HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
FACULTY OF COMPUTER SCIENCE

o0o

PROJECT REPORT: LAPTOP PRICE ANALYSIS

Course: Introduction to Data Science - IT4142E
Class Code: 144108
Supervisor: Associate Prof Than Quang Khoat
Member of our group:

Trinh Hoang Giang	20214893	giang.th214893@sis.hust.edu.vn
Ho Ngoc Anh	20214877	anh.hn214877@sis.hust.edu.vn
Hoang Thanh Dat	20214899	dat.ht214889@sis.hust.edu.vn
Lang Van Quy	20214928	quy.lv214928@sis.hust.edu.vn
Vu Lam Anh	20214876	anh.vl214876@sis.hust.edu.vn

Ha Noi 2023

Contents

1 Introduction

In the ever-evolving landscape of laptops, understanding the intricate factors influencing prices is crucial for both consumers and businesses. This project employs advanced techniques to develop a model that accurately anticipates laptop prices based on essential features. Through comprehensive data exploration and collection, we analyze data to identify key factors affecting prices. The model development phase utilizes machine learning algorithms to capture nuanced relationships within the market. Evaluation and optimization processes ensure the model's reliability. Focused on a specific timeframe and geographic region, the project's scope aims to contribute to pricing efficiency. The successful implementation of this predictive model has broader implications, offering manufacturers, retailers, and consumers a valuable tool for informed decision-making in the dynamic and competitive laptop market. The report is structured to provide insights into data exploration, model development, and evaluation, fostering a deeper understanding of the pricing dynamics in the laptop industry.

2 Data Collection

In this project, one of the most well-known e-commerce sites,**Amazon**, will be mined for information about laptops.

(a) Scrapy library (b) ScrapeOps

Figure 1: Third party we use to collect data from Amazon

We use **Scrapy** library to scrape elements in laptop Amazon website page. The problem with scraping in amazon is amazon uses anti-bot protection to try and prevent developers from scraping their site. Thus, if we want to scrape Amazon reliably at large scale, we need to using rotating proxies, browser-profiles and possibly fortify your headless browser. To be easy and don't have to implement all this anti-bot bypassing logic, we use third party: **ScrapeOps Proxy Aggregator**. Now when we make requests with our scrapy spider they will be routed through the proxy and Amazon won't block them.

To be specific, here the overview data flow inside system:

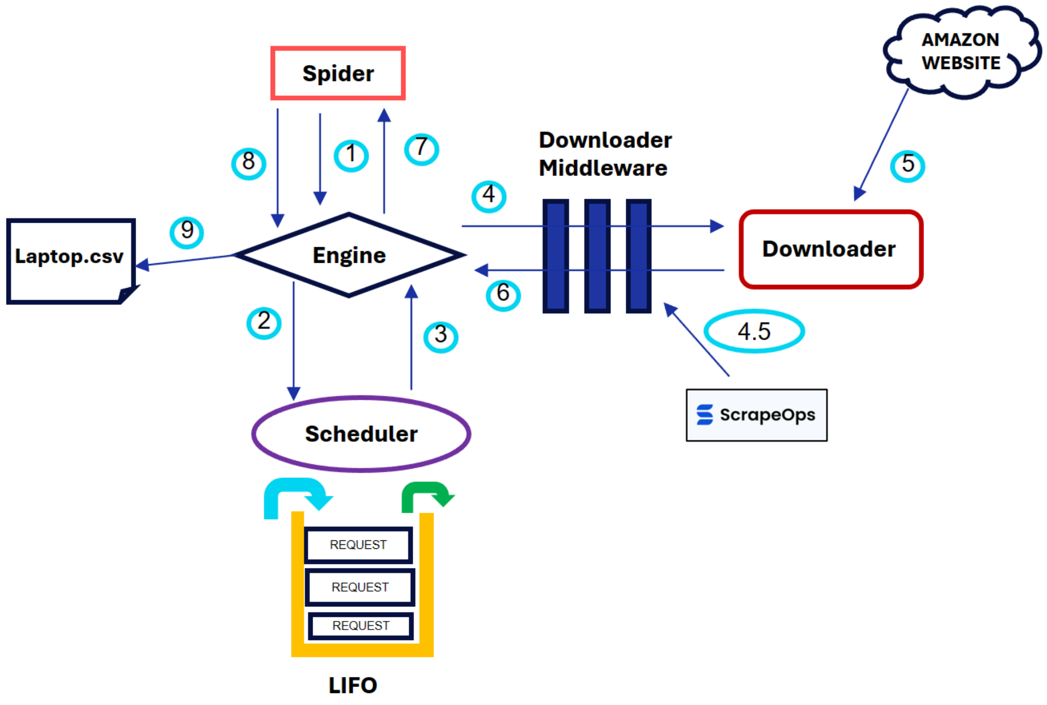


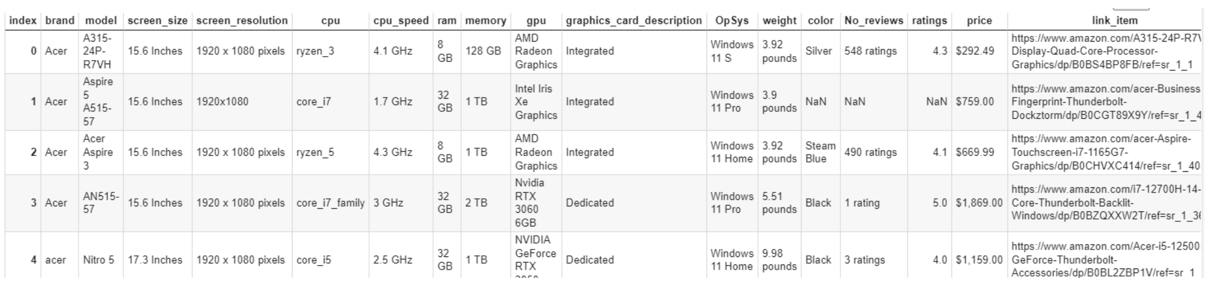
Figure 2: Architecture component and data flow inside system - Simplified version of scrapy architecture in source: https://docs.scrapy.org/en/latest/topics/architecture.html

- 1. Spider(Crawler) send request to Engine.
- 2. The Engine schedules the Requests in the Scheduler and asks for the next Requests to crawl.
- 3. The Scheduler returns the next Requests to the Engine (based on LIFO - Last In First Out mechanism)
- 4. Engine send the Request to the Downloader through Downloader Middleware.
- 4.5 Before the Request is sent to Downloader, we apply ScrapeOps Proxy Aggregator to route the proxy
- 5. Downloader fetch the content of website page(response) from Amazon website.
- 6. Downloader send the Response to the Engine
- 7. Engine send the Response to spider to scrape element from the Response
- 8. Spider return Scaped item and new Request to Engine
- 9. Scraped item will be save into the csv file.
- New request will repeat from step 2 until no more request from Spider

Below is the table of attribute which we scrape from the Amazon laptop product:

Attribute name	**Data type**	**Description**
brand	Categorical	Brand name of the laptop
model	Categorical	Model name of the laptop
screen size	Discrete numerical	Size of the laptop screen
screen resolution	Categorical	Resolution of laptop screen
cpu	Categorical	CPU brand of the laptop
cpu speed	Discrete numerical	CPU clock speed of the laptop
ram	Discrete numerical	RAM size of the computer
memory	Discrete numerical	ROM size of the computer
gpu	Categorical	GPU brand of the laptop
gpu description	Nominal categorical	GPU description of the laptop
OpSys	Categorical	Operating system of the laptop
weight	Continuous numerical	Weight of the laptop
color	Categorical	Color of the laptop
No_reviews	Continuous numerical	Number of reviews for that laptop
ratings	Continuous numerical	Ratings for that laptop in Amazon
price	Continuous numerical	Price of the laptop

3 Data Prep-rocessing



index	brand	model	screen_size	screen_resolution	cpu	cpu_speed	ram	memory	gpu	graphics_card_description	OpSys	weight	color	No_reviews	ratings	price	link_item
0	Acer	A315-24P-R7VH	15.6 Inches	1920 x 1080 pixels	ryzen_3	4.1 GHz	8 GB	128 GB	AMD Radeon Graphics	Integrated	Windows 11 S	3.92 pounds	Silver	548 ratings	4.3	$292.49	https://www.amazon.com/A315-24P-R7V Display-Quad-Core-Processor-Graphics/dp/B0BS4BP8FB/ref=sr_1_1
1	Acer	Aspire 5 A515-57	15.6 Inches	1920x1080	core_i7	1.7 GHz	32 GB	1 TB	Intel Iris Xe Graphics	Integrated	Windows 11 Pro	3.9 pounds	NaN	NaN	NaN	$759.00	https://www.amazon.com/acer-Business Fingerprint-Thunderbolt-Dockztorm/dp/B0CGT89X9Y/ref=sr_1_4
2	Acer	Acer Aspire 3	15.6 Inches	1920 x 1080 pixels	ryzen_5	4.3 GHz	8 GB	1 TB	AMD Radeon Graphics	Integrated	Windows 11 Home	3.92 pounds	Steam Blue	490 ratings	4.1	$669.99	https://www.amazon.com/acer-Aspire-Touchscreen-i7-1165G7-Graphics/dp/B0CHVXC414/ref=sr_1_40
3	Acer	AN515-57	15.6 Inches	1920 x 1080 pixels	core_i7_family	3 GHz	32 GB	2 TB	Nvidia RTX 3060 6GB	Dedicated	Windows 11 Pro	5.51 pounds	Black	1 rating	5.0	$1,869.00	https://www.amazon.com/i7-12700H-14-Core-Thunderbolt-Backlit-Windows/dp/B0BZQXXW2T/ref=sr_1_3
4	acer	Nitro 5	17.3 Inches	1920 x 1080 pixels	core_i5	2.5 GHz	32 GB	1 TB	NVIDIA GeForce RTX	Dedicated	Windows 11 Home	9.98 pounds	Black	3 ratings	4.0	$1,159.00	https://www.amazon.com/Acer-i5-12500 GeForce-Thunderbolt-Accessories/dp/B0BL2ZBP1V/ref=sr_1

