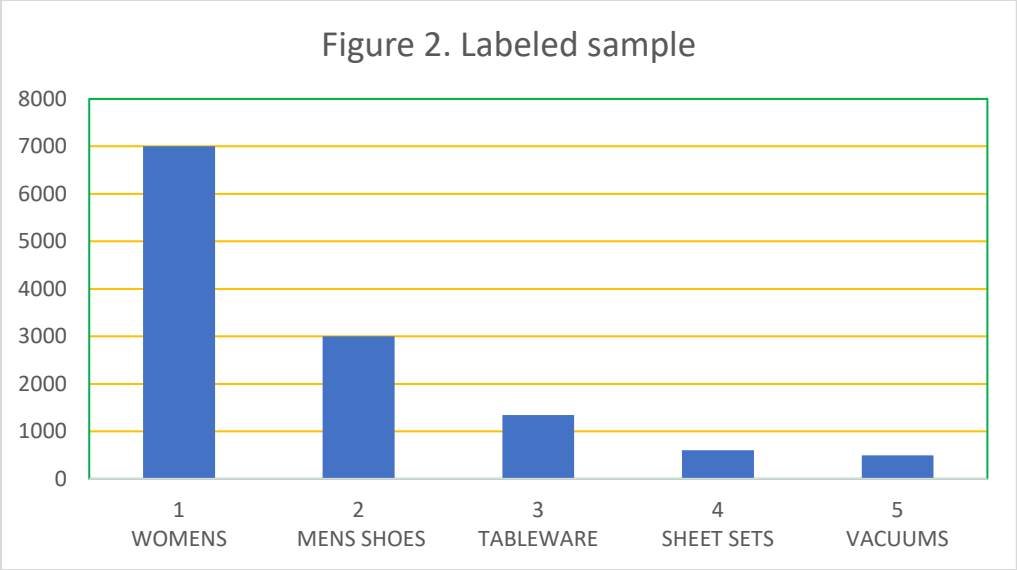
This report is based on the project of the eBay University Machine Learning Competition. As one of the largest e-commerce operations in the world, eBay deals with hundreds of millions of unstructured listings. The challenge is to correctly identify two or more listings as being for the same group, and they call it Product Level Equivalency (PLE). The PLE has a very strict requirement when comparing two products, which is that they must have the exact same specifications. Therefore, building a highly accurate clustering model is the goal of this challenge. This report introduces how we targeted title and attribute columns for data analysis and preprocessing, performed TF-IDF with TruncatedSVD for dimensionality reduction, and clustered the dataset with Hierarchical agglomerative clustering.

Data Analysis and Statistics

The provided dataset package has three parts, a 7GB challenge dataset, a validation dataset, and a PDF file. The challenge set contains unlabeled 1,002,275 samples collected by eBay from their listings, and each sample has 8 columns including category, title, subtitle, gallery_url, picture_url, attributes, description_htm_base64, and index. In our approach, we focused on titles and attributes. There are a total of 5 categories, each containing approximately 30,000 to 500,000 samples, as shown in Figure 1.



The validation dataset includes 12,443 labels and associate indexes which can trace the sample in the challenge set (Figure 2).



Method used

1. Attributes parsing

The attribute:value pair strings were dissociated into separate features with a script that removed parentheses surrounding parentheses and sometimes quotes, split by colons, further split by commas, and then recombined. An attempt was made to manually reduce the complexity of the text data through data cleansing, with a python script that changed all attribute names to lowercase, removed leading and trailing special characters and resulting white space, and stripped punctuation. In retrospect, hand writing the data cleansing script was unnecessary, as a linguistic functions Python library, NLTK (Natural Language Toolkit), was found to perform better, in less time.

Unique Attribute Names	Step
8710	Initial
7626	Lower Case
6958	Stripped punctuation, removed interior spaces
6920	NLTK in place of hand written script

Remaining attribute names were condensed to a distinct set, an integer assigned to each, and the integers put back in the place of the original attribute names. A 1,006,089 by 6,959 sparse matrix was constructed consisting of item id and a one or zero for the presence or absence

of each possible attribute. However, the matrix proved to be too large to utilize due to memory constraints. Further, the TFIDF approach ultimately used found unique features in the data.

