University of Science and Technology of Hanoi

DS3.001: Fundamental of Data Science Report

Laptop Selection Guide: Data-Driven Recommendations with K-Nearest Neighbors

Group members

22BI13199	Bui Nguyen Ngoc Huyen
22BI13320	Cao Nhat Nam
22BI13120	Nguyen Duc Duy

TABLE OF CONTENTS

1. Introduction

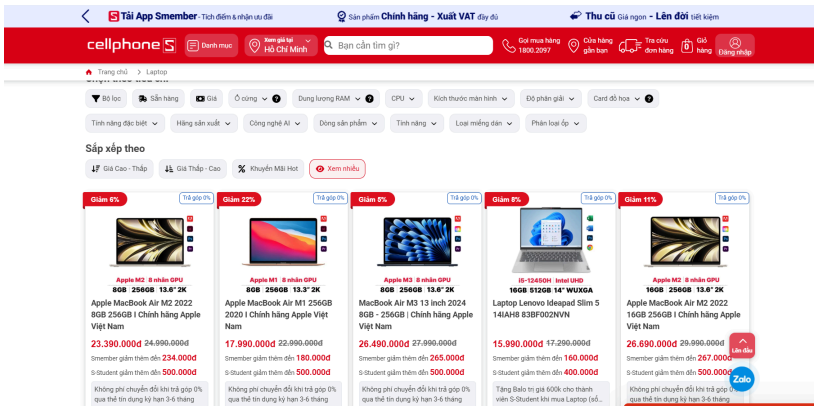
Choosing the right laptop can be challenging with so many models, features, and price ranges available. This project uses a data-driven approach to simplify this decision by applying KNN algorithm on key laptop specifications, such as processor, RAM, storage, and price. By clustering laptops with similar features, we can provide clear, categorized recommendations tailored to different needs—whether for budget-conscious students, multitaskers, or users needing high performance. This guide aims to make laptop selection easier, offering smart, data-backed suggestions for a variety of users and budgets.

2. Data Collection Process

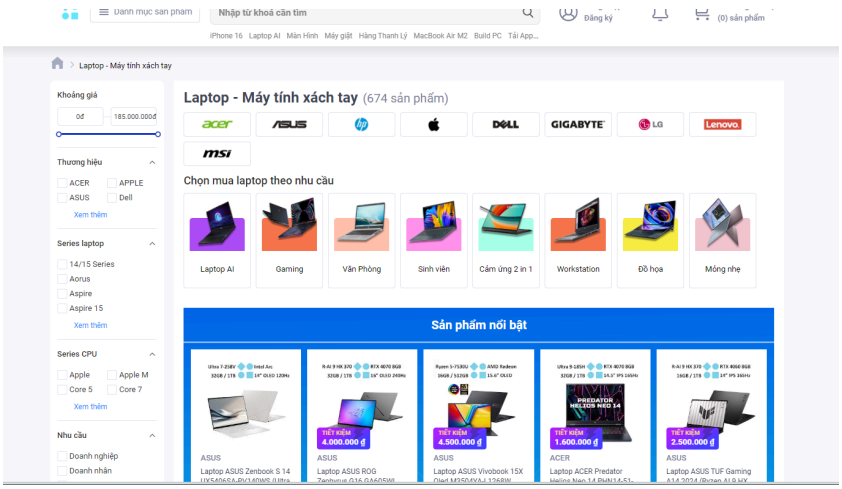
To create a dataset of laptops with various specifications and price ranges, data was collected from two popular Vietnamese e-commerce websites: **CellPhoneS** and **PhongVu Laptop**. The data collection process involved web scraping with Python libraries, ensuring compliance with each website's terms of service.

2.1 Data Sources

- **CellPhoneS Website**: This website provided a wide range of laptop models, specifications, and prices. Data was gathered using a combination of **Scrapy** (for efficiently navigating through multiple pages and links) and **Selenium** (for handling dynamic content that required interaction with the webpage).



- **PhongVu Laptop Website**: PhongVu offered another set of laptop data with detailed specifications that complemented data from CellPhoneS. Here, data was collected using **BeautifulSoup** (for straightforward HTML parsing and extraction) alongside **Selenium** to manage dynamically loaded content.