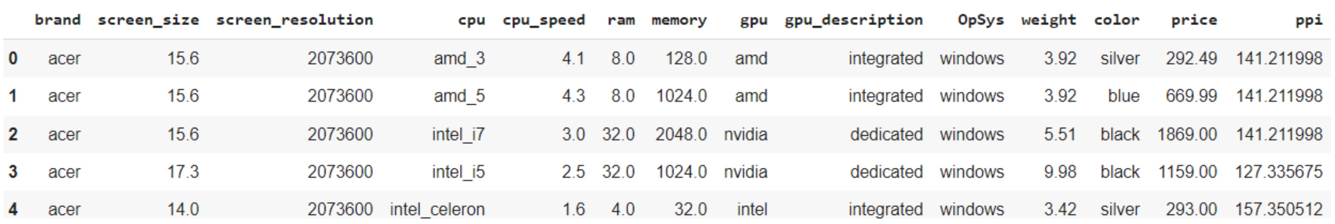
Figure 3: Raw data after crawling

We standardize the data through multiple steps. First, since the "link_item", "model", "no_reviews" and "ratings" columns are superfluous to our analysis and forecast, we remove them. This can be considered as a step of feature engineering. After that, we lowercase every value and get rid of duplicates and missing data. We convert data with units of measure into the same unit and eliminate the unit. We create a general form for string data synchronization. These values are combined into one even though they have different representations but the same meaning.

Additionally, we change the "screen_resolution" column, which is expressed as "width x height," into two new columns: "ppi," which stands for pixels per inch, and "screen_resolution," which is expressed as width * height(total pixels).

$$PPI = \frac{\sqrt{width^2 + height^2}}{screen\_size} \tag{1}$$

We substitute ”other” for values with few instances for categorical attributes. In more detail, we classify as ”other” all CPU types with a number of instances less than 12. This is indicated in the ”cpu” column. We maintain ”intel”, ”nvidia”, ”amd”, and ”apple” as the primary categories for the ”gpu” column, with other categories falling under ”other”. The colors that are not ”black,” ”blue,” ”grey,” ”silver,” ”gold,” or ”white” are likewise referred to as ”other.”



	brand	screen_size	screen_resolution	cpu	cpu_speed	ram	memory	gpu	gpu_description	OpSys	weight	color	price	ppi
0	acer	15.6	2073600	amd_3	4.1	8.0	128.0	amd	integrated	windows	3.92	silver	292.49	141.211998
1	acer	15.6	2073600	amd_5	4.3	8.0	1024.0	amd	integrated	windows	3.92	blue	669.99	141.211998
2	acer	15.6	2073600	intel_i7	3.0	32.0	2048.0	nvidia	dedicated	windows	5.51	black	1869.00	141.211998
3	acer	17.3	2073600	intel_i5	2.5	32.0	1024.0	nvidia	dedicated	windows	9.98	black	1159.00	127.335675
4	acer	14.0	2073600	intel_celeron	1.6	4.0	32.0	intel	integrated	windows	3.42	silver	293.00	157.350512

Figure 4: Preprocessed Data

There are 2435 instances of processed data and 5559 instances of raw data.

4 Exploratory Data Analysis

4.1 Outlier Detection and Removal

We use box plots and scatter plots for numerical features to display and identify outliers. Furthermore, we think that values that fall outside of the range $(mean - 3 \times std, mean + 3 \times std)$ are probably outliers.

