Secondhand CPU Processor Price collection

Chien Wei, Hsiung – G01272835

1. Introduction

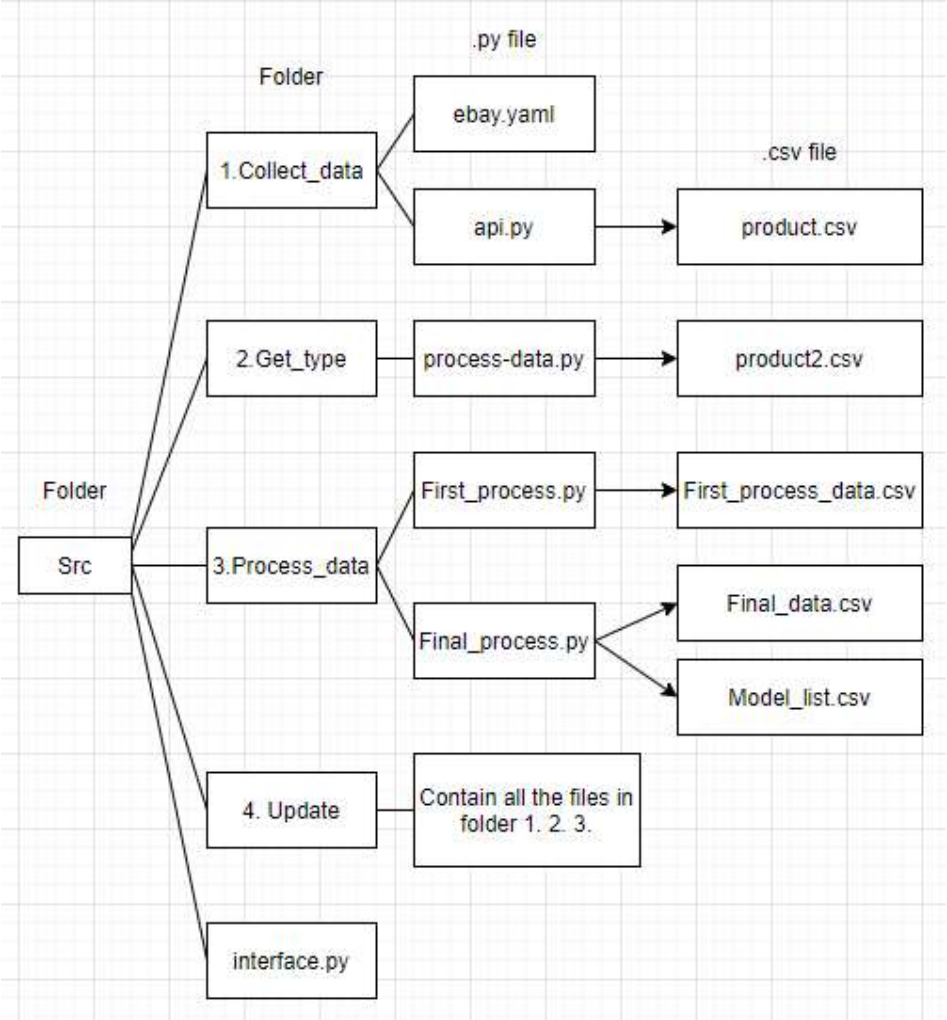
Nowadays, the newest computer components, like CPU processors, are more and more expensive. Instead of buying a new one with full price, we can use a secondhand one as a transition and wait for price cut or the new products coming out.

2. Problem Statement

I want to look for some used high-level CPU processors, ex : i9-10900K, with the lowest price. It can not only meet my need but also let me save lots of money.

3. Src Folder Structure

- The following picture is the structure of my src folder :
  - **"1. Collect_data" folder** : Use Ebay API to collect price, start date, and URL link of each case. Save data as csv file.
  - **"2. Get_type" folder** : Use Beatifulsoup to request URL link of each case to get the processor type of each case. Add this information into the data and then remove attribute : URL link. Save data as csv file.
  - **"3. Process_data" folder** :
    i. "First_process.py" : Change the format of data and remove unclear cases. Then, sort the data by processor type.
    ii. "Final_process.py" : First, sort the data by date and save them as csv file. Second, count the number of cases of each processor type. Save this information as "Model_list.csv".
  - **"4. Update" folder** : Contain all the same files in "Folder 1. 2. 3.". "Folder 1. 2. 3." process all the cases from about one month ago. But "Folder 4." just gather the data within two or three days. And then save the updated data into "3. Process_data" folder as "Model_list.csv", "Final_data.csv".
  - **"interface.py"** : Read "Model_list.csv", "Final_data.csv" from "3. Process_data" folder. Use pyqt5 to create a user interface. It shows a scatter plot of a processor type.