2.2 Data Fields Collected

For each laptop, the following data fields were mainly gathered from both websites:

- **Model** (e.g., Inspiron 15, MacBook Air)
- **CPU Type** (e.g., Intel i5, AMD Ryzen 5)
- **RAM Size** (e.g., 8GB, 16GB)
- **Storage Type and Size** (e.g., 256GB SSD, 512GB SSD)
- **Screen Size** (in inches)
- **Price** (in Vietnamese Dong)

Other fields like: brand, CPU cores, CPU frequency range, cache size, RAM slots, RAM speed etc were also attached to each

3. Data Preprocessing and Feature Engineering

The raw data collected from the CellPhoneS and PhongVu Laptop websites required several preprocessing steps to ensure consistency and usability in clustering. Key data fields like **Price**, **CPU**, **GPU**, **Storage**, and **RAM Size** were standardized and transformed into numerical or categorical formats, allowing them to be used effectively in clustering analysis.

We inspected the dataset for missing values and dropped columns with over 90% missing data, as they were unlikely to provide meaningful insights. Additionally, we identified and merged duplicate columns with similar meanings (e.g., “CPU” and “CPU Information”), filling in missing values from one column to another to ensure data completeness. For categorical columns containing multiple categories in a single entry, we processed these values to extract individual categories, enabling appropriate encoding for modeling.

3.1 Processing Key Features

- **Price**: Price data was initially collected as text, often including currency symbols and punctuation. This field was cleaned by removing unnecessary characters, converting the values to integers, and standardizing them to a single currency (Vietnamese Dong).
- **CPU (Processor)**: CPUs were initially represented in various formats, such as "Intel i5" or "AMD Ryzen 5". Processor names were standardized into categories by series (e.g., Intel i3 1350H, etc) to allow for better grouping.
- **GPU (Graphics Processor)**: GPU data was also categorized based on each type: (eg Nvidia GeForce RTX 3050, etc).
- **Storage**: Storage data includes the size of memory of the hardware in GB (eg 512, 256, 1024, etc) .
- **RAM Size**: RAM was represented in various formats, such as "8GB" or "16 GB". This was cleaned and converted to a numeric format (in gigabytes), allowing RAM size to be treated as a continuous variable in clustering.

Feature Extraction:

Using regular expressions and string manipulation, we extracted valuable information from complex strings. For example, from the CPU specification “Intel® Core™ Ultra 5-125H (3.6 GHz - 4.5 GHz / 18MB / 14 nhân),” we derived features such as:

- CPU Model Name: “Intel Core Ultra 5 125H”

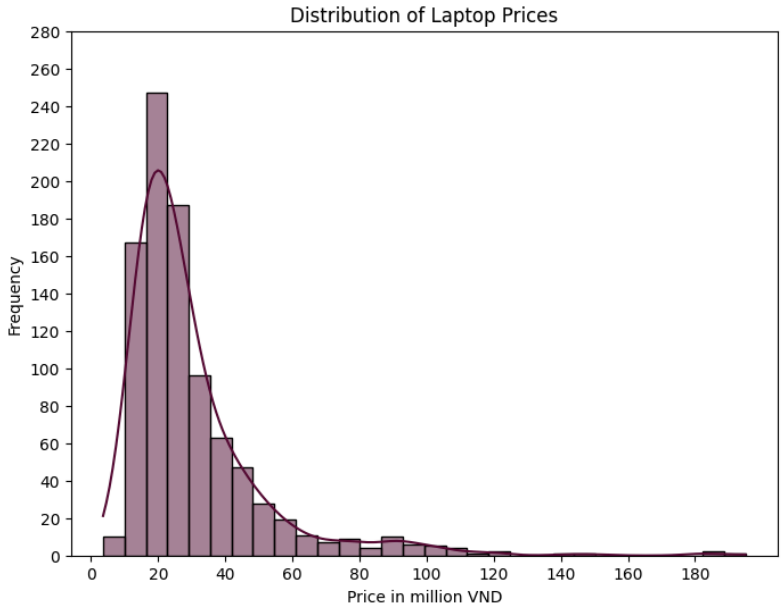
3.2 Loading Data into MongoDB Atlas Cluster