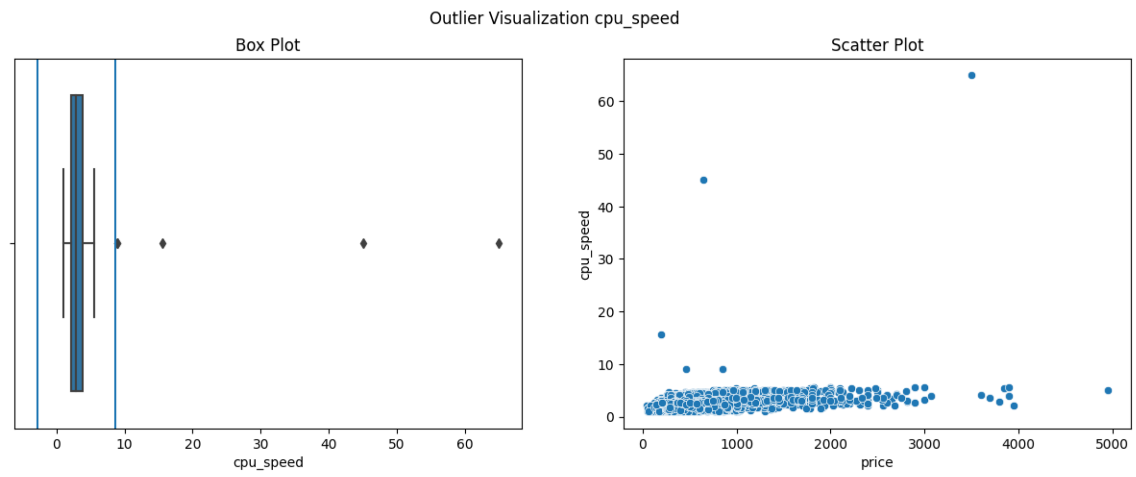


Figure 5: Box Plot and Scatter Plot of CPU speed

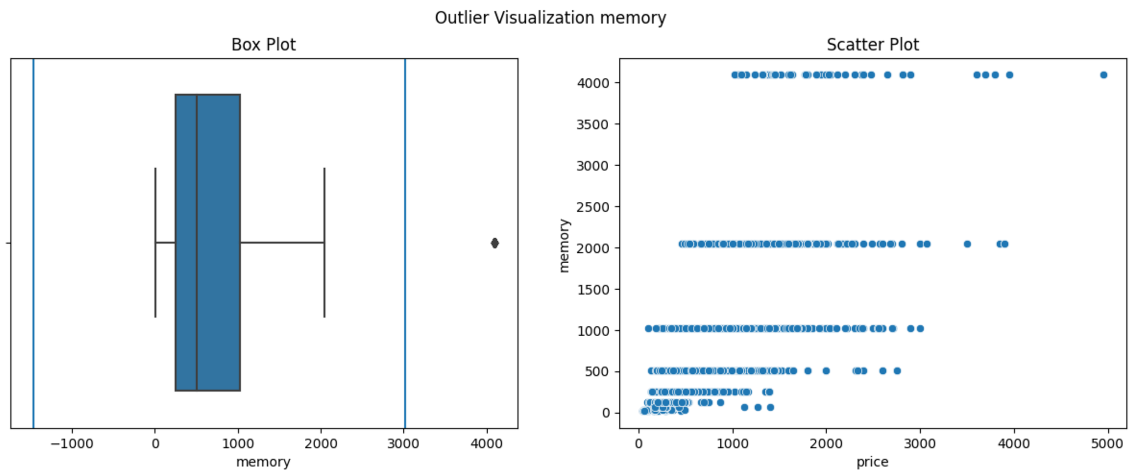


Figure 6: Box Plot and Scatter Plot of Memory

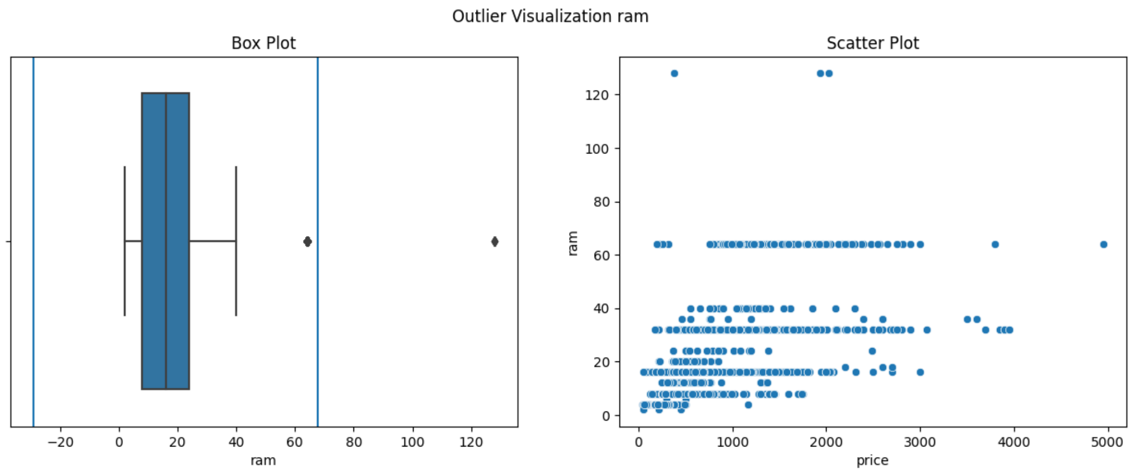


Figure 7: Box Plot and Scatter Plot of RAM

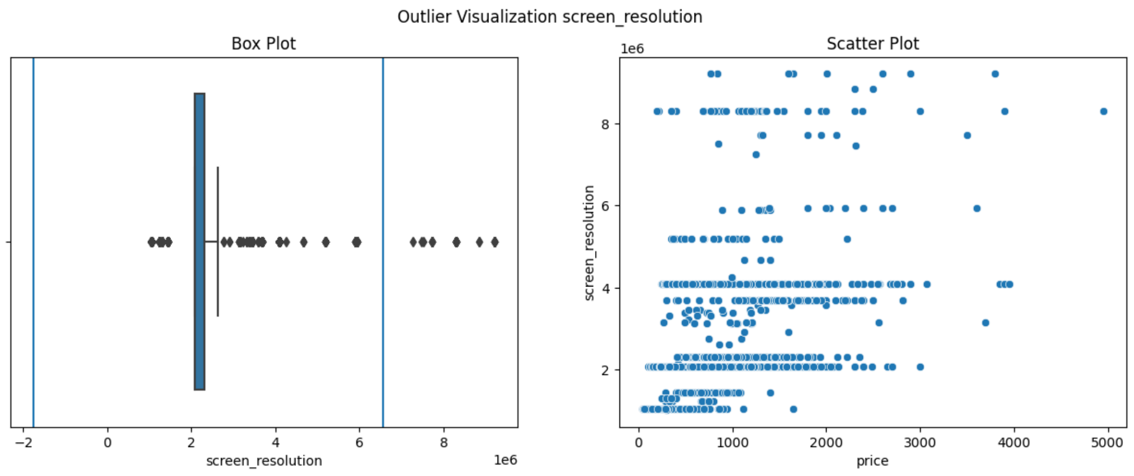


Figure 8: Box Plot and Scatter Plot of Screen resolution

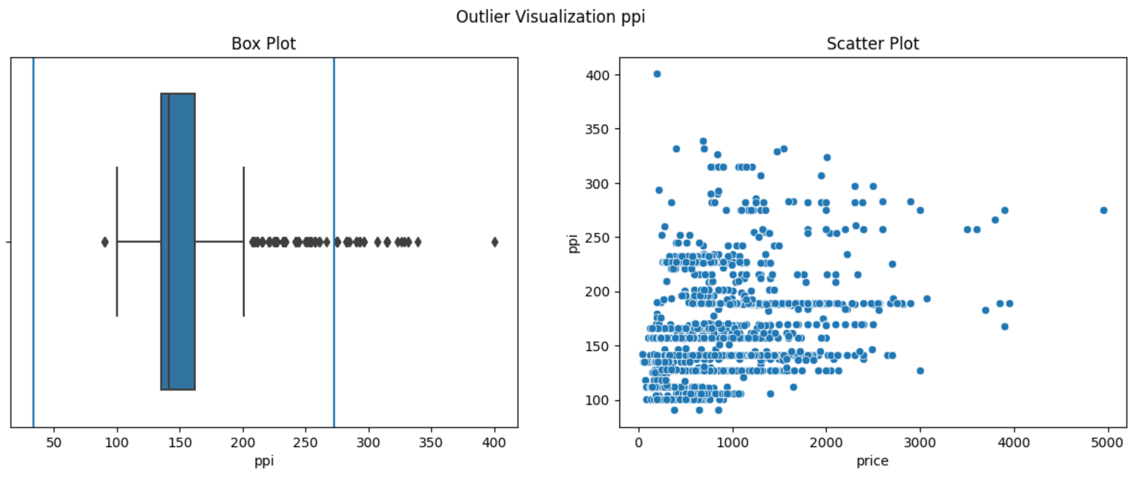


Figure 9: Box Plot and Scatter Plot of PPI

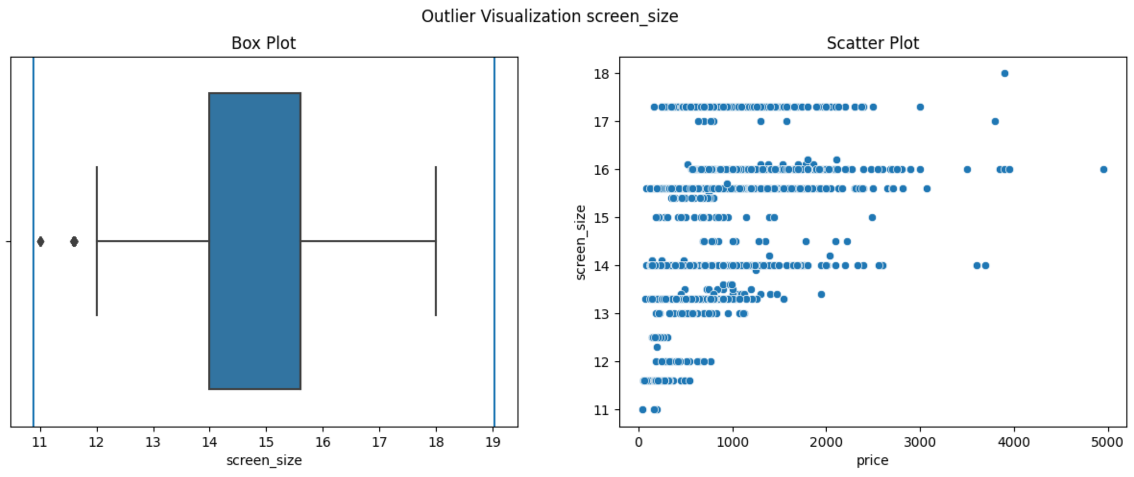


Figure 10: Box Plot and Scatter Plot of Screen size

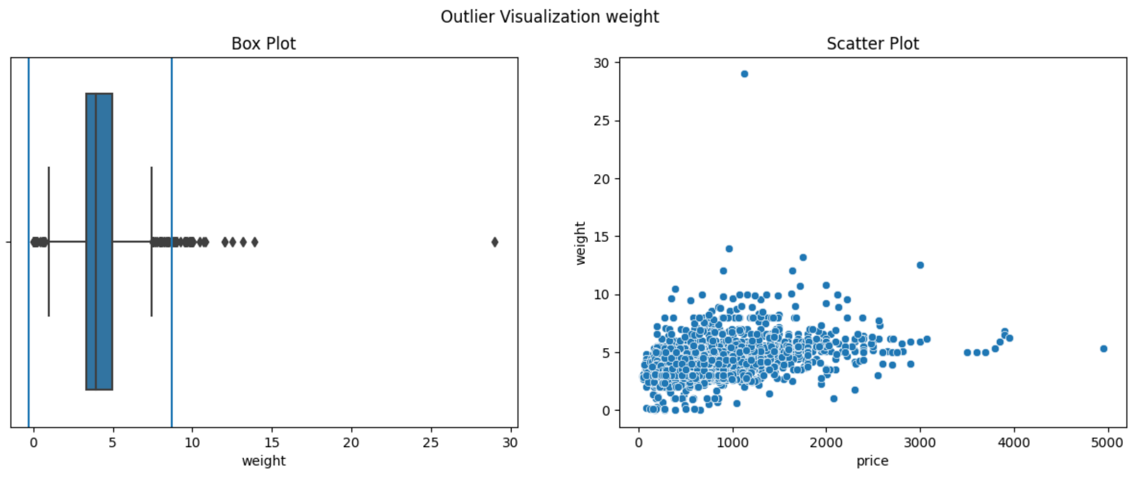
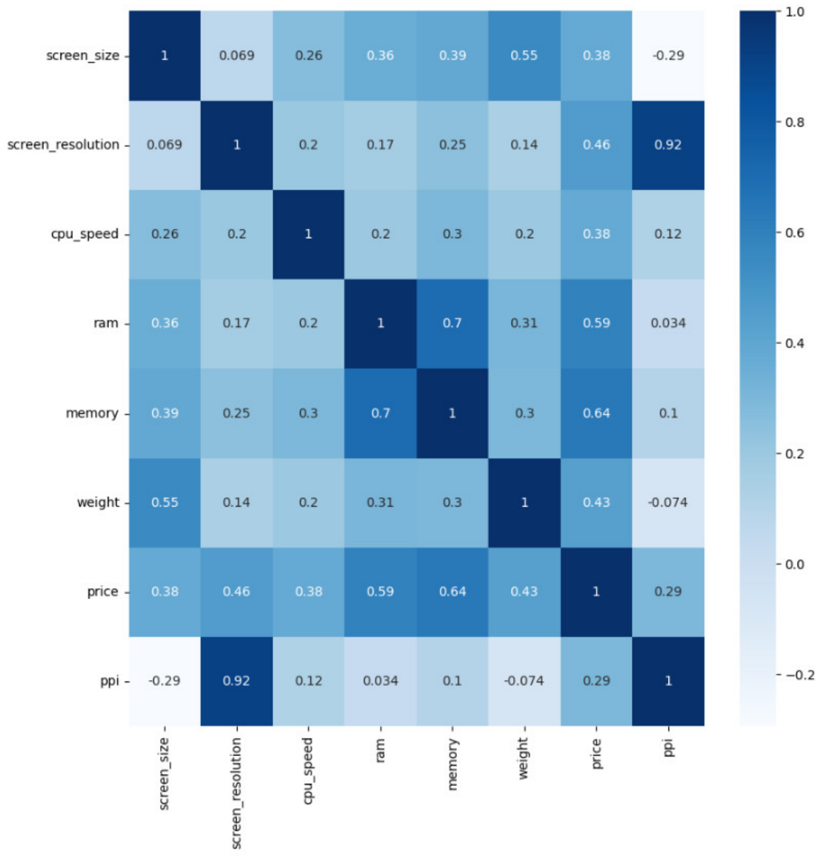