2. TF-IDF

When dealing with text data, we first thought of the TF-IDF method, which is a widely used approach for many text-related situations such as Natural Language Processing. TF-IDF stands for Term Frequency–Inverse Document Frequency, a numerical statistic that reflects the importance of each word shown in a document or a collection of text data. It is constructed in two parts: TF and IDF. TF gives the weight of a term that occurs in the eBay data set a proportional value according to the term frequency, and IDF measures of how much information the term provides. We chose titles and attributes columns that seem to be the most representative information of the eBay data set samples and appropriate to use TF-IDF.

By passing titles and attributes into this function, the data set is transferred to a 1,002,275-row times 232,730-column numerical matrix with an associated value. The rows are the 1,002,275 samples, and the columns represent each word in the data set, such as brand, color, model number, etc. Due to the fact that each term in the data set becomes an attribute, we have all the information about the data set, and it is more convenient to do comparisons. One point is that if two words have the same frequency in the data set, they get the same weight in terms of values; however, it is not a concern because we are doing the comparison between samples, and even though these words have the same value assigned, they are still placed in different columns. Ideally, if two products reach Product Level Equivalency, they should have the same values in terms of vectors.

3. Truncated SVD

Picking up all the information is definitely beneficial for performing the comparison, especially for reaching the PLE, but it also brings problems in how to deal with a huge data set with 232,730 features. With the physical limitations in hardware, we need to do decomposition to the dimensionality. PCA is the first approach when thinking of dimensionality reduction. However, the matrix is sparse, and PCA could not center the data. Contrary to PCA, Truncated SVD does not center the data before computing the singular value decomposition, which works efficiently with the data matrix TF-IDF generated. Most samples, which combine a title and an

attribute in a row, contain less than a hundred terms. Balancing the goals of information retention and hardware limitations, the dimensionality reduction is done with Truncated SVD and keeps the most useful features in hundreds.

4. Hierarchical agglomerative clustering

Hierarchical clustering is the algorithm we used to perform clustering on the data set with unknown clusters. More specifically, we used hierarchical agglomerative clustering with many different linkage criteria such as Ward, complete linkage, average linkage, and single linkage. This treats each sample as a singleton cluster at the outset, and then successively merges pairs of clusters until all clusters have been merged into a single cluster that contains all samples. Although dimensionality reduction has been performed, parsing this large data set into the model still needs a huge amount of time.

In order to test our methods and be able to adjust it in a short time, sub-datasets were needed. We tested our approach in sub data set by dividing the 1 million data lines into five categories according to category labels and performed the clustering in each category along with different linkage criteria. In testing each category, we generated a specific matrix through TF-IDF by feeding the titles and attributes from this specific category. Even though the samples were reduced, the smallest dimensionality of the data matrix generated by TF-IDF is still larger than 30,000. To have a uniform format, we reduced dimensionality of the data matrix to 300. We then parsed these matrices into the model, along with sub validation sets to calculate the accuracy by using the eBay official evaluation formula and determine the performance of our model. After testing on all five categories, we returned to the origin huge data set and parsed titles and attributes into the model with the origin validation data set for evaluation.

Results

In our test for each sub dataset and the 1 million dataset, we generated baselines and evaluated performance under different given distance threshold along with the dendrogram. (Using the validation set)

	Rand index (D=0, Baseline)	Rand index
Category 1	0.999985271618887	0.9999876787504262 (D = 0.1)

Category 2	0.9997182817808232	0.9997418137991312 (D = 0.1)
Category 3	0.999457663887194	0.9995287899347752 (D = 1e-4)
Category 4	0.9991819699499165	0.9992543127434613 (D = 1e-16)
Category 5	0.996657363536055	0.9975498150191472 (D = 0.1)
1 million dataset	0.9999654169678265	0.9999689695766594 (D = 1e-17)

(The dendrograms are attached as addendums)

Lessons Learned & Future Direction

We did not go in the right direction at first. However, because of those experiences, we learned a lot more in dealing with categorical data along with many different clustering models. For categorical data, data preprocessing is very important. Even though TF-IDF is capable of filtering some special characters, there are still many useless and meaningless characters and terms, and they are the biggest limitation of accuracy. Furthermore, the description from the eBay system is user-defined. Thus, although two products are exactly the same on the PLE level, they still have the possibility of being clustered into different groups due to different descriptions.