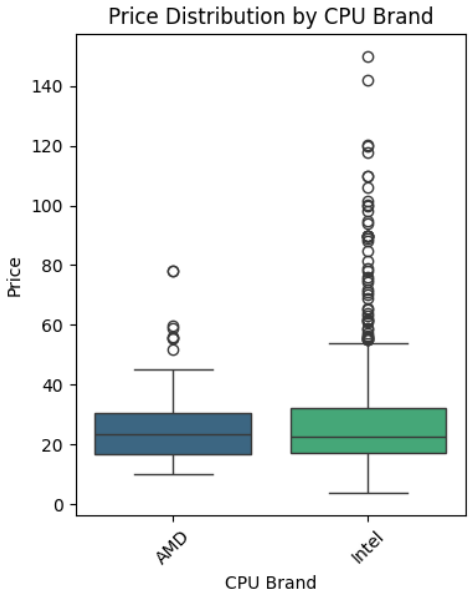
Once preprocessed, the data was loaded into a **MongoDB Atlas cluster**, a cloud-based database solution. MongoDB was chosen for its flexibility and ease of handling semi-structured data like specifications, which often vary in format and details across different laptops.


```
_id: ObjectId('6728ec53a40e0f5a48004fd3')
Tên máy : " Laptop HP Gaming Victus 16-E0170AX 4R0U7PA "
Giá : 22990000
Link : "https://cellphones.com.vn/laptop-hp-gaming-victus-16-e0170ax-4r0u7pa.h…"
Card đồ họa : "NVIDIA GeForce RTX 3050"
CPU : "AMD Ryzen 7 5800H"
Dung lượng RAM (GB) : 8
Loại RAM : "DDR4 3200 MHz"
Số khe ram : "2 khe (hỗ trợ nâng cấp 32GB)"
Dung lượng Ổ cứng (GB) : 512
Chất liệu : "Vỏ nhựa"
Màn hình cảm ứng : "Không"
Chất liệu tấm nền : "Tấm nền IPS"
Kích thước màn hình (inches) : 16.1
▸ Công nghệ màn hình : Array (3)
Độ phân giải màn hình : "1920 x 1080 pixels (FullHD)"
Webcam : "HD webcam"
Khe đọc thẻ nhớ : "Có"
Hệ điều hành : "Windows 11 Home SL"
Fi : "Wi-Fi 6 (802.11ax)"
Công nghệ âm thanh : "Bang & Olufsen audio"
▸ Pin : Array (2)
Kích thước : "370 x 260 x 23.5mm"
Trọng lượng : "2.46 kg"
▸ Cổng giao tiếp : Array (6)
▾ Show 2 more fields
```

By storing the data in MongoDB, this project benefits from both the structure required for efficient processing and the flexibility to handle complex, varied laptop specifications.

