Python Task

General

1. Use python 3.8 and above
2. You can use any library that is out there, no need to implement what others already have...
3. Make your code as readable as you can
4. Add README.md file

Tasks

1. CSV
    - Convert “tsv” (tab separated values) to “csv” (comma separated values)
      python_home_task_file.tsv file attached is currently in tsv format. convert it to a valid csv format into the file named python_home_task_file.csv and attach it.

    - Add column to the csv
      add a column named price_edited. fill it with the float value you have in “search_price” column
      we expect to run your code like so:

      ```
      $ python csv-task.py --infile python_home_task_file.csv --out python_home_task_file_with_price.csv
      ```

2. Regex
   Take out from python_home_task_file.csv file all the Knit products without jumpers. use regex and not code for doing the selection task.
   we expect to run your code like so:

   ```
   $ python regex-task.py --infile python_home_task_file.csv --out python_home_task_file_regex.csv
   ```

Design Task

Description

Currently we receive a full customer catalog on an hourly basis. Each time we run over all products in the catalog. For every product we run our algorithm on product image to fetch the deeptags for the image. Then we write the catalog data, with the new calculated deeptags into the Elastic search database. So each run creates a new index which is a production index. Meaning this is an index that we use for searching customers' products.

Problem

The process as described above is not ideal, and certainly not efficient. Usually not all products change from one indexing to another. Some stay as they were, some get small changes - like change in price, in_stock count. Other products have change in the images themselves. And of course there are new items added and items that no longer exist in the catalog, but did exist in the previous version of the catalog.

Solution

Please describe a flow that will make the whole process more efficient. For example - we don't want to process products that we indexed before, and they didn't change.
Feel free to describe any tools, data structures, or other resources that you think can help.
* Remember that the production index is used by the customers - so we don't want to touch it, or to overload it, with read/write actions.