4. Literature Review and Dataset

- All the documents I used are on the websites of the tools. Like, Ebay API, BeautifulSoup, pyqt5 and Matplot.
- I use Ebay API and BeautifulSoup to create my own dataset.

5. Methods and Techniques

- **Step 1 (“api.py” in Folder “1. Colloect_data”) :**
  - Use Ebay API. It offers a filter that help me find the item I want. For example, use a variable “Condition” to filter new, used, refurbished item; use “MinPrice” to search an item above this minimal price.
  - After sending a request, it returns each case. Show as below :

```
{'itemId': '353476255611', 'title': 'amd athlon 64 x2 4200+ and heatsink', 'globalId': 'EBAY-US',
'primaryCategory': {'categoryId': '164', 'categoryName': 'CPUs/Processors'}, 'galleryURL': 'https://
thumbs4.ebaystatic.com/m/mhq06sbI-jaIzKpg4TMBFhA/140.jpg', 'viewItemURL': 'https://www.ebay.com/itm/amd-
athlon-64-x2-4200-and-heatsink-/353476255611', 'autoPay': 'true', 'postalCode': '018**', 'location':
'Lowell,MA,USA', 'country': 'US', 'shippingInfo': {'shippingServiceCost': {'_currencyId': 'USD', 'value':
'0.0'}, 'shippingType': 'Free', 'shipToLocations': 'Worldwide', 'expeditedShipping': 'false',
'oneDayShippingAvailable': 'false', 'handlingTime': '3'}, 'sellingStatus': {'currentPrice': {'_currencyId':
'USD', 'value': '20.0'}, 'convertedCurrentPrice': {'_currencyId': 'USD', 'value': '20.0'}, 'sellingState':
'Active', 'timeLeft': 'P29DT23H51M30S'}, 'listingInfo': {'bestOfferEnabled': 'true', 'buyItNowAvailable':
'false', 'startTime': datetime.datetime(2021, 4, 30, 3, 9, 25), 'endTime': datetime.datetime(2021, 5, 30,
3, 9, 25), 'listingType': 'FixedPrice', 'gift': 'false'}, 'returnsAccepted': 'true', 'condition':
{'conditionId': '3000', 'conditionDisplayName': 'Used'}, 'isMultiVariationListing': 'false',
'topRatedListing': 'false'}
```


- The red underlines showed on above are the attributes I want. URL, price and start time show. Collect these of each case and then save them as a csv file.
- The below figure is part of the data I collected :