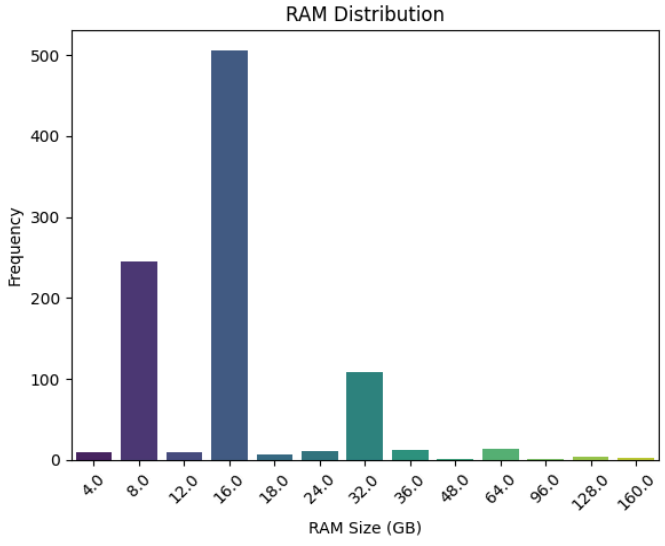
4. Exploratory Data Analysis

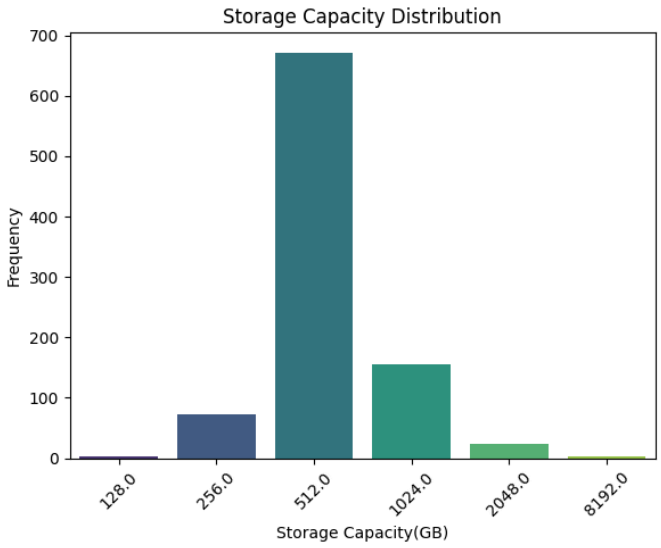


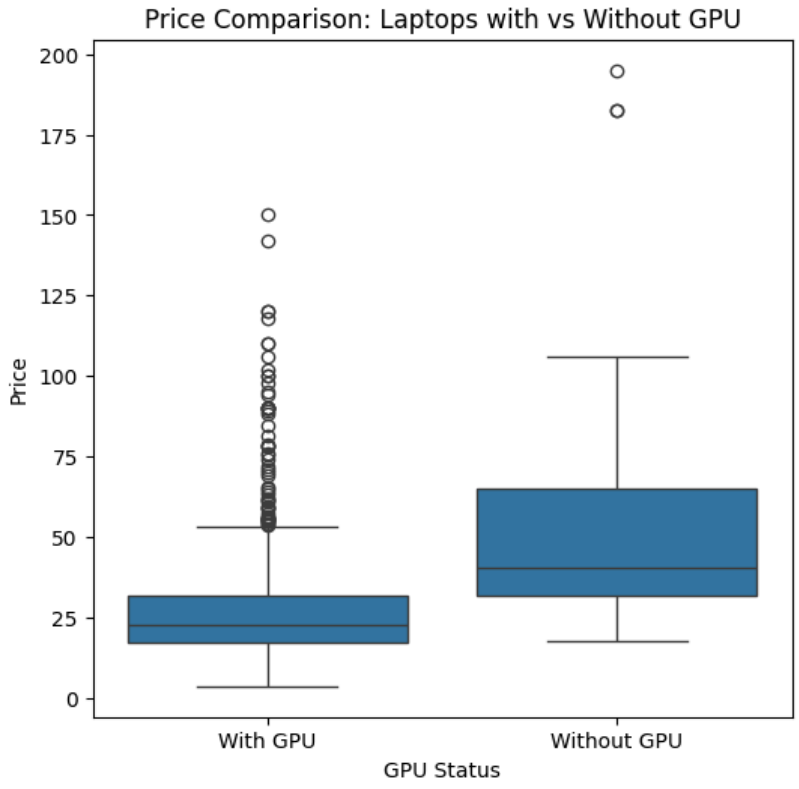


The distribution graphs indicate that the majority of laptops are priced between 15 and 25 million VND. Notably, laptops equipped with Intel CPUs tend to have a slightly higher average price compared to the AMD ones.