How to minimize the difference between the same products with different descriptions is our future direction. One possible approach might be utilizing a web service to a publicly available language translator. Of the 1 million listings, only a small fraction (less than 20 in our data set) contained non-English words. On the one hand, the overall gain in matching from replacing them with English words would be small. However, the fact that there were so few might make utilization of a modest web service practical.

References

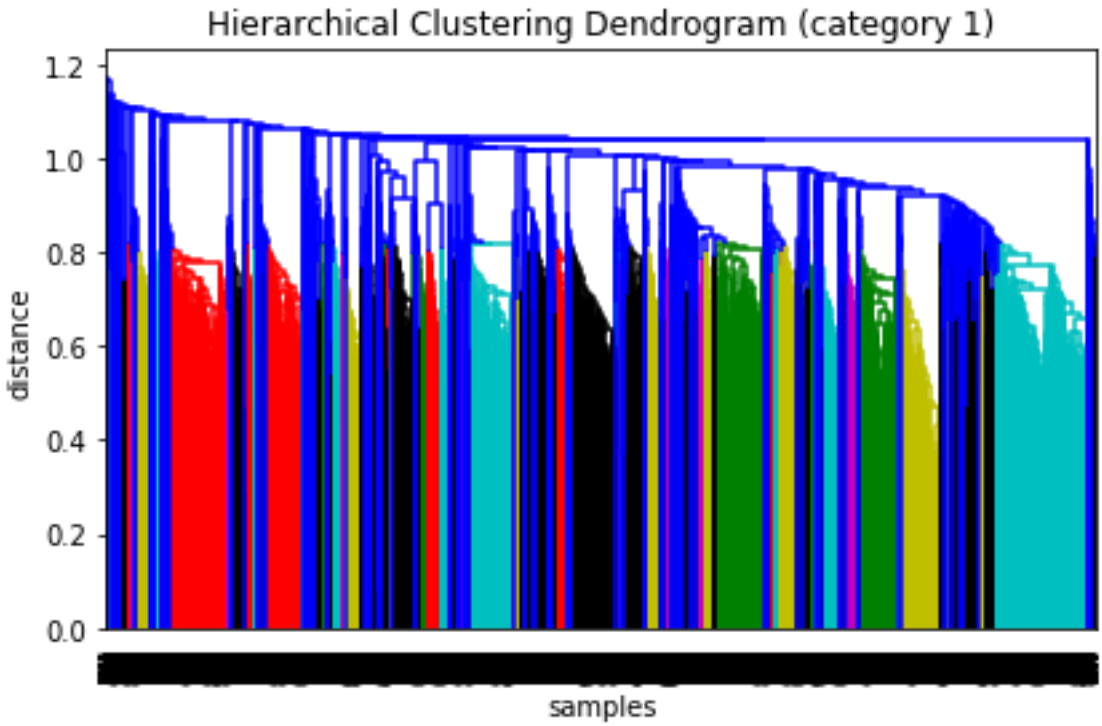
eBay University Machine Learning Competition official website https://evalai.cloudcv.org/web/challenges/challenge-page/462/overview

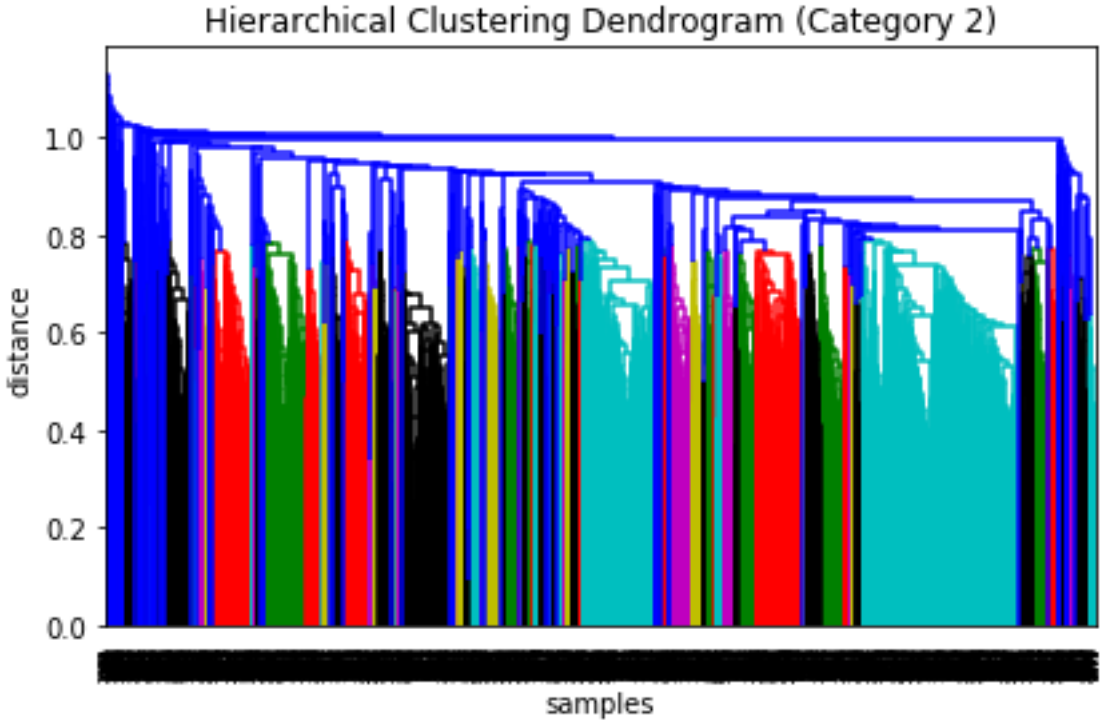
Scikit-learn official website https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