	column 1	column 2	column 3
1	soldprice	date	link
2	10.0	2021-04-24 19:01:18	https://www.ebay.com/itm/ryzen-7-3700x-/184790105355
3	28.49	2021-04-24 19:00:40	https://www.ebay.com/itm/AMD-Phenom-II-X4-945-Quad-Core-3GHz-6MB-Socket-AM2-AM3-Processor-HDX945WFK4DGM-/402807293640
4	15.69	2021-04-24 19:00:39	https://www.ebay.com/itm/AMD-Athlon-II-X4-635-Quad-Core-2-9GHz-Socket-AM2-AM3-Processor-ADX635WFK42GM-/402807293635
5	28.99	2021-04-24 19:00:38	https://www.ebay.com/itm/AMD-A8-Series-A8-3520M-1-6GHz-Quad-Core-Socket-FS1-35W-Laptop-CPU-AM3520DDX43GX-/402807293638
6	39.49	2021-04-24 19:00:37	https://www.ebay.com/itm/AMD-A10-5750M-Quad-Core-2-5GHz-Socket-FS1-35W-Laptop-CPU-AM5750DEC44HL-Processor-/402807293641
7	22.49	2021-04-24 19:00:36	https://www.ebay.com/itm/AMD-A8-Series-A8-3500M-1-5GHz-Quad-Core-Socket-FS1-35W-Laptop-CPU-AM3500DDX43GX-/402807293647
8	24.49	2021-04-24 19:00:35	https://www.ebay.com/itm/AMD-FX-4100-Quad-Core-3-6GHz-12MB-2600MHz-Bus-Socket-AM3-CPU-FD4100WMW4KGU-/402807293651
9	12.49	2021-04-24 19:00:34	https://www.ebay.com/itm/AMD-Phenom-II-X2-550-Black-Edition-Dual-Core-3-1GHz-Socket-AM2-AM3-HDZ550WFK2DG-/402807293645
10	11.49	2021-04-24 19:00:33	https://www.ebay.com/itm/AMD-Athlon-II-X2-280-3-6GHz-Dual-Core-Socket-AM2-AM3-65W-CPU-ADX280OCK23GM-/402807293627
11	29.49	2021-04-24 19:00:33	https://www.ebay.com/itm/AMD-A8-3870K-Quad-Core-3GHz-4MB-Socket-FM1-CPU-AD3870WNZ43GX-Radeon-HD-6550-/402807293636
12	100.0	2021-04-24 18:59:58	https://www.ebay.com/itm/AMD-Ryzen-3-1200-Desktop-Processor-YD1200BBM4KAE-Socket-AM4-CPU-Wraith-Cooler-/124695655380
13	899.99	2021-04-24 18:49:49	https://www.ebay.com/itm/AMD-Ryzen-9-5950X-Desktop-Processor-4-9GHz-16-Cores-Socket-AM4-/194062750020
14	4.56	2021-04-24 18:45:45	https://www.ebay.com/itm/AMD-Turion-64-X2-TL-58-1-9GHz-Dual-Core-TMDTL58HAX5DM-Processor-/124695641136
15	25.6	2021-04-24 18:42:38	https://www.ebay.com/itm/AMD-FX-6300-6-Core-3-50GHz-CPU-Processor-FD6300WMW6KHK-/174744923704
16	65.0	2021-04-24 18:36:43	https://www.ebay.com/itm/AMD-Athlon-3000G-Desktop-Processor-3-5GHz-2-Cores-Socket-AM4-No-Box-/284270595710
17	81.0	2021-04-24 18:33:21	https://www.ebay.com/itm/AMD-Ryzen-5-3600x-DAMAGED-PINS-/393267820863

- **Step 2 (“process-data.py” in Folder “2.Get_type”) :**
  - Read the csv file created by step 1.
  - Use “BeautifulSoup” to request the URL link of each case.
  - Get the information “processor type” on the website. And there are two type of presentation on the website. Show as below :

    Type 1 :


```
▼<li class>
    <div class="s-name">Processor Model</div>
    <div class="s-value">AMD Athlon 3000G</div>
```

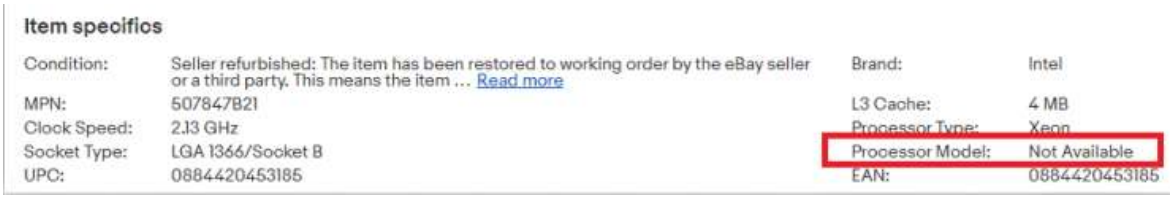
    Type 2 :

```
▶<td width="50.0%">…</td>
  <td class="attrLabels">
                                                    Processor
  Model: </td>
▼<td width="50.0%">
    <span>AMD Athlon II X4 635</span>
  </td>
```

  - Then, add this information into data and remove the attribute – link. Save the data as a csv file.
  - The below figure is part of the data after these processes :

	column 1	column 2	column 3
1	soldprice	date	model
2	10.0	2021-04-24 19:01:18	ryzen 7 3700x
3	28.49	2021-04-24 19:00:40	AMD Phenom II X4 945
4	15.69	2021-04-24 19:00:39	AMD Athlon II X4 635
5	28.99	2021-04-24 19:00:38	AMD A8-3520M
6	39.49	2021-04-24 19:00:37	AMD A10-5750M
7	22.49	2021-04-24 19:00:36	AMD A8-3500M
8	24.49	2021-04-24 19:00:35	AMD FX-4100
9	12.49	2021-04-24 19:00:34	AMD Phenom II X2 550 Black
10	11.49	2021-04-24 19:00:33	AMD Athlon II X2 280
11	29.49	2021-04-24 19:00:33	AMD A8-3870K
12	100.0	2021-04-24 18:59:58	AMD Ryzen 3 1200
13	899.99	2021-04-24 18:49:49	Amd Ryzen 9 5950X

