A

Project Report on

Laptop Price Predictor

(Core Module: 5)

Submitted in partial fulfillment of completion of the course

Advanced Diploma in IT, Networking and Cloud Computing

Submitted by:

NISHAT PERWEZ

RIYA SHARMA

SK RAJESH

Under Guidance of:

Arpita Roy (Edunet)

Sayanti Manna (Edunet)

Year 2023

PROJECT OVERVIEW

ABSTRACT

In the rapidly evolving landscape of consumer electronics, accurately predicting the price of laptops has become a critical task for both consumers and manufacturers. This project aims to develop a robust Laptop Price Predictor using advanced data analysis techniques. Leveraging a diverse dataset encompassing various laptop specifications, brand information, and market trends, our analysis delves into the intricate relationships between these factors and the final retail price.

The methodology involves data pre processing, feature engineering, and the application of machine learning algorithms to construct a predictive model. Exploratory Data Analysis (EDA) techniques are employed to uncover hidden patterns and insights within the dataset. Feature importance analysis helps identify key factors influencing laptop prices, allowing for informed decision-making in product development and pricing strategies.

The project contributes to the field of data analysis by offering a comprehensive approach to predicting laptop prices, enhancing transparency in a market characterized by rapid technological advancements. The developed model not only serves as a valuable tool for consumers to make informed purchasing decisions but also provides manufacturers with insights to optimize product pricing strategies and maintain competitiveness in the dynamic consumer electronics industry. The findings from this project can be utilized to streamline marketing efforts, enhance product positioning, and ultimately improve the overall efficiency of the laptop supply chain.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have supported and contributed to the successful completion of this project. Your assistance and encouragement have been invaluable throughout this journey.

First and foremost, we want to thank my project supervisors, Arpita Roy(Edunet) and Sayanti Manna(Edunet), for their guidance, expertise, and unwavering support. Their insights and feedback have been instrumental in shaping the direction of this project.

We would also like to extend my appreciation to my fellow classmates who provided valuable input, shared resources, and engaged in stimulating discussions that enriched the project. Your collaborative spirit was a driving force behind our achievements.

Furthermore, we want to acknowledge our friends and family for their patience, understanding, and encouragement throughout this endeavour. Your support provided the motivation we needed to see this project through to its completion.

Last but not least, we are grateful to the entire faculty and staff of NSTI Howrah for providing a conducive learning environment and the necessary resources to undertake this project.

This project has been a rewarding learning experience, and we are thankful for the collective efforts of everyone involved. Your support has been instrumental in making this project a reality.

Thank you all for being a part of this journey.

Riya Sharma

Nishat Perwez

SK Rajesh

NSTI Howrah

18/11/2023

TEAM COMPOSITION AND WORKLOAD DIVISION

The team composition for the " Laptop Price Predictor" project can vary depending on the project's complexity, goals, and scope. This is the team composition:

TEAM COMPOSITION

1. Nishat Perwez(Leader):

- Responsible for overseeing the overall project coordination, planning, and communication Ensures that the project stays on schedule and meets objectives, Coordinates with other team members to gather requirements and documentation.

2. Riya Sharma(Data Analysts):

- Responsible for data collection, cleaning, and preprocessing.
- Use Python libraries such as Pandas, NumPy for data manipulation and analysis.
- Creates visualizations using libraries like Matplotlib, Seaborn, or Plotly.
- Transforms data insights into visually appealing and informative charts and graphs.

3. SK Rajesh(Testers):

- Tests the code for bugs and ensures that the analysis results are accurate.
- Validates the consistency of the data and the reliability of the code.

WORKLOAD DIVISION

Detail how the workload is divided among the team members highlighting their responsibilities and tasks:

1. Nishat Perwez: Project Planning and Conceptualization
2. Riya Sharma: Data Analyzing and data visualization
3. SK Rajesh :Testing and quality assurance

Documentation and Report Writing:

Lead the documentation process, with contributions from all team members for their respective areas of expertise.

INTRODUCTION TO PROBLEM

In the dynamic and ever-evolving realm of consumer electronics, laptops stand out as indispensable tools that bridge the gap between technology and daily life. As consumers navigate through an extensive array of laptop options, one significant challenge they face is the uncertainty surrounding laptop prices. The intricate interplay of numerous features, specifications, and market dynamics makes it difficult for both consumers and manufacturers to accurately gauge the fair value of a laptop.

