Python developer

Home assignment (Python Task)

General

- Use Python 3.8 and above
- You can use any library
- Make your code as readable as you can
- Add a README.md file
- Please use the attached file (*python_home_task_file.tsv*)

Tasks

- CSV
  a. Convert "TSV" (tab-separated values) to "CSV" (comma-separated values
     - The file *python_home_task_file.tsv* is attached and is currently in TSV format.
     - Convert it to a valid CSV format into the file named *python_home_task_file.csv* and attach the file.

  b. Add a column to the CSV file
     - Add a column named *price_edited*.
     - Fill the column with the float value from the "*search_price*" column.
     - We expect to run your code like this:

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

Regex

Remove all the knit products without jumpers from the *python_home_task_file.csv* file. To perform the selection task, use regex rather than code.

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 an entire customer catalog on an hourly basis. In each run, we scan all products in the catalog. For every product, we process the image through our algorithm, fetching the AI tags for the image. Then, we write the catalog data, with the newly calculated AI tags, into the Elastic Search database. So, each run creates a new index, which is a production index. We use this index to search for customers' products.

Problem

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

Your task

Please describe a flow that will make the whole process more efficient. For example - we don’t want to process products we indexed before and didn’t change.
Please describe any tools, data structures, or other resources you think can help.

Remember that customers rely on the production index, so we should avoid modifying or overloading it with read/write actions.

Good luck!