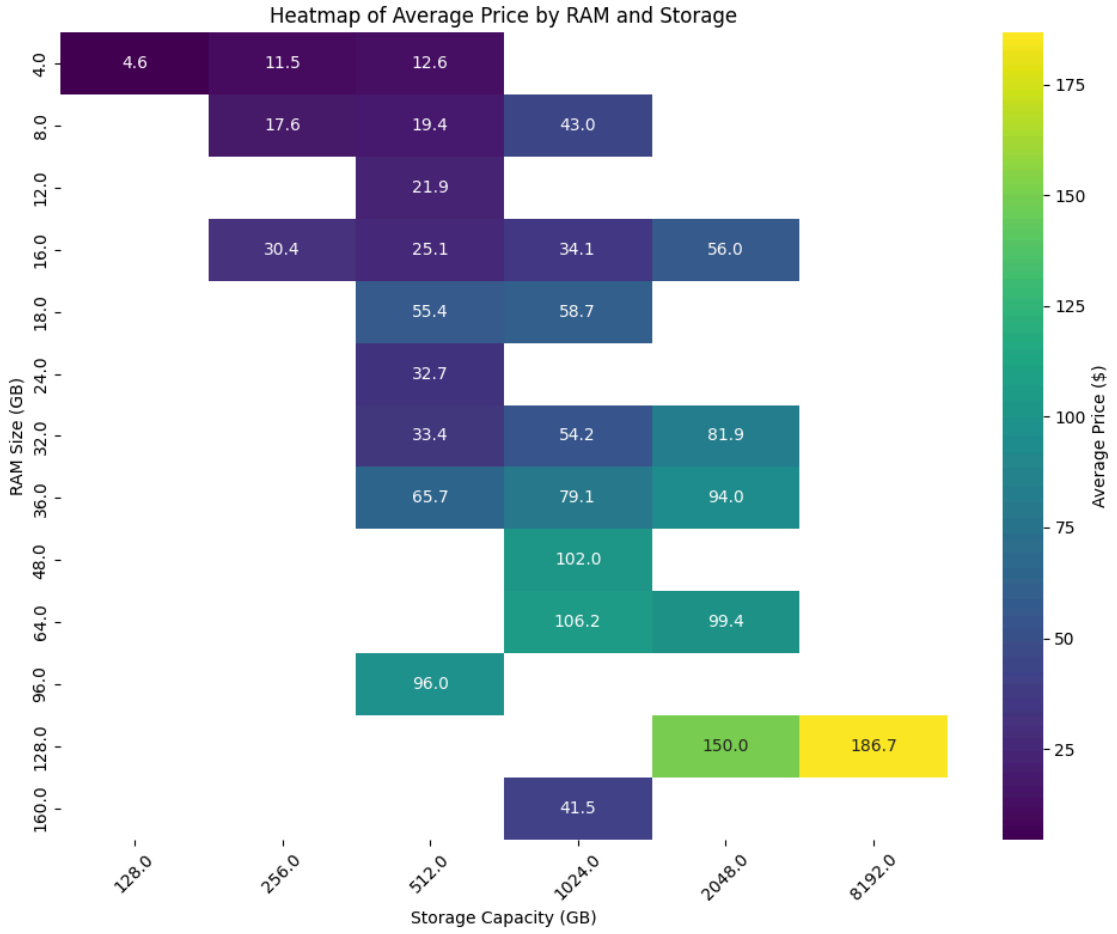
In terms of RAM and storage distributions, the majority of laptops feature a RAM size of 16 GB, followed by 8 GB and 32 GB. For storage, most laptops are equipped with 512 GB, indicating a preference for this capacity among consumers.







This box plot shows that laptops with GPUs generally have a wider and higher price range, including outliers reaching close to 200 million VND. Laptops without GPUs are more affordable, with prices generally below 50 million VND.



Higher Prices for Larger RAM and Storage Combinations: Generally, laptops with higher RAM and storage capacities have higher average prices. For example, laptops with 128 GB RAM and 2 TB (2048 GB) storage or 8 TB (8192 GB) storage are among the most expensive options. This general trend helps to establish a baseline for mid-range laptop prices in our dataset.

Common Configurations and Moderate Prices: The most cost-effective range is 8–16 GB RAM with 512 GB storage, balance between affordability and functionality. These models offer sufficient performance without a high price tag, making them accessible to a broad range of users.

5. Background knowledge: K Nearest Neighbor(KNN)

5.1. Definition

The K-Nearest Neighbors (KNN) algorithm is a supervised machine learning method employed to tackle classification and regression problems.

Advantages: Simple, interpretable, non-parametric, and effective for smaller datasets.

Limitations: Computationally expensive on large datasets and sensitive to irrelevant or scaled features.

