PC PARTS SCALPERS MINING & ANALYSIS - PROJECT REPORT

ITEC 4305

April 7, 2021

Anirudh A, 216786642
Ibtihaj B, 215006315
Elwin M, 214821995
Aditi T, 214856389

OVERVIEW - GENERAL INTRODUCTION:

PC Pricing is becoming a major issue these days with prices of parts rising and general availability scarce. It is really important to make sure parts are available for everyone to not hamper innovation. The scarcity has also encouraged scalping by individuals and official retailers to an extent. We are surrounded by technology today and are somewhat dependent on it for everything big or small. This project will focus on finding out the best possible MSRP for the product.

RESEARCH AGENDA:

Our goal was to help determine if the current retail prices of the products are reasonable compared to the MSRP given the availability of the products, and how the prices set by individual 1st party and 3rd party sellers compare. The products include graphics cards from AMD, and Nvidia, as well as CPUs from AMD and Intel.

While our original plans were to compare several brands, and a collection of their offerings, it was easy to see later on that our ideas were ambitious given the time frame. We have then decided to analyze only the RTX 3070 graphics card from Nvidia. This time however, we found that we were able to expand our data collection relatively easily to the entirety of the RTX 30 series line up from Nvidia.

SOURCE & INTRODUCTION TO DATA:

We first obtained the MSRP from the respective product page from Nvidia's website. However, this posed a problem as there are several models (from different partners) listed, each one with different prices. Due to each model having a separate price, we chose to use the Founder's Edition, offered by Nvidia as the base product.

Our next step was to obtain the prices offered by 1st party sellers as well as 3rd party sellers. For the 1st party seller, we chose Newegg. This seller has a major influence online, but no brick-and-mortar stores, making them a prime candidate for our web scraping attempts. Then, for the 3rd party seller, we chose eBay as they also only have an online presence with individuals as the main source of sales.

TECHNIQUES USED ON DATA - METHOD:

We manually collected the MSRP due to the variance of manufacturers websites. Then, web scrapers/ bots were used to collect information from retailers' websites. We will be using various techniques to figure out if the market price is fair for the product using the market data available. Once the data is collected, they will be analyzed using the methods discussed in lectures to determine the accuracy of the MSRP, as well as the variance of 3rd party sellers' prices.

We first analyzed the data and figured out the best techniques to use and then we calculated basic statistics and other stats such as standard deviation, mean, variance, confidence intervals, etc. From that data we also found the distribution of the data of the sample collected.

DATA GATHERING:

We have decided to use Python to create web scraper bots, in combination with other scripts to consolidate our data. By using Python, we were able to make use of existing libraries to make our bots. The libraries we used include Requests, Requests-HTML, Beautiful Soup, and NumPy.

While in most cases only either Requests OR Requests-HTML is required, certain eBay pages failed to respond with Requests, but worked with Requests-HTML. The root of this issue was unresolved. A simple timeout/ failed to respond message was returned. A potential fix for this issue could be to use valid headers requesting the URL, but this fix was not attempted. Beautiful Soup is a very powerful HTML parser that was invaluable for us to gather our data. Lastly, NumPy was used for statistical analysis after we retrieved the required data.

The bots we created are very similar to each other as the premise of web scraping is the same for each website. The only difference is the specifics and data we are searching for. Both scrapers take an input link, which is the search results page, then search the web page for individual product page links. It visits each product page and scrapes the specific details we are looking for.

For Newegg, everything is well organized and all relevant information is present, so it is relatively simple to find the data.

For eBay, it is not as simple as all the results are user created and while certain information is suggested, there is no enforcement regarding the matter. This leads to many posts missing key information. Anytime the scraping attempt fails to detect data from the proper field, the field is registered as "NA" The other issue is that if the search terms are not properly curated, there is a potential to return hundreds, if not thousands of results. Many of these are completely irrelevant. Our original queries returned up to 15,000 mostly irrelevant results, but we were able to narrow it down to under 500 relevant results per search.

Both scripts create respective folders that store their relevant information. This includes links to the actual product pages, the number of results, as well as the JSON data that we created with it. Both of these are stored in separate files.