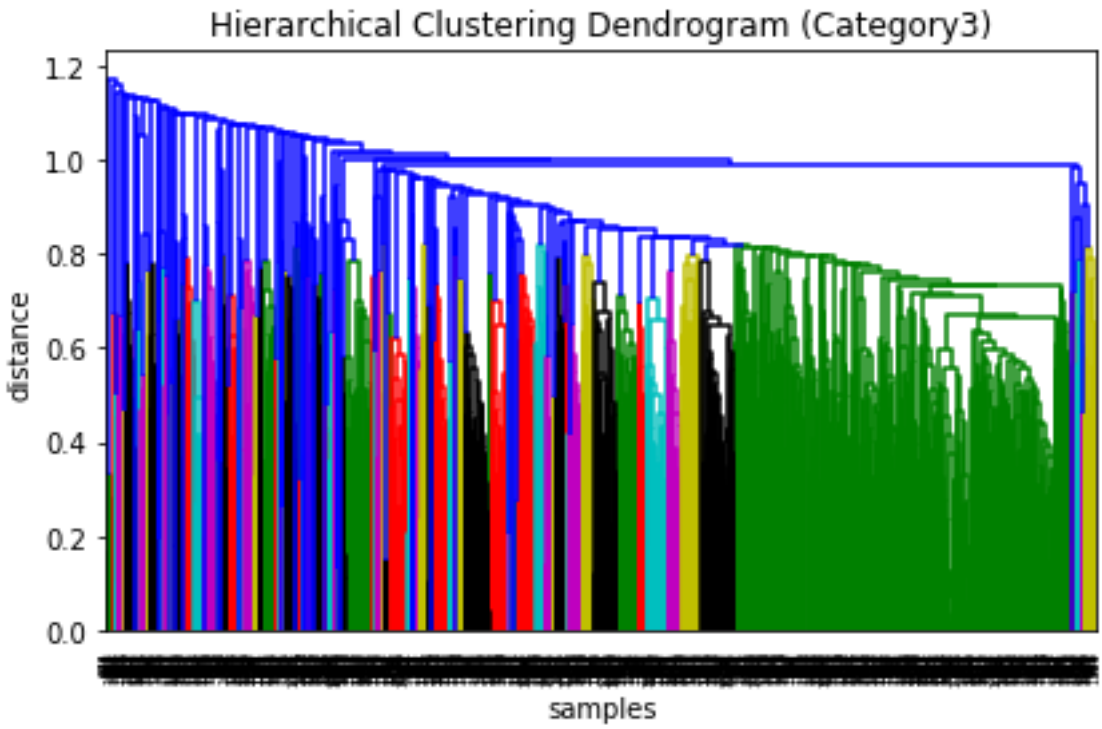
TF-IDF official website http://www.tfidf.com/