5.2. How KNN Works

The algorithm compares the input features (e.g.,CPU, GPU, price range, etc.) with other laptops in the dataset. By identifying the closest matches (neighbors), KNN recommends laptops most similar to the user's specifications.

To find the neighbors, the KNN algorithm uses a distance metric, such as the Euclidean or Manhattan metric, to calculate the distance between the data point to be classified and all other data points in the dataset.

6. Generating Laptop Recommendation

In the recommendation system, each specific laptop can be represented by its features (CPU, GPU, RAM, storage, screen size, etc.). KNN helps identify similar laptops by finding "neighbors" in terms of these features, making it a natural choice for recommendations.

6.1. Import necessary libraries and initialize the dataset

We used *pandas* and *numpy* for efficient data manipulation and numerical operations. **LabelEncoder** and **StandardScaler** from *sklearn.preprocessing* helped in encoding categorical

variables and standardizing features, respectively, to improve distance measurements in KNN. The NearestNeighbors class from *sklearn.neighbors* implemented the KNN algorithm.

	Laptop_name	Price(VND)	GPU	CPU	RAM(GB)	Storage(GB)	Screen_size(inches)
0	Laptop HP Gaming Victus 16-E0170AX 4R0U7PA	22990000.0	NVIDIA GeForce RTX 3050	AMD Ryzen 7 5800H	8.0	512.0	16.1
1	Laptop Lenovo Yoga 7 2-in-1 14IML9 83DJ001FVN	28990000.0	INTEL Arc Graphics	Intel Core Ultra 7 155H	16.0	512.0	14.0
2	Laptop Lenovo Yoga Pro 7 14ASP9 83HN0022VN	40990000.0	AMD Radeon 880M Graphics	AMD Ryzen AI 9 365	32.0	1024.0	14.5
3	Laptop Lenovo Yoga Pro 7 14IMH9 83E2005DVN	43490000.0	NVIDIA GeForce RTX 4050	Intel Core Ultra 7 155H	32.0	1024.0	14.5
4	Laptop Lenovo LOQ 15IRX9 83DV00UGVN	30490000.0	NVIDIA GeForce RTX 4050	Intel Core i7 13650HX	24.0	512.0	15.6
5	Laptop Lenovo Ideapad 5 14ABA7 82SE007EVN	17490000.0	AMD Radeon Graphics	AMD Ryzen 7 5825U	16.0	512.0	14.0
6	Laptop ASUS Gaming ROG Strix SCAR 16 G634JZR-...	105990000.0	NVIDIA GeForce RTX 4080	Intel Raptor Lake i9 14900HX	64.0	2048.0	16.0
7	Laptop ASUS ExpertBook B1 B1402CVA-NK0104W	10690000.0	INTEL UHD Graphics	Intel Core i3 1315U	8.0	256.0	14.0
8	Laptop ASUS ExpertBook B1 B1502CVA-NJ0149W	10990000.0	INTEL UHD Graphics	Intel Core i3 1315U	8.0	512.0	15.6
9	Laptop Asus Zenbook 14 UX3402ZA-KM219W	23690000.0	INTEL Iris Xe Graphics	Intel Core i5 1240P	16.0	512.0	14.0

Figure 6.1.1. A sample of the processed dataset

6.2. Label Encoding and Handling Missing Value

In our data preprocessing steps, we applied label encoding to convert categorical variables into numerical format, which is essential for machine learning algorithms that require numerical input. This transformation allows the model to interpret and utilize categorical data effectively.

For handling missing values, we removed rows containing non-value entries, ensuring a complete dataset for analysis. This approach helps maintain data integrity and improves the reliability of our results.