DATA STORING:

To store our data, we used JSON. To store the individual products, we used the specific format:

```
{
  "000000000000-uniqueID": {
    "Chipset/GPU Model": "string | NA",
    "auction": "Boolean",
    "price": {
      "amount": "double | NA",
      "currency": "string | NA"
    },
    "manufacturer": "string | NA",
    "brand": "string | NA",
    "memory": "int | NA", #appended with “GB”
    "upc": "string | NA",
    "link number": "int",
    "link": "string",
    "vendor": "Newegg | eBay"
  }
}
```

Firstly, the key is a unique identifier retrieved from the URLs for each product page. Then the values are more sets of key-value pairs. The Chipset/GPU Model can either be the [RTX] 3060, 3060 ti, 3070, 3080, or 3090. Then, in the case of eBay, a Boolean value which dictates if the product posting is an “auction” or “buy now” product. The price and its currency are stored because in the next script, a rough conversion will be done to convert USD into CAD. The manufacturer for the set of results we are expecting will always be Nvidia as they are the GPU manufacturer. Had we been searching for a wider variety of products, this field would be more relevant. The brand is the maker of the specific model, for example, Asus, MSI, Gigabyte, etc. The VRAM of each graphics card is then stored appended with “GB” to normalize the data. The UPC/ SKU of the model is kept. The link number for identification purposes is kept as well as the link for it, and lastly, the retailer is kept as well. If any of this data is missing, it will be registered as “NA” in the output file.

This formatting and saving are done with the same script for their respective website.

Once we scraped the data from Newegg and eBay, the data is stored in individual files for permanent storage and easy retrieval. In the case of eBay, several files are produced, separating them between USD and CAD currencies, as well as the Auction values.

To consolidate all our data into a simple small file, we created another script to do just that. It reads the JSON data from any input files and consolidates it into the following format:

```
{
      "3060 | 3060 ti | 3070 | 3080 | 3090": {
        "GPUCount": "int",
        "Prices": "[double]", #stored as a string due to formatting
        "PricesMean": "double",
        "PricesMedian": "double"
      }
}
```

The script loops through each input file and creates a key that is the same as each GPU. This is done so that each GPU is not duplicated. It counts how many occurrences of each GPU is present and stores it. Then it checks the prices to see whether it is CAD or USD, if USD, it converts it at a rate of 1.3 as it is a rough estimate of current values. This is done to normalize the data. It appends the price to a list for the respective GPU. Finally, we used NumPy to display the mean and median from the values in the list.

We omitted certain results from our stored results because they provided inaccurate results, such as a failure to provide the correct Chipset/GPU Model while providing all other data. This reduced our total number of results from 309 to 236 for eBay. These numbers varied depending on when the script was used, but this was the result from our final attempt. This was not a problem for Newegg as all the data was correctly formatted and provided for us to scrape. Newegg returned 36 items all together. Totalling 272 items.

The individual product data and consolidated prices data is stored into separate JSON files for easy readability and analysis. The final consolidated prices data is used for our analysis.

Furthermore, we also kept the links and link number in a separate text file for identification purposes - this corresponds with the data in the JSON files. (see attached list of files)

DATA ANALYSIS:

Based on data collected using the techniques mentioned above, below are the statistics collected for the dataset:

Overall Sample

Count	272
Mean	2073.20
Standard Deviation	1400.70
Min	45.50
25th Percentile	1294.46
50th Percentile	1756.86
75th Percentile	2265.15
Max	11503.70