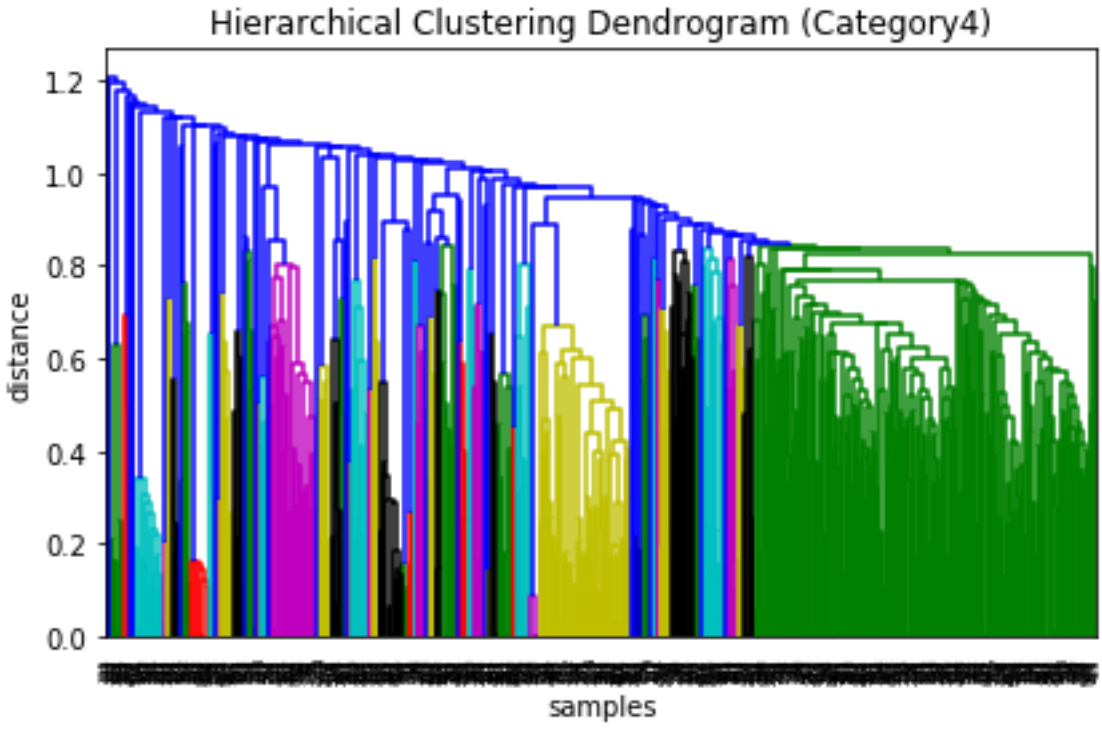
NLTK Documentation website https://www.nltk.org