Certainly! Here are some key points to consider for the introduction to the problem in the Laptop Price Predictor project:

Feature Overload: Laptops come with a multitude of features, ranging from processing power and storage capacity to display quality and brand reputation. Understanding how these features contribute to the final price is crucial for consumers to make well-informed choices.

Market Dynamics: The consumer electronics market is characterized by rapid technological advancements and shifting market trends. These dynamics make it challenging for manufacturers to set competitive prices that align with consumer expectations and market demands.

Data-Driven Solution: The Laptop Price Predictor project aims to bridge this gap by harnessing the power of data analysis. Through comprehensive data exploration and the application of machine learning algorithms, the project seeks to unveil patterns and correlations, providing a systematic approach to predicting laptop price.

PROPOSED SOLUTION

To tackle the challenge of predicting laptop prices in a dynamic and complex market, the Laptop Price Predictor project proposes a comprehensive data-driven solution. The methodology involves a series of steps aimed at harnessing the power of data analysis and machine learning to develop an accurate and robust predictive model.

Data Collection and Preprocessing:

Gather a diverse dataset encompassing a wide range of laptops, including specifications (processor, RAM, storage, etc.), brand information, and historical pricing data.

Clean and preprocess the data to handle missing values, outliers, and ensure uniformity in format.

Exploratory Data Analysis (EDA):

Conduct in-depth exploratory data analysis to uncover patterns, correlations, and insights within the dataset.

Visualize key relationships between individual features and the target variable (price) to identify trends and outliers.

Feature Engineering:

Engineer new features or transform existing ones to enhance the predictive power of the model.

Consider feature scaling, normalization, and categorical encoding to prepare the data for machine learning algorithms.

Model Selection:

Evaluate and select appropriate machine learning algorithms for regression tasks, considering factors such as model interpretability, performance, and scalability.

Training and Validation:

Split the dataset into training and validation sets to train the model on a subset of the data and assess its performance on unseen data.

Machine Learning:

Apply ML models for predictive analysis.

REQUIREMENTS

TECHNOLOGY STACK

Programming Language:

1. Python:
   - Core language for data analysis, manipulation, and visualization.

Data Analysis and Visualization:

1) Jupyter Notebooks:
   - Interactive and collaborative environment for data analysis.

Libraries and Frameworks:

1. Pandas:
   - Data manipulation and analysis.
2. NumPy:
   - Numerical operations on data.
3. Matplotlib, Seaborn:
   - Data visualization.
4. Scikitlearn, TensorFlow:
   - Machine learning models.

HARDWARE

- CPU and RAM
- Storage: Adequate storage space for the database, application code, and uploaded files.
- Network Infrastructure: Reliable internet connectivity, firewalls, and security measures to protect the system.

SOFTWARE

- Operating System
- Python Environment
- Jupyter Notebook
- Web Server

DEPLOYMENT ENVIRONMENT

Local Development:

- Team members can initially work on their local machines.

USER REQUIREMENTS

1. **User-Friendly Interface:**

   The system should feature an intuitive and user-friendly interface accessible to both consumers and manufacturers.
   The design should facilitate easy navigation, with clear instructions on how to input laptop specifications and interpret predicted prices.

2. **Input Flexibility:**

   Users should be able to input a variety of laptop specifications, including processor type, RAM size, storage capacity, display features, and brand information.
   The system should accommodate different input formats and units to enhance flexibility for users.

3. **Real-Time Predictions:**

   The platform should provide real-time predictions, delivering instantaneous results based on the input provided by the user.
   Users should receive predicted prices promptly to support on-the-spot decision-making.

4. **Educational Resources:**

   Include educational resources or tooltips within the interface to help users understand the significance of different laptop specifications and features.
   Provide guidance on how users can use the tool effectively for their specific needs.

IMPLEMENTATION DETAILS

The implementation of the Laptop Price Predictor project involves several steps, including data loading, cleaning, analysis, and visualization.

STEP1- At Firstl, we have to download csv file from Kaggle (https://www.kaggle.com/datasets/andrewgeorgeissac/laptop-price-predictor).

STEP 2- Secondly, We have to extract file.

STEP 3- We have to open JUPYTER NOTEBOOK.

STEP 4 - Further we have to import all necessary libraries like NumPy, pandas, math plot, seaborn etc.

STEP 5 - Then, we have to read the whole csv file by this command [df = pd. read_csv (udemy_courses.csv')

print(df)]

STEP 6 - Finally We have Explore the data to get a sense of its structure and content.