- **Step 3 (“First_process.py” in Folder “3.Process_data”) :**
  - Read the csv file created by step 2.
  - Because the information “processor type” is filled by sellers, some cases just lack this information. And some cases show the information but in a weird format. For example : Lack of information :



Wrong format :

  - So, get rid of these special cases by checking whether there is a “amd” or “intel” substring in the processor type.
  - Turn processor type into lower case. Then, sort the data by processor type and save the data as “First_process_data.csv” file. Show as below :

	column 1	column 2	column 3
1	soldprice	date	model
2	35.19	2021-04-17 15:49:11	intel xeon x7560
3	38.39	2021-04-17 15:17:45	intel xeon x7560
4	15.0	2021-04-21 16:23:27	intel xeon x7560
5	26.39	2021-04-17 15:32:52	intel xeon x7560
6	65.0	2021-04-09 18:29:49	intel xeon x7550
7	49.5	2021-04-19 19:25:35	intel xeon x5690
8	72.5	2021-04-19 05:55:52	intel xeon x5690
9	49.5	2021-04-23 19:35:12	intel xeon x5690
10	49.5	2021-04-23 19:40:12	intel xeon x5690
11	49.5	2021-04-23 19:00:05	intel xeon x5690
12	150.0	2021-04-19 18:57:33	intel xeon x5690
13	49.5	2021-04-19 19:00:01	intel xeon x5690
14	49.5	2021-04-19 19:05:43	intel xeon x5690
15	49.5	2021-04-19 19:10:25	intel xeon x5690
16	68.78	2021-03-26 14:01:37	intel xeon x5690
17	137.7	2021-04-15 14:58:05	intel xeon x5690
18	49.5	2021-04-19 19:20:59	intel xeon x5690
19	49.5	2021-04-24 19:00:51	intel xeon x5690

- However, there are still some wrong processor types. I have to delete them by myself. But it only takes one or two minutes. It's obvious because all the type are sorted. Show as below :

	column 1	column 2	column 3
1	45.0	2021-04-15 18:20:46	intel, amd, and cyrix
2	23.33	2021-04-13 12:31:23	intel ® core™ i3-1082 8/12ft processor
3	335.0	2021-04-23 15:05:57	intel xeone7-4820v4 sr2s4
4	35.19	2021-04-17 15:49:11	intel xeon x7560
5	38.39	2021-04-17 15:17:45	intel xeon x7560
6	15.0	2021-04-21 16:23:27	intel xeon x7560
7	26.39	2021-04-17 15:32:52	intel xeon x7560
8	65.0	2021-04-09 18:29:49	intel xeon x7550
9	49.5	2021-04-19 19:25:35	intel xeon x5690
10	72.5	2021-04-19 05:55:52	intel xeon x5690
11	49.5	2021-04-23 19:35:12	intel xeon x5690
12	49.5	2021-04-23 19:40:12	intel xeon x5690
13	49.5	2021-04-23 19:00:05	intel xeon x5690
14	150.0	2021-04-19 18:57:33	intel xeon x5690