Grouped by Model

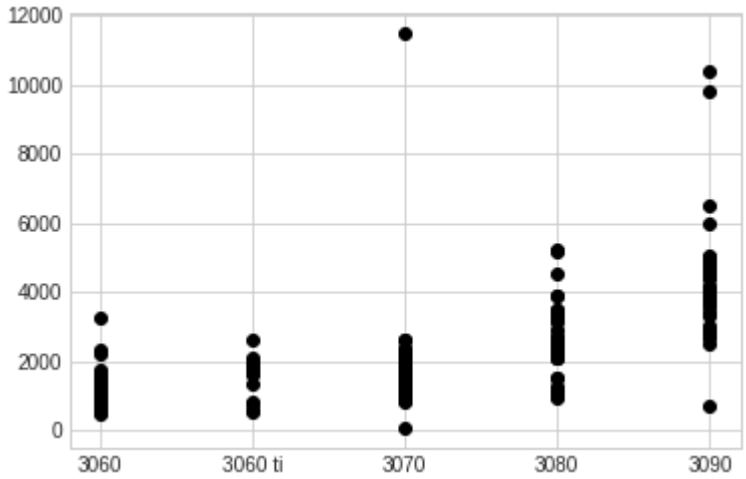
Model	Count	Mean	Standard Dev	Min	25%	50%	75%	Max	MSRP
3060	88	1268.76	354.16	499.99	1110.75	1294.79	1400.99	3249.70	427.70
3060 ti	19	1610.72	547.47	539.99	1476.25	1819.98	1929.84	2599.98	519.98
3070	90	1883.10	1100.49	45.50	1651.86	1851.24	1950	11503.70	648.70
3080	41	2552.46	1104.97	949.99	1499.99	2502.50	3249.98	5199.98	908.70
3090	34	4339.02	1830.13	710.00	3348.75	4116.85	4824.15	10398.70	1948.70

Distribution

Based on our data, we concluded that the data is a normal distribution. We also used the Gaussian function to come to that conclusion. Our calculations suggested that the distribution is normal and not a gaussian. This conclusion was made at various levels (1%,2.5%,5%,10%,15%).

Confidence Interval

The following graph shows the data points where x represents the model and y represents the price per GPU. From the graph we can see that there are a couple of outlier prices in each of the categories. Other than those outliers most data points are within range.



Confidence Interval Calculations:

CI (α = 0.90): (1933.02, 2213.37)

CI (α = 0.95): (1905.99, 2240.40)

CI (α = 0.99): (1852.88, 2293.51)

CI for RTX 3060 (α = 0.90): (1205.99, 1331.52)

CI for RTX 3060 ti (α = 0.90): (1392.92, 1828.51)

CI for RTX 3070 (α = 0.90): (1690.28, 2075.91)

CI for RTX 3080 (α = 0.90): (2261.88, 2843.03)

CI for RTX 3090 (α = 0.90): (3807.84, 4870.18)

ACTUAL OUTCOME - COMPARISON WITH EXPECTED RESULTS:

Based on the results from the above data, we concluded that the MSRP of the GPUs do not actually fall within the confidence interval with α at 0.95 specifying that the prices of the GPUs are not what the manufacturers claim to be. Most GPUs listed in the market are priced a lot higher than the base MSRP except for some exceptional cases.

TIME COMPLEXITY:

The time complexities of our scripts are listed below:

Newegg web scraper: $O(n^2)$

> Initial link is visited storing new relevant links in the list. Each link is visited once. List of links is stored once. Links list is called again to scrape details, with nested loops increasing the complexity to $O(n^2)$ because certain elements are nested together, and it is not possible to gather everything with a single pass-through. Other operations occur, but the total complexity is unchanged. This script is of medium computational intensity as the data set is not extensive enough. (1-2 minutes on average for 35$^{\sim}$ results)

eBay web scraper: O(n)

> The complexity is slightly reduced as most of the procedure is the same as above, except the data is presented on a single page and multiple nested loops are not required to scrape the relevant data. Most portions of the script are O(n) complexity, but there are no looping nested components, allowing the overall complexity to be less. This script is the most computationally expensive due to the larger data set provided. (3-5 minutes on average for 350$^{\sim}$ results)

Data consolidation script: $O(n^2)$

> Each input file is opened and stored in a list. The list is accessed once to create a dictionary of GPUs. The dictionary is then looped through while cross referencing with the initial list for each element, making the complexity $O(n^2)$. This script is the fastest as there is significantly less data to process in an intensive manner. (less than 1 minute on average)

PROBLEMS WITH CURRENT METHODOLOGY:

With our current scripts, we ran into several problems. Many were out of the scope of our assignment to fix in time. Some of the problems are:

> The Requests-HTML library was failing to receive a response from eBay. Cause of this issue is unknown. A lack of valid headers could be a potential issue. This was solved by using the regular Requests library. This was problematic as moving to different search result pages was not efficiently possible with Requests. Timeouts were significantly more common, and Requests-HTML handled JavaScript better.
>
> eBay had 3 types of price labels. Buy Now, Auction, and Sale. All 3 had to be accounted for and was only made aware of each once the script ran into errors. It is possible that more exist and can provide errors later, but until that happens, it is unlikely to be found.