	Laptop_name	Price(VND)	GPU	CPU	RAM(GB)	Storage(GB)	Screen_size(inches)
0	Laptop HP Gaming Victus 16-E0170AX 4R0U7PA	22990000.0	41	17	8.0	512.0	16.1
1	Laptop Lenovo Yoga 7 2-in-1 14IML9 83DJ001FVN	28990000.0	17	60	16.0	512.0	14.0
2	Laptop Lenovo Yoga Pro 7 14ASP9 83HN0022VN	40990000.0	9	33	32.0	1024.0	14.5
3	Laptop Lenovo Yoga Pro 7 14IMH9 83E2005DVN	43490000.0	47	60	32.0	1024.0	14.5
4	Laptop Lenovo LOQ 15IRX9 83DV00UGVN	30490000.0	47	139	24.0	512.0	15.6
5	Laptop Lenovo Ideapad 5 14ABA7 82SE007EVN	17490000.0	12	19	16.0	512.0	14.0
6	Laptop ASUS Gaming ROG Strix SCAR 16 G634JZR-...	105990000.0	50	177	64.0	2048.0	16.0
7	Laptop ASUS ExpertBook B1 B1402CVA-NK0104W	10690000.0	24	75	8.0	256.0	14.0
8	Laptop ASUS ExpertBook B1 B1502CVA-NJ0149W	10990000.0	24	75	8.0	512.0	15.6
9	Laptop Asus Zenbook 14 UX3402ZA-KM219W	23690000.0	22	95	16.0	512.0	14.0

Figure 6.2.1. A sample of dataset after applying Label Encoding

6.3. Data Scaling and KNN Model Initialization

To ensure that each feature contributes equally to the distance calculations in our KNN model, we applied feature scaling. This involved standardizing the dataset using **StandardScaler**, which transforms the features to have a mean of zero and a standard deviation of one.

For the KNN model initialization, we set the number of neighbors, $k = 7$, allowing for adjustable value based on model performance. We then instantiated the KNN model using the NearestNeighbors class with the specified k and the Euclidean distance metric, which is appropriate for our data's characteristics.

6.4. User input and Recommendation Functions

The function for input is called `get_user_preferences`, used to prompt users to input their preferences for various laptop features, including CPU model, GPU model, RAM, storage, screen size, and price range.

- **CPU and GPU Input**: It requests the user to enter the CPU and GPU model names. If the entered model is not recognized, the function suggests the closest match using the `get_close_matches` function. If no close matches are found, it prompts the user to try again.
- **Other Preferences**: The function then collects additional preferences, such as RAM, storage, screen size, and the minimum and maximum price range.
- **Return Values**: Finally, the function returns a list containing the encoded CPU and GPU values, RAM, storage, screen size, and the average price, along with the minimum and maximum price values.

The `recommend_laptops` function generates laptop recommendations based on user input.

- **Input Transformation**: It begins by transforming the user input using the same scaler applied to the training data, ensuring consistency in feature scaling.
- **Finding Neighbors**: The function then uses the KNN model to find the nearest neighbors by calculating the distances and indices of the closest matching laptops in the dataset.
- **Retrieving Recommendations**: It retrieves the recommended laptops from the dataset using the indices of the nearest neighbors.
- **Price Filtering**: The function filters the recommendations based on the user's specified price range, ensuring that only laptops within the defined budget are returned.

  The output is a DataFrame containing the recommended laptops that best match the user's preferences, facilitating informed decision-making for potential buyers.

7. Results

7.1. Recommendation system's output

The output consists of a list of recommended laptops, including a 'Distance' column that indicates how closely each laptop matches the user's specifications, helping users identify the best options available.

For example, the user's input is:

- **CPU:** AMD Ryzen 7 6800H
- **GPU:** NVIDIA GeForce RTX 3050
- **RAM:** 32 (GB)
- **Storage:** 1024 (GB)
- **Screen Size:** 16 (inches)
- **min_price:** 6000000 (VND)
- **max_price:** 100000000 (VND)

The system's output is:

Laptop_name	CPU
laptop gaming hp omen 16-n0085ax	amd ryzen 9 6900hx
laptop ai hp omen 16-xf0071ax - 8w946pa	amd ryzen 7 7840hs
laptop gaming hp omen 16-xf0070ax - 8w945pa	amd ryzen 9 7940hs
macbook pro 16 inch m1 max 10 cpu - 32 gpu 32g...	apple m1 max
laptop ai hp victus 16-s0138ax - 9q985pa	amd ryzen 7 7840hs
laptop gaming hp victus 16-s0142ax - 9q989pa	amd ryzen 5 7640hs
laptop asus rog zephyrus g16 ga605wi 2024	amd ryzen ai 9 hx 370