Figure 11: Box Plot and Scatter Plot of Weight

Note that the green vertical lines in the box plots represent $mean - 3 \times std$ and $mean + 3 \times std$ In conclusion, we eliminate rows with CPU speed>40, RAM>80, PPI>350 and weight>15 based on these presumptions. Our data set is reduced to 2427 by eliminating outliers.

4.2 Data Visualization

Our first concern focuses on numeric attributes. Because the number of numerical attributes is quite large, we would first look at the heatmap of correlation between each numerical attribute to each other :



	screen_size	screen_resolution	cpu_speed	ram	memory	weight	price	ppi
screen_size	1	0.069	0.26	0.36	0.39	0.55	0.38	-0.29
screen_resolution	0.069	1	0.2	0.17	0.25	0.14	0.46	0.92
cpu_speed	0.26	0.2	1	0.2	0.3	0.2	0.38	0.12
ram	0.36	0.17	0.2	1	0.7	0.31	0.59	0.034
memory	0.39	0.25	0.3	0.7	1	0.3	0.64	0.1
weight	0.55	0.14	0.2	0.31	0.3	1	0.43	-0.074
price	0.38	0.46	0.38	0.59	0.64	0.43	1	0.29
ppi	-0.29	0.92	0.12	0.034	0.1	-0.074	0.29	1

Figure 12: Heatmap of correlation between numerical attribute

By observing the coefficient in heatmap of correlation between each numerical attribute in Figure 12 , we can conclude some relationships as follow:

- All numerical attribute has positive correlation with price and the magnitude is also quite large, value range from 0.29 to 0.64. So the price all depends on these attributes.
- RAM size and ROM size have large positive correlation. Ram size is dependent on ROM size.
- Screen size is dependent on weight, which is easy to understand.

In the next section, we will analyze price, and examine particularly at each attribute and its relationship with that attribute.

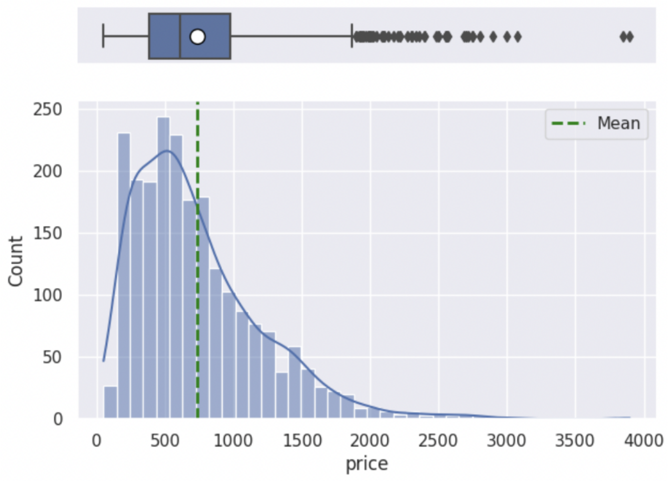


Figure 13: Distribution of Price

The present research data showed that this attribute Price has many outliers beyond the 75% quantiles. Another point told is that the distribution of price over all product is not located at the middle but is left-shifted.

Above figures shows that the most common price lies between about 400-600 dollars. The lowest price to afford a laptop from those sites is 49.99 dollars and the most luxury laptop costs 3898 dollars.

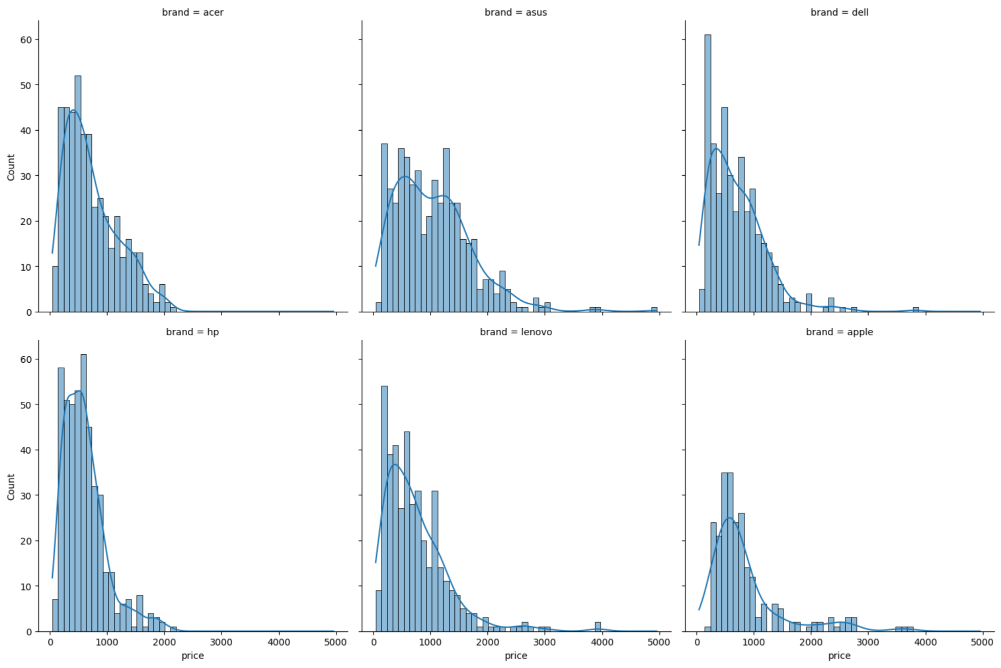


Figure 14: Price distribution corresponding to brands

The majority of computers from these brands are priced affordably, predominantly falling within the 0$ to 2000$ range. This strategic pricing approach caters to a diverse consumer base with varied budget considerations. It aligns with prevailing market trends, where a significant portion of computer purchases is concentrated within this accessible price spectrum.

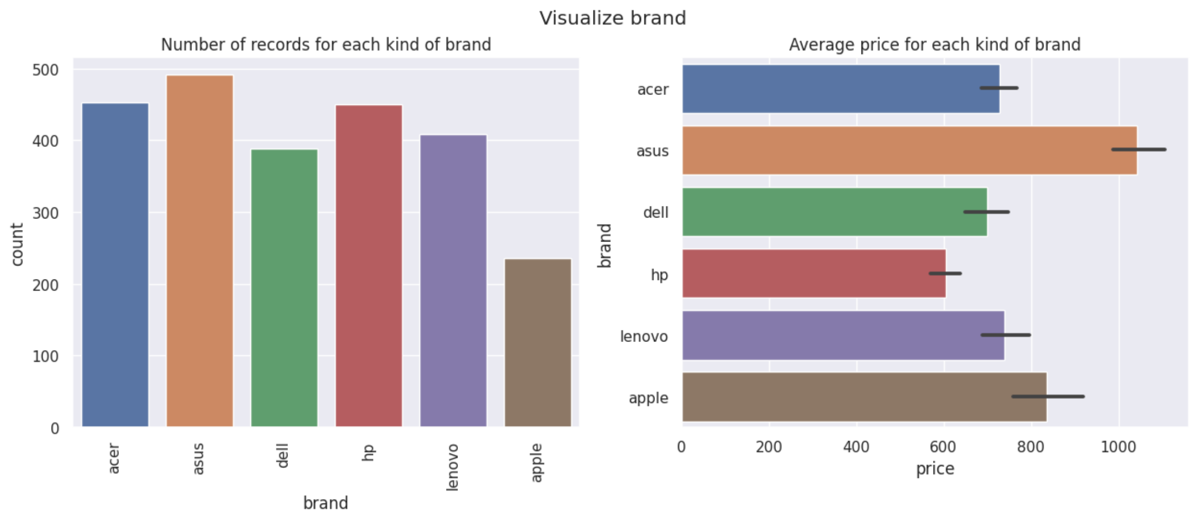


Figure 15: Brand visualization with number of records and average price

Apple produces approximately half the number of laptops compared to other brands. On average, an ASUS laptop costs around $1000, whereas the average cost of laptops from other brands is approximately $800. This disparity in manufacturing quantities and pricing highlights the distinct market position and pricing strategy employed by Apple in contrast to its competitors, with a focus on a more exclusive and premium market segment.