No retailer provides information regarding actual stock of an item other than currently in stock. Our current data is not enough as these items are constantly out of stock and it is almost impossible to track the actual stock numbers (i.e., how many items are moved per week or in some other time period). Our current methods also do not consider potential duplicate postings made by individuals on eBay.

While it is relatively simple to tailor our bots to scrape a variety of websites, it can be time consuming to do so and may not present relevant enough information in a timely manner. This was even problematic for eBay as our scraper only worked on certain pages meeting specific requirements. Even if using the proper search queries, individual pages contained different elements for the exact same visual content.

Our scripts do not incorporate multi-threading. This is extremely inefficient as it can take several minutes for only a few hundred pages to be scraped. There are multiple portions that are possible to be run concurrently, however this can potentially lead to a problem as well if too many requests were made, triggering the website's DDOS/ bot protection.

These were some of the major issues that we ran into with gathering the data, but others can exist as well.

IMPROVEMENTS FOR NEXT TIME:

Smaller improvements to our code that could be made include:

- ➔ More 1st party retailers
- ➔ Make it easier to visit further pages
- ➔ Cleaner code
- ➔ Error catching
- ➔ Handle timeouts
- ➔ Multi-threading

Determine actual stock (although unlikely due to retailers not revealing actual stock per time period)

ROBOTS.TXT:

Newegg: Several content pages are disallowed for any user agents, but product pages and listings are not, allowing access to it with bots.

eBay: Any bots are allowed ONLY with explicit permission. Bots with permission are solely allowed content able to be retrieved from public search engines.

POSSIBLE NEXT STEPS:

- Further analysis on other pc parts such as CPU, RAM/Computer Chips and its shortage, motherboards, SSDs, power supplies and cases - impacted by the US-China Tariffs

https://www.sourcetoday.com/supply-chain/article/21867400/what-do-the-new-us-china-tariffs-mean-for-the-electronics-supply-chain

- Further analysis on websites such as Amazon/Canada Computers.

- Further analysis on how the market value of GPU is increasing exponentially and how demand has risen since midway December 2020.

https://pcpartpicker.com/trends/price/video-card/

https://www.alliedmarketresearch.com/graphic-processing-unit-market

- Further analysis on how while the market value of GPU is increasing exponentially, the computer hardware market has declined from 2019 to 2020 due to tertiary restrictions (to be discussed) - which causes the values of different pc parts to increase.

https://www.businesswire.com/news/home/20200904005427/en/Global-Computer-Hardware-Market-2020-to-2030---COVID-19-Impact-and-Recovery---ResearchAndMarkets.com

TERTIARY INFLUENCES:

North American trade deal increasing tax on imports (i.e., pc parts and other electronics):

https://www.theverge.com/2021/1/7/22217206/nvidia-amd-gpu-trump-tax-china-tariff-exemption-expire

- The Trump administration is now imposing a 25 percent tax on graphics cards imported from China.

- Update regarding MSRP pricing for ASUS components in 2021 applies to graphics cards and motherboards: Their new MSRP reflects increased cost for components, operating costs, and logistical activities plus a continuation of import tariffs.

https://www.techpowerup.com/276858/gpus-to-see-price-increase-due-to-import-tariffs-other-pc-components-to-follow

- This article discusses the GPUs that are selling at much higher prices today compared to the original MSRP and how this is representing a real problem for consumers.

- Prediction indicates that MSRP will increase about $80 for every major GPU manufacturer like ASUS, GIGABYTE, PNY, Zotac, etc. The import tariff exemptions are also supposed to increase MSRPs of other PC components like motherboards, SSDs, PSUs, cases... everything without exemption.

