Q4 [5 points] OpenRefine

OpenRefine is a powerful tool for working with messy data, allowing users to clean and transform data efficiently. Use OpenRefine in this question to clean data from Mercari. Construct GREL queries to filter the entries in this dataset. OpenRefine is a Java application that requires Java JRE to run. However, OpenRefine v.3.6.2 comes with a compatible Java version embedded with the installer. So, there is no need to install Java separately when working with this version. Go through the main features on OpenRefine's homepage. Then, download and install OpenRefine 3.6.2. The link to release 3.6.2 is https://github.com/OpenRefine/OpenRefine/releases/tag/3.6.2

Technology	• OpenRefine **3.6.2**
Deliverables	• **`properties_clean.csv`**: Export the final table as a csv file. • **`changes.json`**: Submit a list of changes made to file in json format. Go to 'Undo/Redo' Tab → 'Extract' → 'Export'. This downloads 'history.json' . Rename it to 'changes.json'. • **`Q4Observations.txt`**: A text file with answers to parts b.i, b.ii, b.iii, b.iv, b.v, b.vi. Provide each answer in a new line in the output format specified. Your file's final formatting should result in a .txt file that has each answer on a new line followed by one blank line.

Tasks and point breakdown

1. Import Dataset
    a. Run OpenRefine and point your browser at https://127.0.0.1:3333.
    b. We use a products dataset from Mercari, derived from a Kaggle competition (Mercari Price Suggestion Challenge). If you are interested in the details, visit the data description page. We have sampled a subset of the dataset provided as "properties.csv".
    c. Choose "Create Project" → This Computer → `properties.csv`. Click "Next".
    d. You will now see a preview of the data. Click "Create Project" at the upper right corner.

2. [5 points] Clean/Refine the Data
    a. [0.5 point] Select the `category_name` column and choose 'Facet by Blank' (Facet → Customized Facets → Facet by blank) to filter out the records that have blank values in this column. Provide the number of rows that return True in `Q4Observations.txt`. Exclude these rows.
        Output format and sample values:
        ```
        i.rows: 500
        ```

    NOTE: OpenRefine maintains a log of all changes. You can undo changes by the "Undo/Redo" button at the upper left corner. You must follow all the steps in order and submit the final cleaned data file `properties_clean.csv`. The changes made by this step need to be present in the final submission. If they are not done at the beginning, the final number of rows can be incorrect and raise errors by the autograder.

    b. [1 point] Split the column `category_name` into multiple columns without removing the original column. For example, a row with "Kids/Toys/Dolls & Accessories" in the `category_name` column would be split across the newly created columns as "Kids", "Toys" and "Dolls & Accessories". Use the existing functionality in OpenRefine that creates multiple columns from an existing column based on a separator (i.e., in this case '/') and does not remove the original `category_name` column. Provide the number of new columns that are created by this operation, excluding the original `category_name` column.
        Output format and sample values:
        ```
        ii.columns: 10
        ```

NOTE: While multiple methods can split data, ensure new columns aren't empty. Validate by sorting and checking for null values after using our suggested method in step b.

c. [0.5 points] Select the column `name` and apply the Text Facet (Facet → Text Facet). Cluster by using (Edit Cells → Cluster and Edit …), and then this opens a window where you can choose different "methods" and "keying functions" to use while clustering. Choose the "keying function" that produces the smallest number of clusters under the "Key Collision" method. Click "Select All" and "Merged Selected & Close." Provide the name of the keying function and number of clusters produced.

   Output format and sample values:

   `iii.function: fingerprint, 200`

NOTE: Use the default Ngram size when testing Ngram-fingerprint.

d. [1 point] Replace the null values in the `brand_name` column with the text "Unknown" (Edit Cells → Transform). Provide the expression used.

   Output format and sample values:

   `iv.GREL_categoryname: endsWith("food", "ood")`

NOTE: "Unknown" is case and space sensitive ("Unknown" is different from "unknown" and "Unknown ".)

e. [0.5 point] Create a new column `high_priced` with the values 0 or 1 based on the "price" column with the following conditions: if the price is greater than 90, `high_priced` should be set as 1, else 0. Provide the GREL expression used to perform this.

   Output format and sample values:

   `v.GREL_highpriced: endsWith("food", "ood")`

f. [1.5 points] Create a new column `has_offer` with the values 0 or 1 based on the `item_description` column with the following conditions: If it contains the text "discount" or "offer" or "sale", then set the value in `has_offer` as 1, else 0. Provide the GREL expression used to perform this. Convert the text to lowercase in the GREL expression before you search for the terms.

   Output format and sample values:

   `vi.GREL_hasoffer: endsWith("food", "ood")`