Lab 5: ML Ecommerce Clustering

Objectives

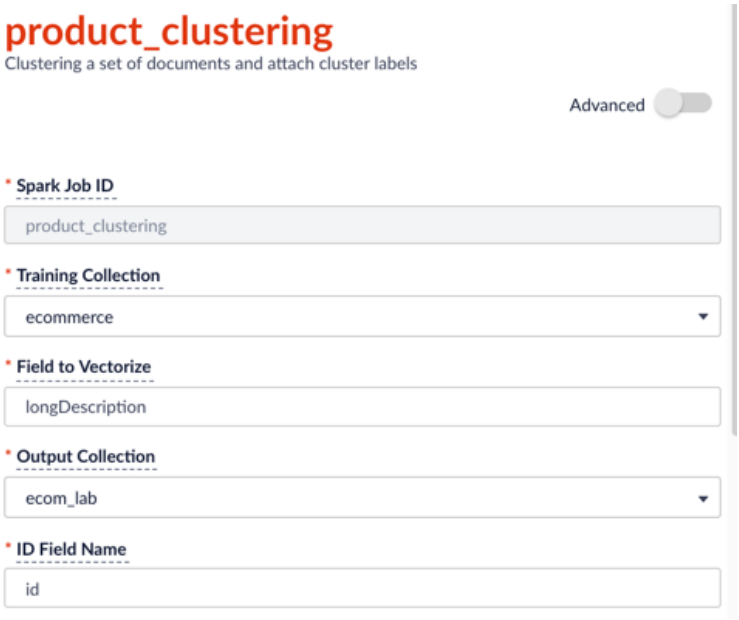
In this lab you will learn how to

- re-group product catalog based on product descriptions and find anomalies using document clustering Spark job in Fusion.

Exercise 1:

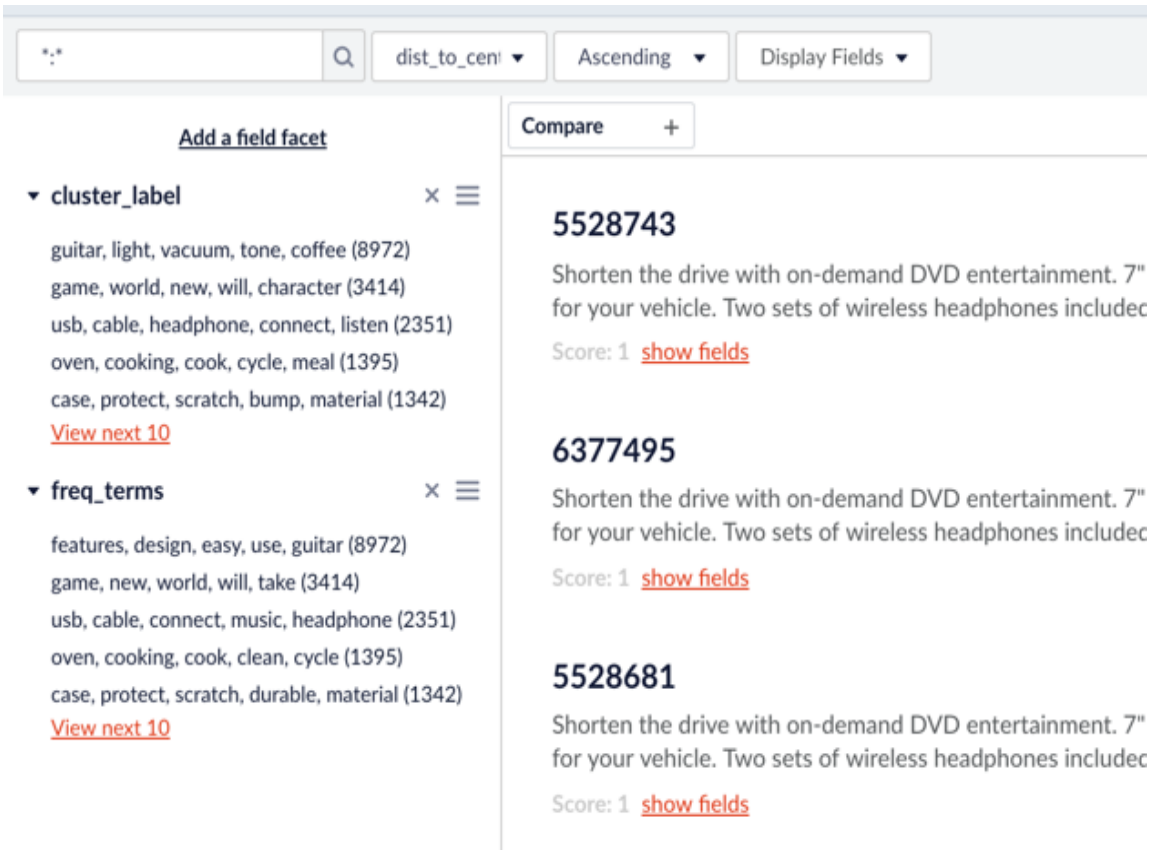
1. Go to jobs list, choose job “document clustering” (or create one if it doesn’t exist)

2. Specify training data collection (“labs”), field to vectorize (“longDescription”), provide job id and output collection (can be the same as training data collection, then the output fields will be attached to the original data). Specify (in advanced area) training data sampling fraction to be 0.2 to reduce running time.