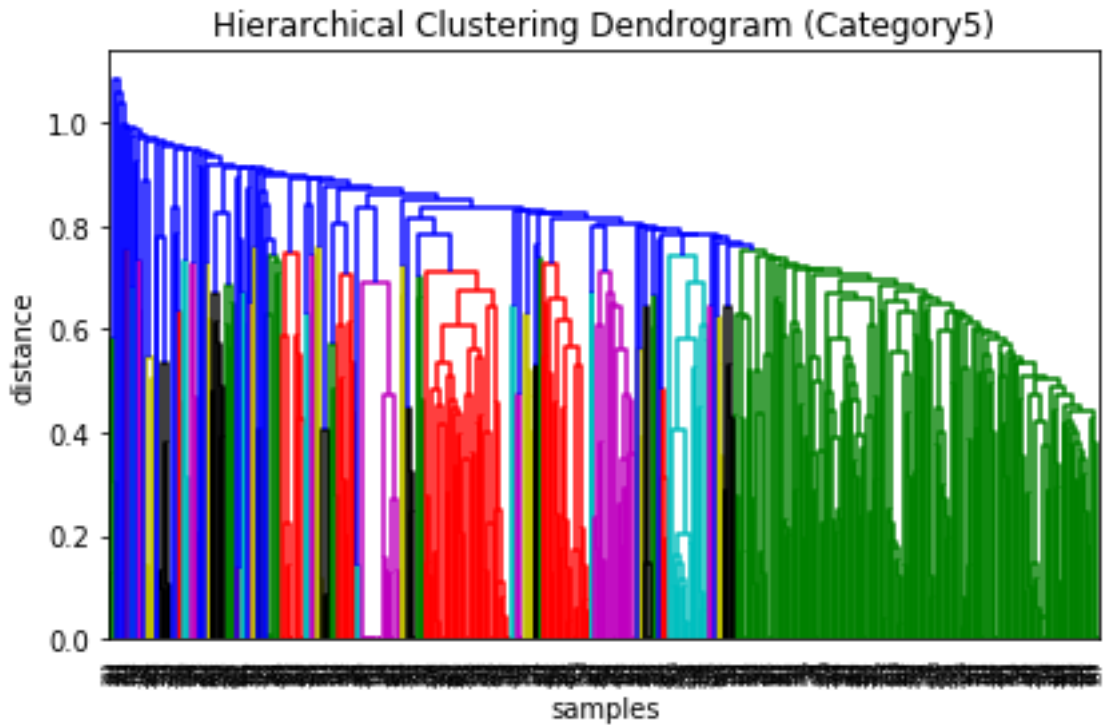
Appendixes

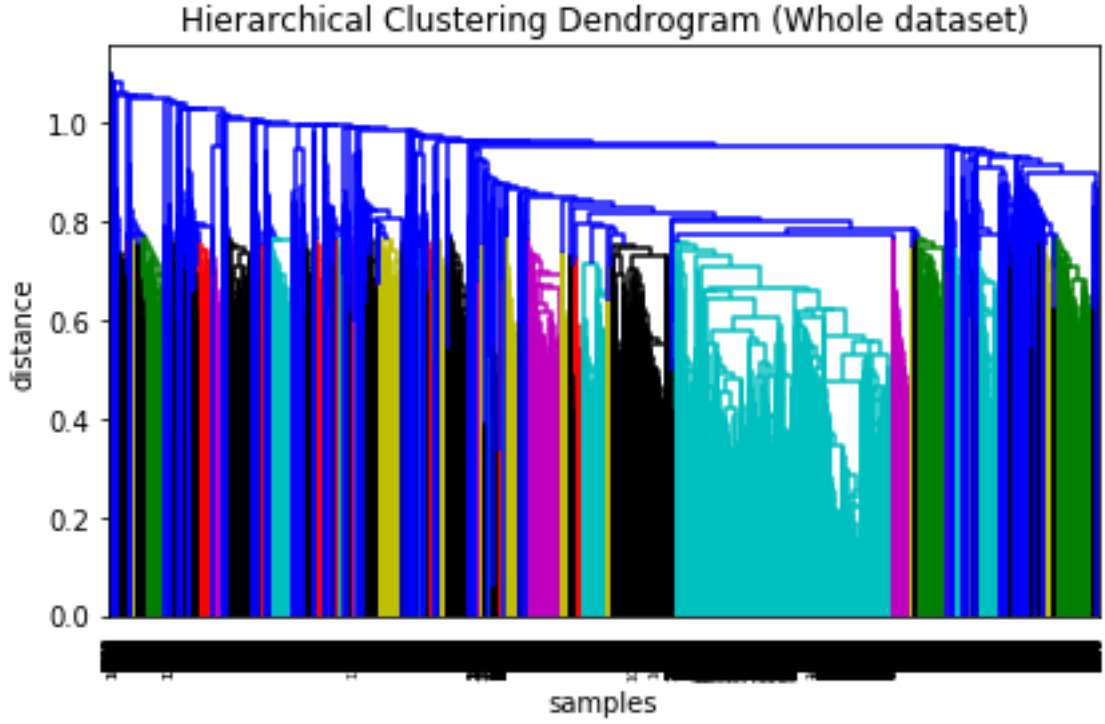

Hierarchical Clustering Dendrogram (category 1)
distance
1.2
1.0
0.8
0.6
0.4
0.2
0.0
samples


Hierarchical Clustering Dendrogram (Category 2)
distance
1.0
0.8
0.6
0.4
0.2
0.0
samples


Hierarchical Clustering Dendrogram (Category3)
distance
1.2
1.0
0.8
0.6
0.4
0.2
0.0
samples


Hierarchical Clustering Dendrogram (Category4)
distance
1.2
1.0
0.8
0.6
0.4
0.2
0.0
samples


Hierarchical Clustering Dendrogram (Category5)
distance
1.0
0.8
0.6
0.4
0.2
0.0
samples


Hierarchical Clustering Dendrogram (Whole dataset)
distance
1.0
0.8
0.6
0.4
0.2
0.0
samples