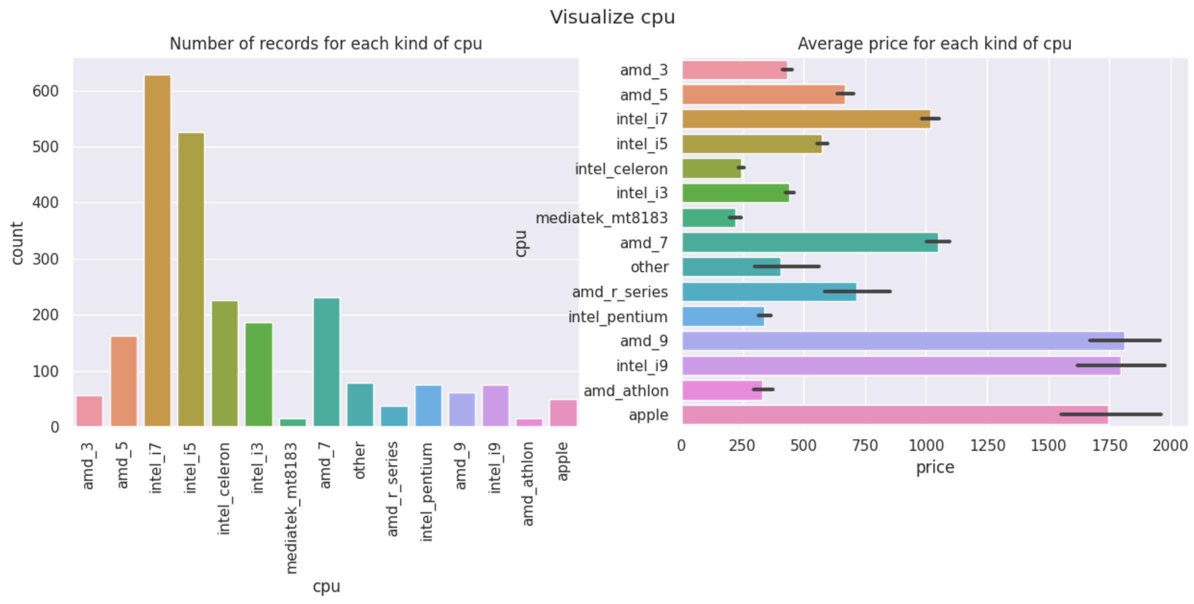


Figure 16: CPU visualization with number of records and average price

The predominant CPUs in the market are from the Intel i5 and i7 series. Notably, the CPUs with the highest average prices belong to Apple, AMD 9, and Intel i9. The Intel i5 and i7 series represent a widely adopted standard in computing, serving as the backbone for numerous devices across various brands. However, when it comes to premium and cutting-edge technology, the Apple, AMD 9, and Intel i9 CPUs emerge as leaders, commanding a higher price point due to their superior performance, innovative features, and stateof-the-art capabilities.

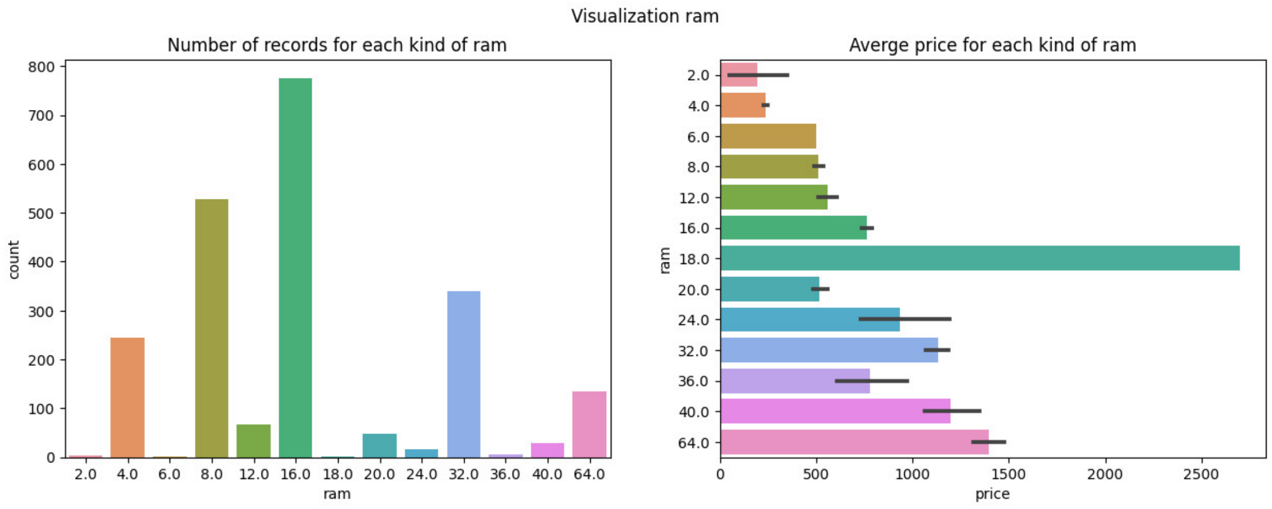


Figure 17: RAM visualization with number of records and average price

The RAM with value 16 GB has the most number of records. Otherwise, number of records RAM with value: 36, 2 GB, 6 GB or 18 GB is extremely small. The reason maybe the default ram recently now is 16.0, 8.0, 32.0,.... But with value 36, 2, 6 or 18, that laptop maybe a customized laptop.With the price relationship, except the ram is 18 GB (perhaps we have outlier here), we can see trend that the price tends to increase when ram size is increase.

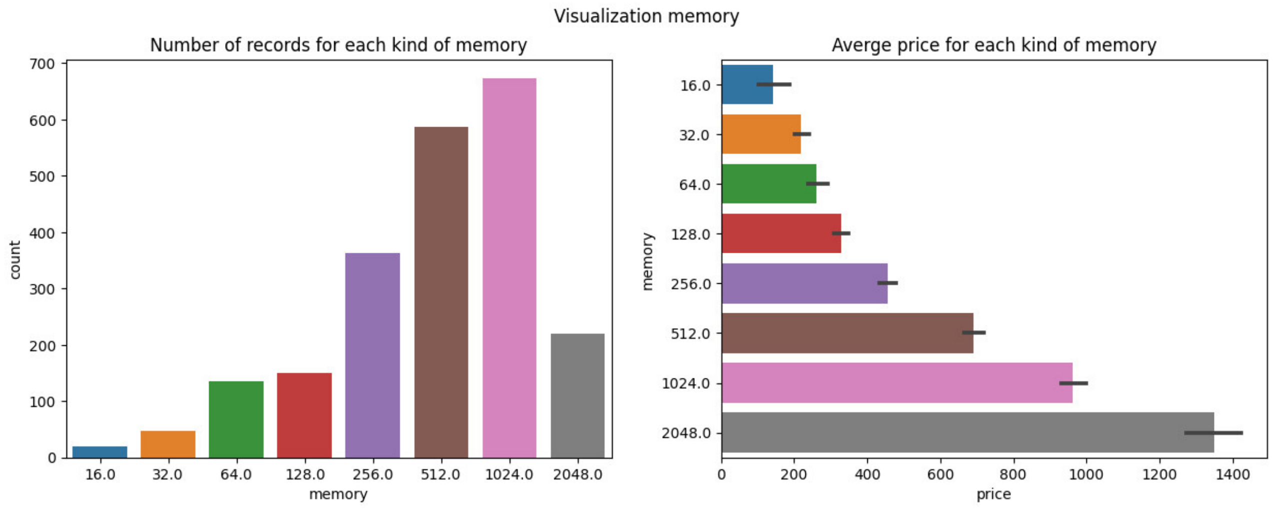


Figure 18: ROM visualization with number of records and average price

We observe a very large concentration of records in the memory range from 256 GB to 2048 GB. Conversely, the number of records with smaller memory is much less. This is suitable because it will meet the needs of modern computers, helping people store more.

A parallel observation can be made when examining the relationship between memory size and laptop prices. Similar with RAM size, we can see very clearly with ROM-Average Price graph, As the memory size increases, there is a discernible upward trend in laptop prices., when memory size increases the laptop price is also increasing.