Covid-19 causing shortages of silicon:

https://www.fastcompany.com/90607876/why-is-there-a-silicon-chip-shortage-three-factors-are-to-blame

- Due to remote work, lesser labor in the workforce, the pandemic led to lockdowns and an increase in remote work.
- This led to an explosion of people buying new gadgets to get work done or just pass the time. Gadgets have semiconductors and supply just could not keep up with demand.

https://techmonitor.ai/techonology/hardware/global-chip-shortage-intel-tsmc-samsung

- Although triggered by Covid-19, the chip shortage has revealed structural weakness in an increasingly fragile supply chain and may foreshadow availability issues for years to come.
- This shortage has been triggered by a dramatic spike in demand, caused by the pandemic. Demand for PCs rose as companies around the world pivoted to remote working and schools switched to online learning in the wake of the Covid-19 pandemic, while the growth of cloud services meant a higher number of server chips was also required.

Silicon manufacturing plants shutting down (covid related)/ damage forcing shutdowns (fires starting in manufacturing plants)

https://venturebeat.com/2021/04/01/global-chip-shortage-affects-more-than-cars/

- The shortage stems from a confluence of factors as carmakers, which shut plants during the COVID-19 pandemic last year, compete against the sprawling consumer electronics industry for chip supplies.
- IHS said a fire at a Japanese chip-making factory owned by Renesas Electronics Corp, which accounts for 30% of the global market for microcontroller units used in cars, has worsened the situation.

https://www.theguardian.com/technology/2021/apr/04/global-silicon-chip-shortage-hits-supply-of-phones-tvs-cars-and-australias-nbn

- A temporary shutdown in the production of silicon computer chips at the start of the coronavirus pandemic, as well as severe storms in Texas causing more recent delays, has caused worldwide chip shortages, with a knock-on effect for the production of phones, laptops and even automobiles.
- Samsung has indicated it could delay the release of the next Galaxy Note smartphone until 2022 as a result of the shortage. Apple, the world's biggest buyer of chips, was one of the worst affected companies, delaying the launch of the iPhone 12 last year as a result.

RESOURCES/ SOURCES:

https://numpy.org/

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

https://2.python-requests.org/en/master/

https://requests-html.kennethreitz.org/

https://www.sourcetoday.com/supply-chain/article/21867400/what-do-the-new-us-china-tariffs-mean-for-the-electronics-supply-chain

https://pcpartpicker.com/trends/price/video-card/

https://www.alliedmarketresearch.com/graphic-processing-unit-market

https://www.businesswire.com/news/home/20200904005427/en/Global-Computer-Hardware-Market-2020-to-2030---COVID-19-Impact-and-Recovery---ResearchAndMarkets.com

https://www.theverge.com/2021/1/7/22217206/nvidia-amd-gpu-trump-tax-china-tariff-exemption-expire

https://www.techpowerup.com/276858/gpus-to-see-price-increase-due-to-import-tariffs-other-pc-components-to-follow

https://www.fastcompany.com/90607876/why-is-there-a-silicon-chip-shortage-three-factors-are-to-blame

https://techmonitor.ai/techonology/hardware/global-chip-shortage-intel-tsmc-samsung

https://venturebeat.com/2021/04/01/global-chip-shortage-affects-more-than-cars/

https://www.theguardian.com/technology/2021/apr/04/global-silicon-chip-shortage-hits-supply-of-phones-tvs-cars-and-australias-nbn

https://www.ebay.ca/robots.txt

https://www.newegg.ca/robots.txt

APPENDIX:

Inside the folder “ITEC 4305 PC Parts Analysis Scripts and Other Files.” There are 3 files and 3 folders initially.

Both the scrapers take an input link, that are stored in the script itself. Both of these files can be run through the shell, assuming Python and the required libraries are installed (requests-html, requests, beautifulsoup). They create their respective folders with JSON and txt files each.

The JSONFormatter script assumes that both the others have been run in its entirety and its relevant files exist. It will not run otherwise. The files and required folder structure are provided. This script will also provide a JSON file with sorted data.

The files in the JSONFormatter Files folder assorted files we created. In its usual run, it will only create 1 file. We have provided 3 separate ones for clarity.

Also contained in the folder is our initial PowerPoint presentation that contains the slide contents in its entirety as opposed to the reduced version we are using to present. This file is only for reference purposes.