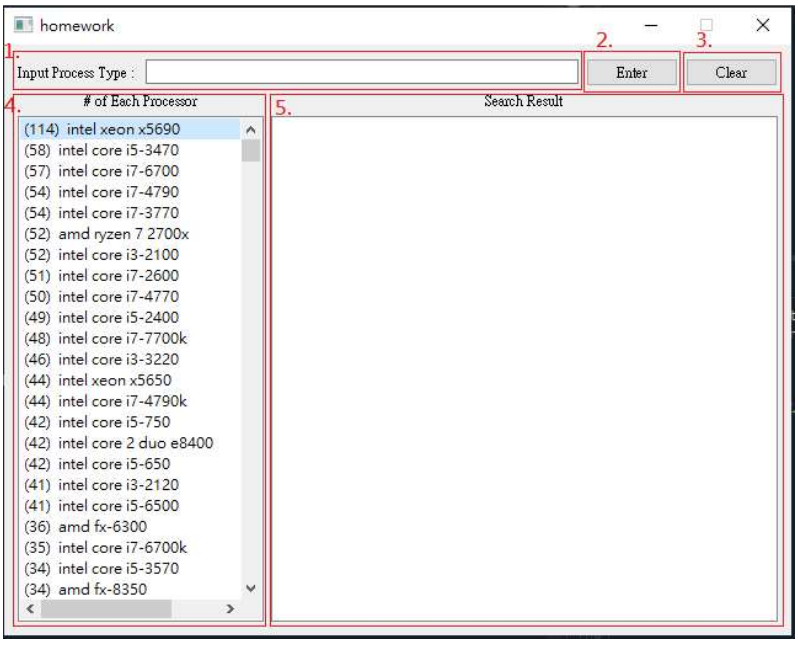
- **Step 4 ("Final_process.py" in Folder "3.Process_data") :**
  - Read the csv file created by step 3.
  - Change the format of data into the one I want. And then sort the data by date. While showing the scatter plot in step 6, the sorted date will present in order on x-axis.
  - Save the data as "Final_data.csv" file. Show as below :

	column 1	column 2	column 3
1	soldprice	date	model
2	85.0	1/04	amd ryzen 3 1200
3	29.85	1/04	amd a8-3850
4	85.0	1/04	amd ryzen 3 1200
5	4.0	1/04	amd athlon ii x2 240
6	17.99	1/04	amd athlon 64 x2 4400+
7	85.0	1/04	amd ryzen 3 1200
8	39.95	1/04	amd a6-7400k
9	29.95	1/04	amd a6-6400
10	22.85	1/04	amd athlon x4 760k
11	721.0	1/04	amd ryzen 9 3900x
12	39.95	1/04	amd a6-7400k
13	125.0	1/04	amd ryzen 5 1600
14	32.99	1/05	amd phenom ii x4 965
15	19.95	1/05	amd phenom ii x3 710
16	99.99	1/05	amd fx-8320
17	22.99	1/05	amd athlon ii x4 630
18	19.19	1/05	amd athlon ii x4 635
19	89.99	1/06	amd phenom ii x6 1055t
20	90.0	1/06	amd a10-9700
21	30.0	1/06	amd phenom ii x4 850

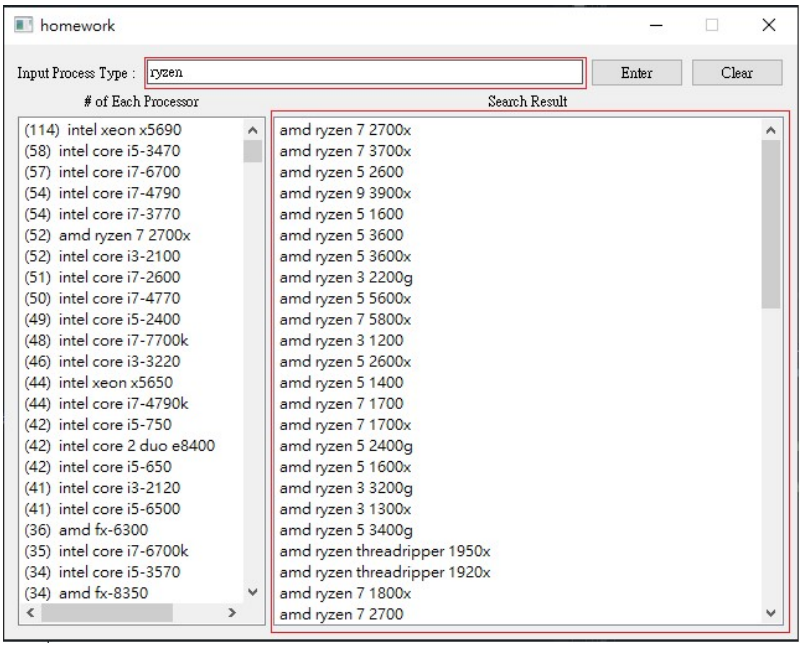
- Use the function “groupby” of Dataframe to count the number of cases of each processor type. Save them as “Model_list.csv” file. Show as below :

	column 1	column 2
1	number	type
2	114	intel xeon x5690
3	58	intel core i5-3470
4	57	intel core i7-6700
5	54	intel core i7-4790
6	54	intel core i7-3770
7	52	amd ryzen 7 2700x
8	52	intel core i3-2100
9	51	intel core i7-2600
10	50	intel core i7-4770
11	49	intel core i5-2400
12	48	intel core i7-7700k
13	46	intel core i3-3220
14	44	intel xeon x5650
15	44	intel core i7-4790k
16	42	intel core i5-750
17	42	intel core 2 duo e8400