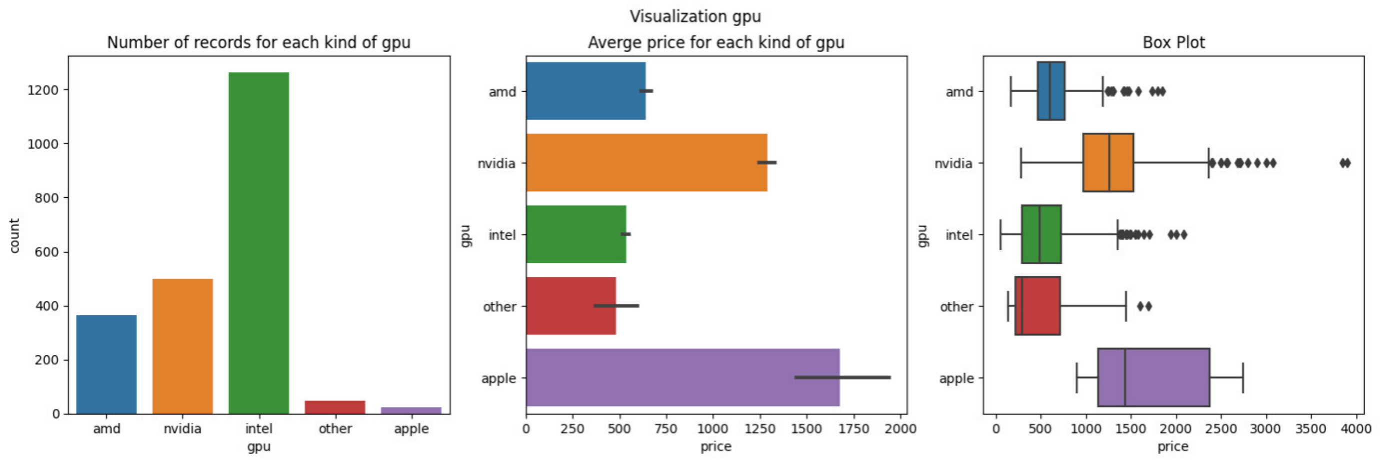


Figure 19: GPU visualization with number of records and laptop price

The laptop has intel GPU has large records and laptop with apple GPU has very small number of records.But We can see that apple GPU laptop has the range of price is larger than the others and also median price is largest so we can observe that laptop with apple GPU usually have the high price and range of value is large.Otherwise,we observe that intel has the most data but the price is avargely small,reaons maybe that intel GPU is usually intergrated GPU.Alternatively, with nvidia GPU, the averge price is quite high (>1250 $) although the records is not too many. The reason is that the nvidia GPU usually is dedicated GPU.

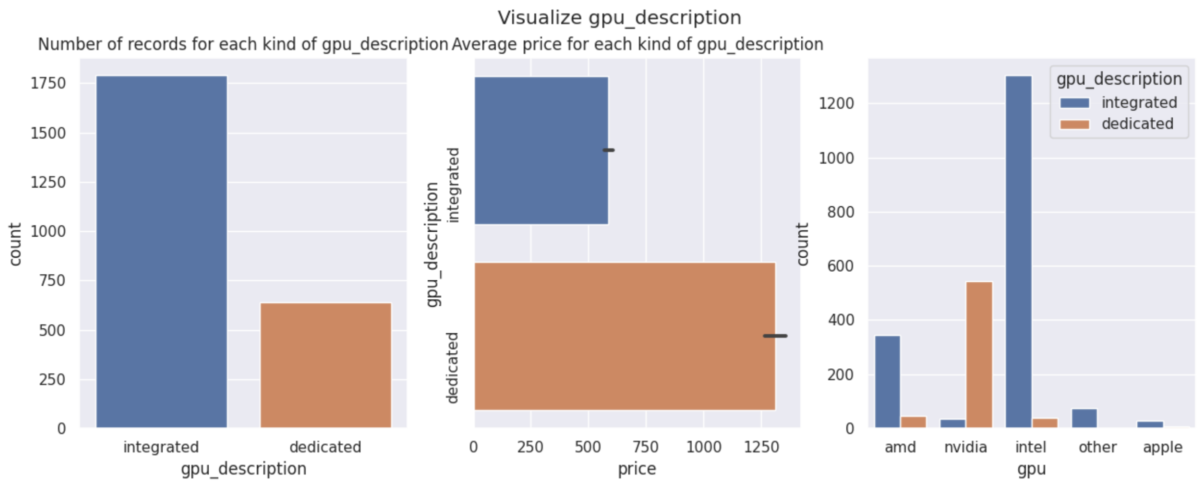


Figure 20: GPU description visualization with number of records and average price

We find a notable distinction between laptops equipped with dedicated GPU and those featuring integrated GPU. Specifically, the number of laptops with dedicated GPU appears nearly one-third of the total, indicating a prevalence of integrated graphics solutions in the market. And laptops equipped with dedicated GPU cost significantly more than integrated GPU.

The observed distribution highlights a trade-off in the market, where users opting for dedicated GPU are likely to prioritize enhanced graphics performance, often associated with more demanding tasks such as gaming or graphic design. Conversely, laptops with integrated GPU offer a cost-effective solution for users with less graphics-intensive needs.

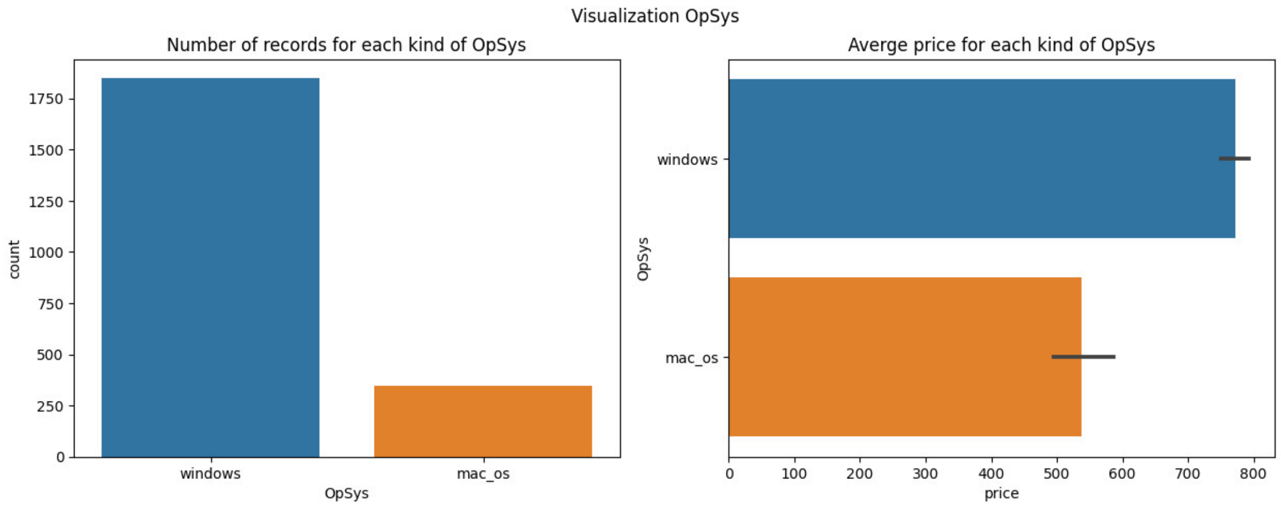


Figure 21: GPU description visualization with number of records and average price

The number of laptop whichs has windows operating systems is much larger than MACOS. This simply understandable since except MacBook, all other laptop use windows operatings system.We can observe maybe strange here since as we have known laptop with MACOS (MacBook) is much expensive than other laptop. The reason we think for this case maybe: The website amazon has large of old version MAC which has low price so it makes the averge price of the MAC is much decrease.Otherwise with Windows, as we mention, laptop maybe have dedicated gpu (like nvidia) so the price maybe higher.

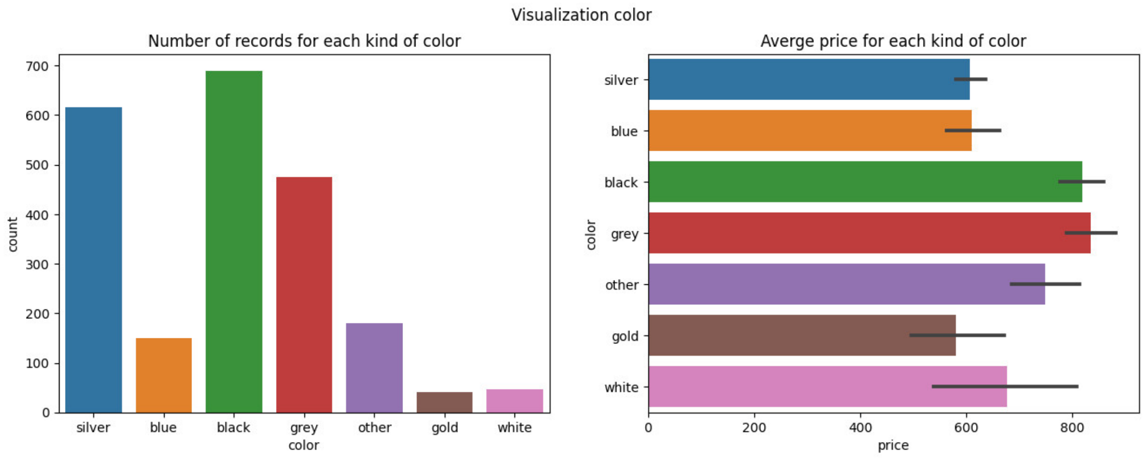


Figure 22: GPU description visualization with number of records and average price

We see that black and grey laptops command higher prices on average. This observation suggests that while silver, black, and grey laptops enjoy greater popularity, the pricing differences among color options are generally subtle.

5 Modelling

5.1 Linear Model

Linear regression is a statistical modeling technique used to analyze the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the variables and aims to find the best-fitting line that represents the data. By estimating the coefficients of the regression equation, we can make predictions and understand the impact of the independent variables on the dependent variable.