3. Save the job and run. While it's running, go to terminal and cd to Fusion folder, issue the following command to check running status:
   tail -f var/log/api/spark-driver-default.log | grep ClusteringTask:

4. After the run finished, go to output collection, add facets: "cluster_labels", "freq_terms", and in choose sort field section, select "dist to center" and ascending order based on this field.



5. Explore the results by looking at the clustering labels, click on a few to see the documents within it. Do you think it needs to be splitted into more clusters, especially for the biggest cluster here? Also, click on the “short_doc” facet, do you find any problem those descriptions?

Exercise 2:

1. Change on the following config on top the current config:
    a. Min possible number of clusters to 20, max possible number of clusters to 40 (bigger number of clusters, more detailed groupings will be found, but may leads to un needed splitting)

b. Change word2vec Dimension from 0 to 100 to switch using word2vec (word2vec can provide more evenly distributed clusters)

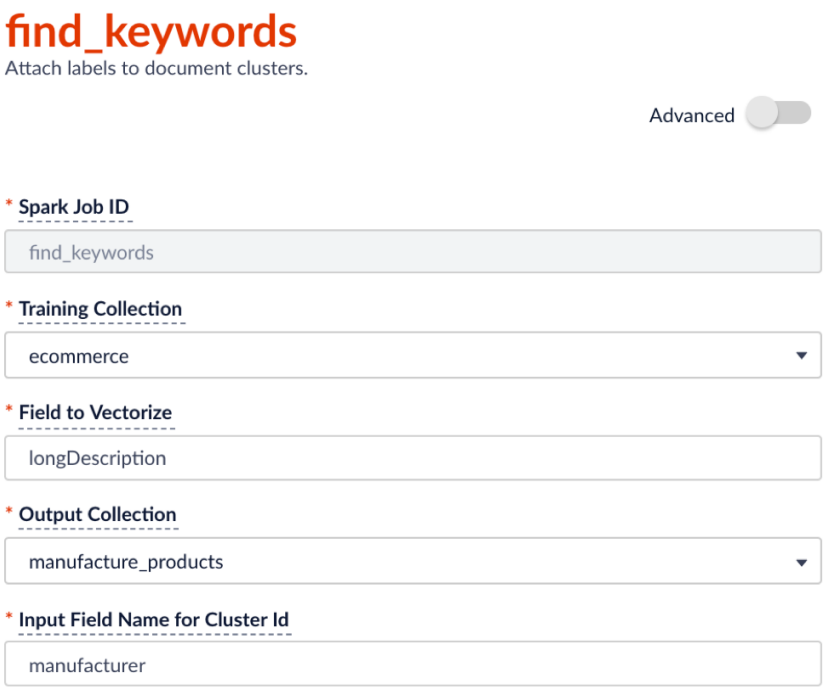
c. Increase outlier cutoff from 0.01 to 0.02 to capture more outliers.

2. Run the job and explore results? Did you find more detailed clusters? How about outlier_groups in the “cluster_label” field. Click on an outlier group and look at the “freq_terms” field to see what are the main keywords show up in the outlier groups.

Exercise 3:
Find what are the top products manufacturers build in the catalog.

1. Go to jobs list, choose job “cluster labeling”.

2. Specify training data collection (“labs”), field to vectorize (“longDescription”), cluster id field (“manufacturer”), provide job id and output collection. Specify (in advanced area) training data filter query: manufacturer:Samsung OR manufacturer:Canon OR manufacturer:Toshiba OR manufacturer:LG OR manufacturer:GE



3. Save the job and run. While it's running, go to terminal and cd to Fusion folder, issue the following command to check running status:
   tail -f var/log/api/spark-driver-default.log | grep ClusterLabel:

4. After the run finished, go to output collection, add facets: "cluster_labels", "freq_terms" and "manufacturer".

5. Explore the output, what are the main products each manufacture sell?

Clustering job troubleshooting tips:

1. For short documents with vocabulary size below 50K, recommend using TFIDF as vectorization method + hierarchical clustering. For long documents with big vocabulary size, recommend using word2vec vectorization method (with window size >= 8 and dimension >=100).

2. If the resulting cluster size is very uneven, i.e, most of the document cluttered into one big cluster, please try to increase the number of clusters K and make sure perform outlier trimming step in the Fusion solution. Also recommend use word2vec as the vectorization method in this case.

3. If there are many outlier clusters detected and each cluster only have a few outliers (which may not be a bad thing), please try to increase the outlier threshold value in config.

4. If the clustering running time is too long for big corpus, recommend to use word2vec + kmeans algorithm with lower number of clusters K. Avoid searching for K step and use exact K instead can help reduce run time too.

5. If log shows all clusters are generated, but have trouble write back to output collection. It's possible due to the schema setup of output collections. E.g., all strings in Spark will be write back as String type in Solr, which won't work for long strings, then need to setup output collection schema for this field as text_en before run job. Or user can try to use input collection as the output collection, then in Fusion 4.0, the clustering result fields will be appended to the original records.