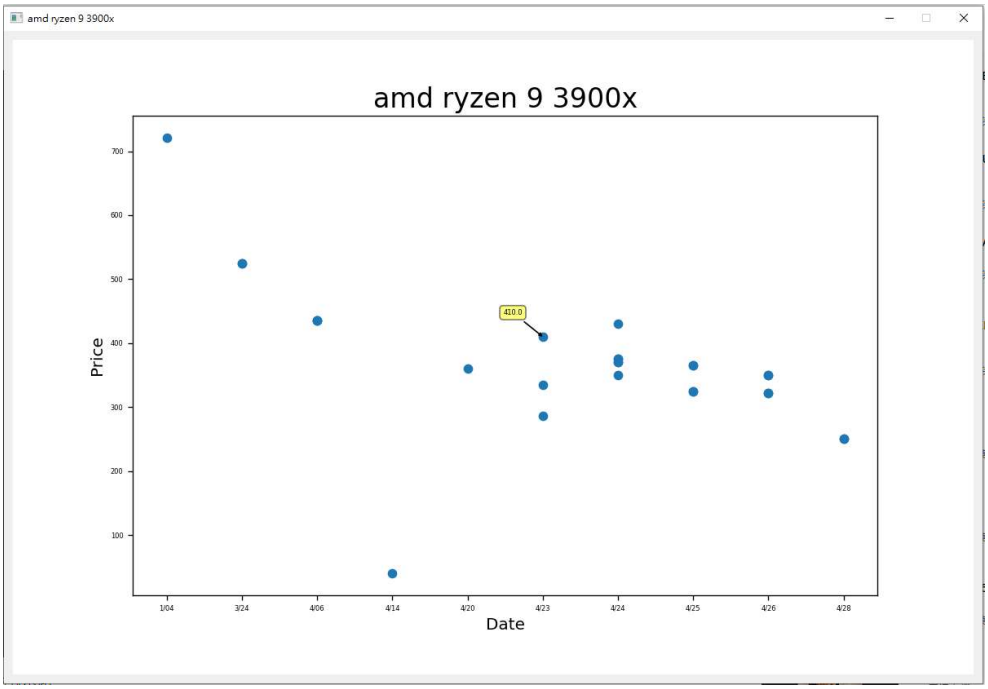
- **Step 5 (All the files In Folder “4. Update”) :**
  - The same as step 1 to 4.
  - But instead of collecting the data from few months ago, it gathers the data within two or three days.
  - Also, instead of saving the data as a new csv file, it put the new updated data into the csv file “Model_list.csv”, “Final_data.csv” in Folder “3. Process_data”.
  - Now, the data will contain the data from long age and the data updated recently.
- **Step 6 (“interface.py”) :**
  - Read the csv file from step 4.
  - Use pyqt5, matplatlib to create user interface and scatter plot. Show as below :



- (1) input a processor type. And then click the (2) Enter button. It will show the search results on (5). For example :



- (3) read the data in “Model_list.csv” created by step 4. It shows the number of cases of each processor type.
- (5) is a clear button. It will clean the input in (1) and the results in (5).
- After clicking any type in (4) or (5), it will show a scatter plot. For example :



- Title is the processor type. X-axis is the date. Y-axis is the price. Put the mouse on a case, it will show the exact price of that case.

6. Conclusion

- There is a big problem. In step 3, I have to delete some cases with wrong information by myself. It's because this information is offered by sellers. Ebay doesn't offer a standard format for each case. And I have no idea about how to solve this problem.
- **Improvement** :
  - As long as I click a dot on a scatter plot, it can open the website directly for me so that I can buy the one I want if I find it's cheap now.
  - Also show the price of a new CPU on the scatter plot so that I can compare the used price to it.

References

1. Ebay api : https://developer.ebay.com/
2. BeautifulSoup : https://www.crummy.com/software/BeautifulSoup/bs4/doc/
3. Pyqt5 : https://realpython.com/python-pyqt-gui-calculator/
4. Matplot : https://matplotlib.org/stable/contents.html
5. Matplot-W3school : https://www.w3schools.com/python/matplotlib_pyplot.asp