5.2 K-Nearest Neighbors

The k-Nearest Neighbors (k-NN) algorithm is a fundamental and intuitive concept in machine learning and pattern recognition. Developed for both classification and regression tasks, k-NN is a non-parametric, instance-based learning algorithm that makes predictions based on the similarity of data points in a given dataset. The central idea behind k-NN is to predict the class or value of a data point by examining its k nearest neighbors in the feature space. The proximity is typically measured using distance metrics such as Euclidean distance, Manhattan distance, or Minkowski distance. The k-NN algorithm operates on the assumption that similar data points tend to have similar outcomes.

5.3 Decision Tree

Decision tree is a supervised machine learning algorithm that is used for both classification and regression tasks. It works by recursively partitioning the data into subsets based on the values of input features, making decisions at each node of the tree. The structure of a decision tree resembles an inverted tree, where each internal node represents a decision based on a feature, each branch represents the outcome of the decision, and each leaf node represents the final predicted output or class label.

5.4 Random Forest

Random Forest Regression is a supervised machine learning algorithm that is used for regression problems. It is a type of ensemble learning that combines multiple decision trees to make predictions. The idea behind Random Forest Regression is to create many decision trees on different subsets of the We employed the R2 score and MAE as our primary assessment metrics, considering their suitability for regression tasks. During the model tuning phase, these metrics played a crucial role in guiding adjustments to improve overall performance. In addition to R2 score and MAE, we also utilized MSE and RMSE to provide a more comprehensive evaluation of our models. This multi-metric approach allowed us to assess accuracy from different angles. In our final analysis, we compared models based on all four metrics, gaining valuable insights into their respective strengths and weaknesses.

5.5 Multi-Layer Perceptron

A Multilayer Perceptron (MLP) is a type of artificial neural network used for classification and regression. It consists of layers of interconnected nodes, or neurons, including an input layer, hidden layers, and an output layer. Neurons apply weighted sums and activation functions to input data, and the network is trained through forward and backward passes, adjusting weights and biases to minimize prediction errors. MLPs are versatile, handling complex patterns, but tuning hyperparameters is crucial. They are widely applied in tasks like image recognition, natural language processing, and time series prediction.

5.6 Ensemble Model

Finally, we implement an ensemble model including all model above: Ridge Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Multilayer Perceptron (MLP). Each model contributes unique strengths to enhance overall predictive accuracy. Ridge Regression brings stability and guards against overfitting, while KNN captures local patterns. Decision Tree manages complex decision boundaries, and Random Forest leverages ensemble learning for robustness. The MLP neural network excels at capturing non-linear relationships. This ensemble capitalizes on the diversity of these models, offering a comprehensive solution for accurate laptop price predictions. By synergizing various approaches, our ensemble model provides a well-balanced and effective strategy to navigate the intricacies of laptop pricing, ensuring reliability and precision in forecasting.

6 Experiments and Results

6.1 Training and Testing Setup

In the setup for training and testing a model, it is essential to preprocess the data appropriately to ensure effective learning and evaluation. Specifically, prior to training, categorical attributes should undergo one-hot encoding, while numerical attributes benefit from log normalization.

One-hot encoding is a technique used to represent categorical variables as binary vectors, thereby enabling the model to interpret and learn from these attributes more effectively. On the other hand, log normalization transforms numerical features by applying the common logarithm, mitigating the impact of extreme values and promoting a more normalized distribution, which can enhance the model's performance.

To evaluate the model's performance robustly, a cross-validation approach with 5 folds is employed. Cross-validation involves partitioning the dataset into multiple subsets, training the model on different combinations of these subsets, and then assessing its performance across various validation sets. The use of 5 folds ensures a balanced and comprehensive assessment of the model's generalization capabilities.

6.2 Linear Model

In this project, we model Linear Model as a Ridge Regression problem. We predict laptops' price based on items' feature vectors. We modified the hyperparameter lambda and derive the Graph below:

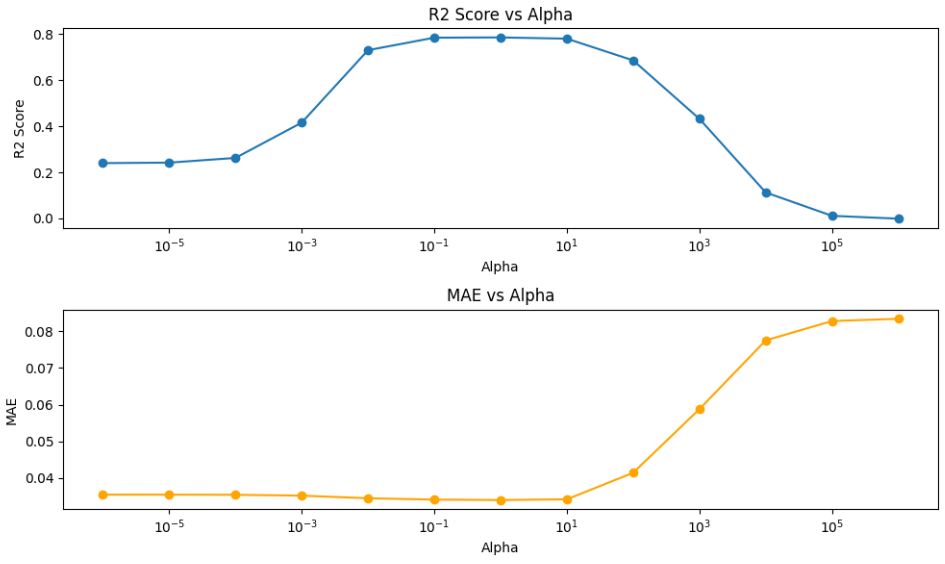


Figure 23: R2 Score and MAE vs Alpha

The error graph shows that error remains stable when lambda reach 10^{-1} and decrease after exceeding 10^1. Hence, we choose $\lambda = 1$ as the best option based on the findings in the graph.

6.3 K-Nearest Neighbors

In KNN model, the hyperparameter is the K most similar neighbors used to predict unknown price. Hence, we adjust the number of neighbors to get the ideal value for this hyperparameter.

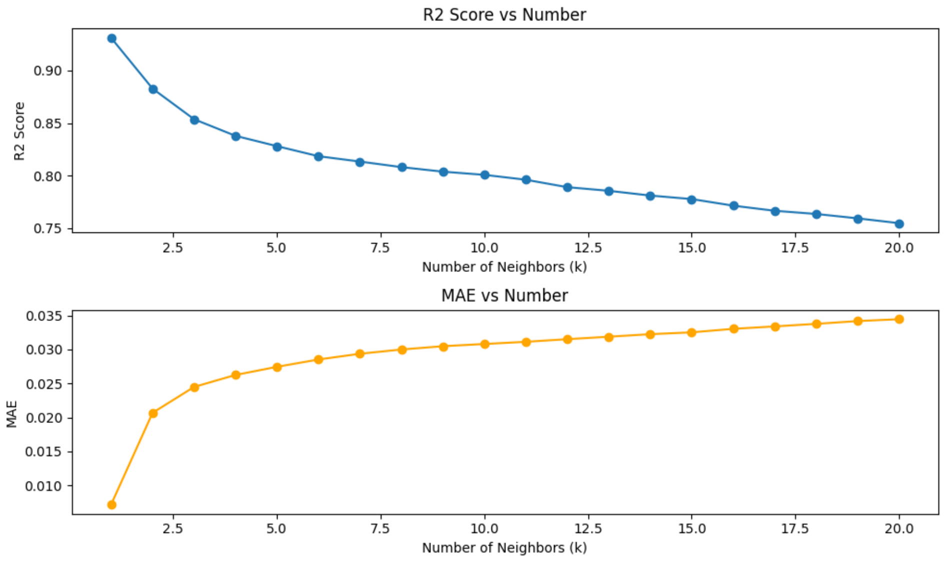


Figure 24: R2 Score and MAE vs No. Neighbors

Based on the results in the graph, we decide that *neighbor* = 25 is the best option, which maximize R2 Score and minimize MAE.

6.4 Decision Tree

The hyperparameter in the KNN model is max depth. Therefore, we adjust the hyperparameter to get the following outcomes. This plot, which illustrates the model's performance at different maximum depths, will help us establish the optimal depth for our decision tree.

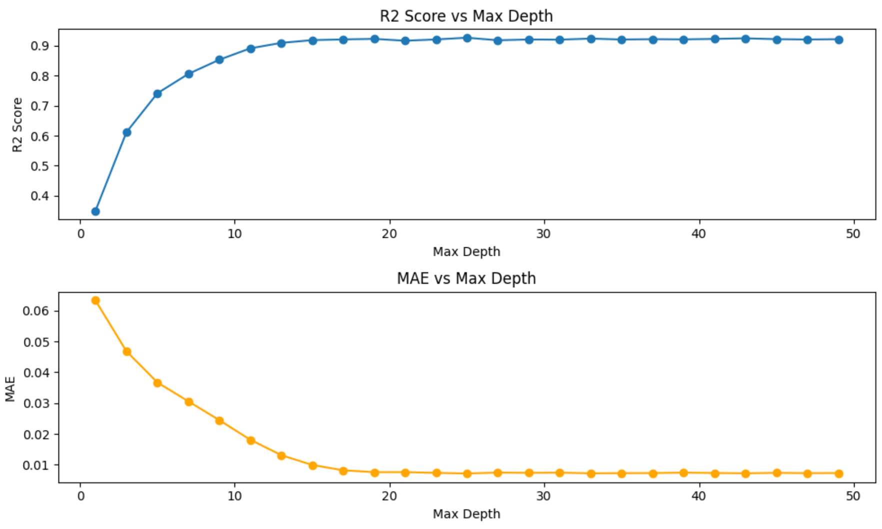


Figure 25: R2 Score and MAE vs Max Depth

The error graph demonstrates that measurements hold steady till the max depth reaches 25. Therefore, based on the information in the graph, we determine that $max\_depth = 25$ is the optimal choice.