Laptop_name	GPU
laptop gaming hp omen 16-n0085ax	nvidia geforce rtx 3070ti
laptop ai hp omen 16-xf0071ax - 8w946pa	nvidia geforce rtx 4060
laptop gaming hp omen 16-xf0070ax - 8w945pa	nvidia geforce rtx 4070
macbook pro 16 inch m1 max 10 cpu - 32 gpu 32g...	NaN
laptop ai hp victus 16-s0138ax - 9q985pa	nvidia geforce rtx 4070
laptop gaming hp victus 16-s0142ax - 9q989pa	nvidia geforce rtx 4060
laptop asus rog zephyrus g16 ga605wi 2024	nvidia geforce rtx 4070

Laptop_name	RAM(GB)	Storage(GB)
laptop gaming hp omen 16-n0085ax	32.0	1024.0
laptop ai hp omen 16-xf0071ax - 8w946pa	32.0	1024.0
laptop gaming hp omen 16-xf0070ax - 8w945pa	32.0	1024.0
macbook pro 16 inch m1 max 10 cpu - 32 gpu 32g...	32.0	1024.0
laptop ai hp victus 16-s0138ax - 9q985pa	32.0	512.0
laptop gaming hp victus 16-s0142ax - 9q989pa	32.0	512.0
laptop asus rog zephyrus g16 ga605wi 2024	32.0	1024.0

Laptop_name	Screen_size(inches)
laptop gaming hp omen 16-n0085ax	16.1
laptop ai hp omen 16-xf0071ax - 8w946pa	16.1
laptop gaming hp omen 16-xf0070ax - 8w945pa	16.1
macbook pro 16 inch m1 max 10 cpu - 32 gpu 32g...	16.2
laptop ai hp victus 16-s0138ax - 9q985pa	16.1
laptop gaming hp victus 16-s0142ax - 9q989pa	16.1
laptop asus rog zephyrus g16 ga605wi 2024	16.0

Laptop_name	Price(VND)	Distance
laptop gaming hp omen 16-n0085ax	58990000.0	0.545927
laptop ai hp omen 16-xf0071ax - 8w946pa	55790000.0	0.619778
laptop gaming hp omen 16-xf0070ax - 8w945pa	59590000.0	0.771004
macbook pro 16 inch m1 max 10 cpu - 32 gpu 32g...	53990000.0	1.248424
laptop ai hp victus 16-s0138ax - 9q985pa	39090000.0	1.353052
laptop gaming hp victus 16-s0142ax - 9q989pa	36590000.0	1.378606
laptop asus rog zephyrus g16 ga605wi 2024	77990000.0	1.385222

7.2. Evaluation

7.2.1. Precision at K

To assess the performance of the recommendation system, we use the precision at K metric. This metric evaluates the accuracy of the top K recommended laptops by measuring the proportion of relevant recommendations among the K most relevant items suggested to the user. A higher precision at K indicates that the system effectively delivers relevant recommendations, enhancing user satisfaction and trust in the recommendations provided.

The formula for Precision at K is given by:

$$\text{Precision@K} = \frac{\text{Number of Relevant Items in Top K}}{K}$$

In this case (depends on the user input), the precision at K is equal to 1, which is all K items recommended are relevant.

This is the best possible score, indicating a perfect recommendation list for the top K items.

7.2.2. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is a commonly used metric to evaluate the accuracy of a model's predictions compared to the actual values. In the context of a laptop recommendation system, the RMSE metric is to assess how well the model's recommended laptops match the actual laptops's specifications.

The formula for calculating RMSE is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

```
RMSE for Distances: 1.1009940938051528
```

8. Conclusion

This project applied K-Nearest Neighbors (KNN) to create a recommendation system that suggests laptops based on specifications and price, tailored to user preferences. Data was collected from CellPhoneS and PhongVu Laptop, preprocessed, and stored in MongoDB Atlas for efficient retrieval. KNN proved effective in generating targeted recommendations by identifying similar laptops based on user requirements. This approach demonstrates the potential of data science to simplify complex purchasing decisions. Future work could enhance the system by adding features like battery life or expanding the dataset for even more precise recommendations.