6.5 Random Forest

Similar to Decision Tree, the hyperparameter in Random Forest is max depth. Therefore, we adjust the hyperparameter to get the following outcomes. This plot, which illustrates the model's performance at different maximum depths, will help us establish the optimal depth for our decision tree.

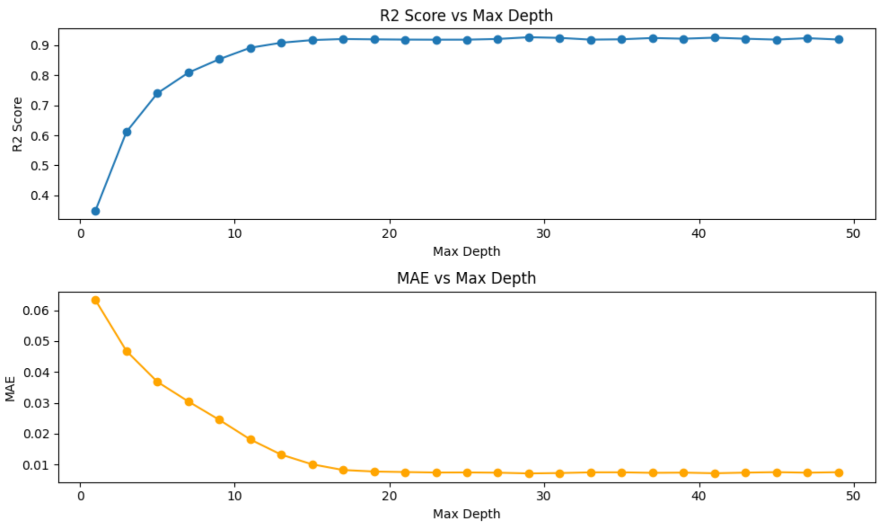


Figure 26: R2 Score and MAE vs Max Depth

The error graph demonstrates that measurements hold steady till the max depth reaches 25. Therefore, based on the information in the graph, we determine that $max\_depth = 25$ is the optimal choice.

6.6 Multi-Layer Perceptron

In MLP model, the hyperparameter is the size of hidden layers. In order to obtain the performance results, we modify the size of the hidden layers. This plot will assist us in determining the ideal hyperparameter for our MLP model since it shows the model's performance at various hidden layer sizes.

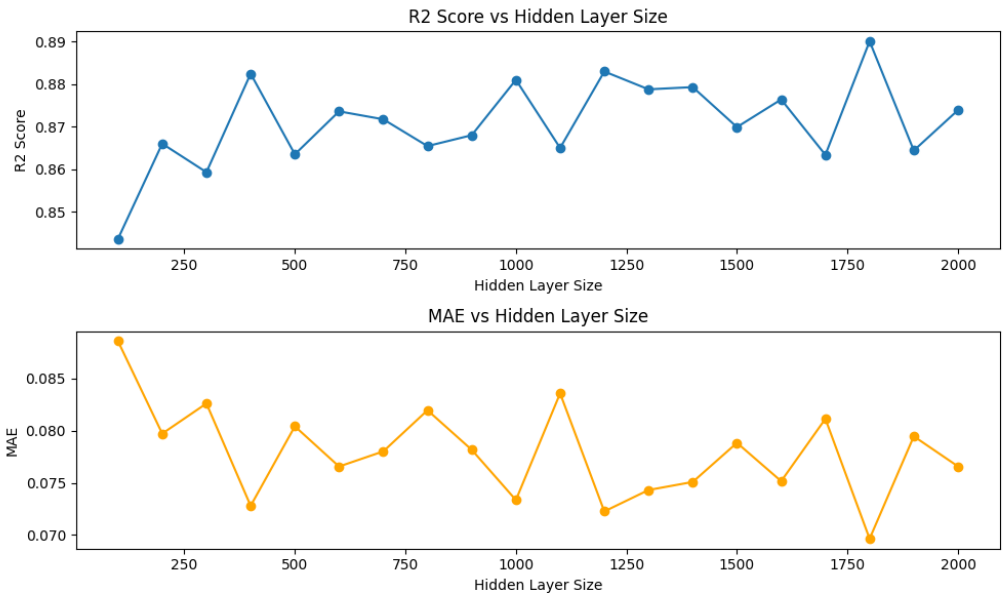


Figure 27: R2 Score and MAE vs Hidden Layer Size

Based on the results in the graph, we decide that $hidden\_layer\_size = 1800$ is the best option, which maximize R2 Score and minimize MAE.

6.7 Comparative Analysis

In summary, the results of all models tuned with optimum hyperparameters are shown in the Table below:

Metrics	R2 Score	MAE	MSE	RMSE
Ridge Regression	0.816	0.098	0.018	0.132
KNN	0.943	0.023	0.006	0.044
Decision Tree	0.944	0.024	0.005	0.04
Random Forest	0.954	0.041	0.004	0.06
MLP	0.89	0.07	0.011	0.102
Ensemble	0.945	0.048	0.005	0.069

Key Insights and Observations:

1. Ridge Regression:
   - R2 Score of 0.816 suggests a reasonably good explanatory power.
   - Moderate MAE (0.098) and RMSE (0.132) indicate decent predictive accuracy.
2. KNN (K-Nearest Neighbors):
   - Impressive R2 Score of 0.943, highlighting strong model performance.
   - Minimal MAE (0.023) and RMSE (0.044) demonstrate high precision in predictions.
3. Decision Tree:
   - Similar to KNN, Decision Tree exhibits a high R2 Score (0.944).
   - Remarkably low RMSE (0.040) indicates precise prediction capability.
4. Random Forest:
   - The highest R2 Score among all models (0.954) underscores exceptional explanatory power.
   - While exhibiting a slightly higher RMSE (0.060), the model showcases robust predictive accuracy.
5. MLP (Multi-Layer Perceptron):
   - R2 Score of 0.890 suggests good performance, though slightly below some other models.
   - Higher MAE (0.070) and RMSE (0.102) indicate some variance in predictions.
6. Ensemble:
   - Strong overall performance with a high R2 Score of 0.945.
   - Moderate MAE (0.048) and RMSE (0.069) indicate reliable predictive accuracy.

Conclusion:

- The Random Forest model stands out with the highest R2 Score, emphasizing its superior ability to explain variance in the data.
- Decision Tree and KNN also demonstrate robust performance across all metrics.
- Consideration of both explanatory power (R2 Score) and prediction precision (MAE, RMSE) is vital for selecting the most suitable model for the specific task at hand.

7 Conclusion and Future work

In conclusion, the performance analysis of various machine learning models has provided valuable insights into their effectiveness for the given predictive task. Each model exhibits strengths in different aspects, and the choice of the most suitable model depends on the specific requirements and priorities of the application.

The Random Forest model emerges as a frontrunner, demonstrating the highest R2 Score, indicating excellent explanatory power, and a robust overall predictive accuracy. KNN and Decision Tree models also showcase strong performance, particularly in terms of precision, with low RMSE values. Ridge Regression, MLP, and the Ensemble model present competitive results, each with its own set of advantages and considerations.

The decision-making process should involve a careful trade-off between the interpretability of the model, computational efficiency, and the specific goals of the predictive task. Moreover, the analysis highlights the importance of considering multiple metrics to gain a holistic understanding of model performance.

In the future, we can focus on enhancing the models' applicability, robustness, and ethical considerations. A specific avenue for future work is real-world validation: We will validate our models on external datasets or in a real-world setting to assess their generalization capabilities and robustness.

8 References

References

[1] Hilt, Donald E.; Seegrist, Donald W. (1977). Ridge, a computer program for calculating ridge regression estimates

[2] Scikit-learn: Machine Learning in Python

[3] Python Scrapy: Build A Amazon Products Scraper with ScrapeOps[2023]

[4] Scrapy 2.11 documentation

9 Work distribution

1. Proposal: Giang
2. Data Crawling: Lâm Anh
3. Data preprocessing: Lâm Anh, Giang, Ánh
4. Exploratory Data Analysis: Lâm Anh, Giang, Ánh
5. Implement prediction models: Dat, Quý
6. Evaluating Model: Dat, Quý
7. Demo: Lâm Anh
8. Report:
   (a) Introduction: Dat
   (b) Data Collection: Lâm Anh

   (c) Data Preprocessing: Giang
   (d) EDA: Lâm Anh, Giang, Ánh
   (e) Modeling: Quý
   (f) Experiment and Result: Dat

9. Slide: Giang, Ánh, Lâm Anh, Quý, Dat

10. Presentation
    (a) Introduction: Lâm Anh
    (b) Data Crawling: Lâm Anh
    (c) Data Preprocessing: Giang
    (d) Data Exploration: Ánh
    (e) Modeling: Quý
    (f) Experiment and result: Dat