#to find the rows and columns of the dataframe.

```
print(df.shape)
```

#To find what are the column index of the dataframe

```
print(df.columns)
```

#to find the datatype of the CSV file

```
print(df.dtypes)
```

# To find the information about the file like storage,datatypes,class etc.

```
df.info()
```

#to see the last 10 data from the dataframe

```
print(df.tail(n=10))
```

#it take all value present in csv file method returns a Data Frame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False

```
df.isnull().values.any()
```

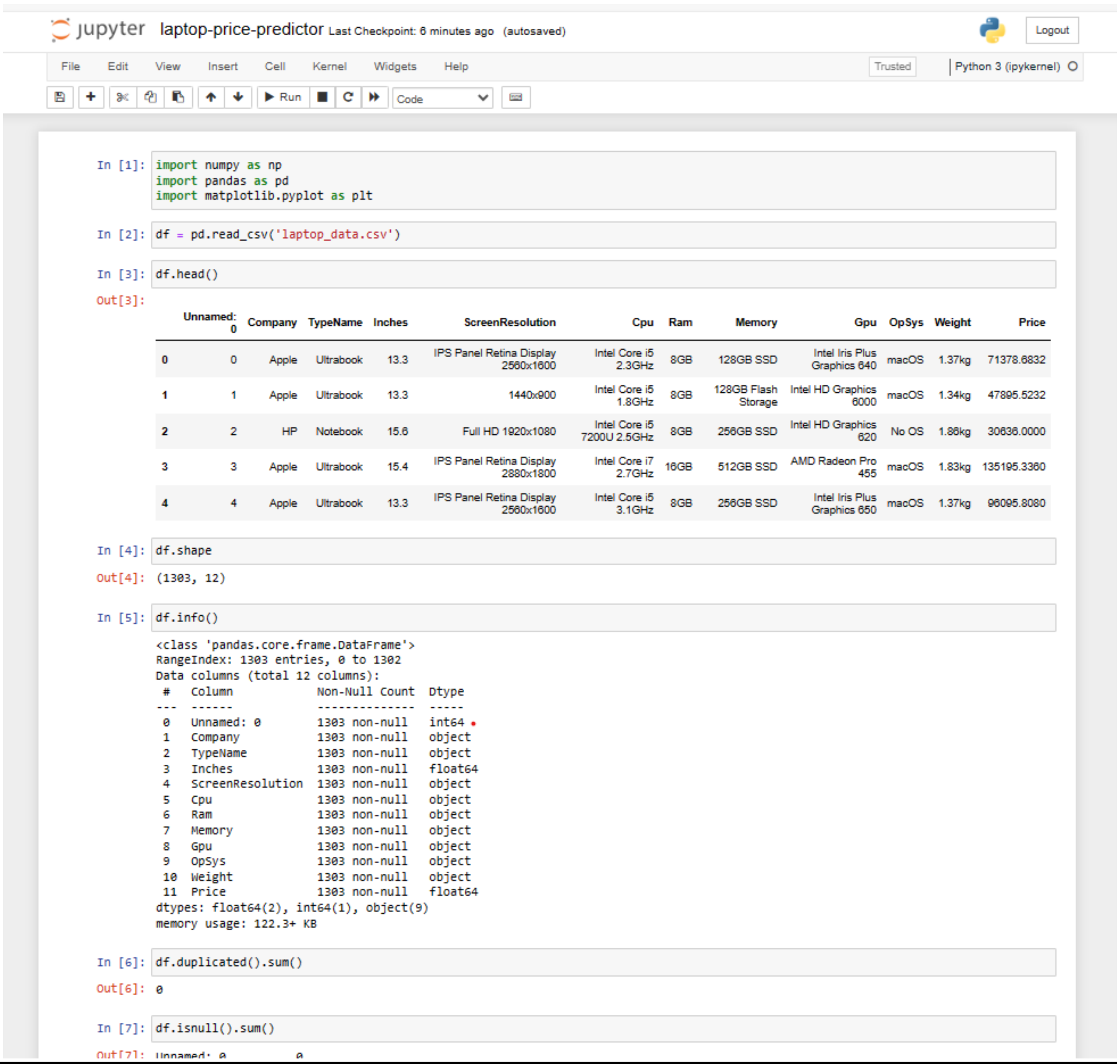
STEP 7 - We have to show different charts and graph for understanding that is in our csv file the necessary queries .

STEP 8 - Correlation matrix heat map. So, we have to check the courses relation with other columns. It shows the relation with all.

STEP 9 - We have to update and improve platform based on user because day by day the demand of courses is increasing.

STEP 10 – We have also created a dashboard so that we can check the different changes according to the subject present.

TESTING

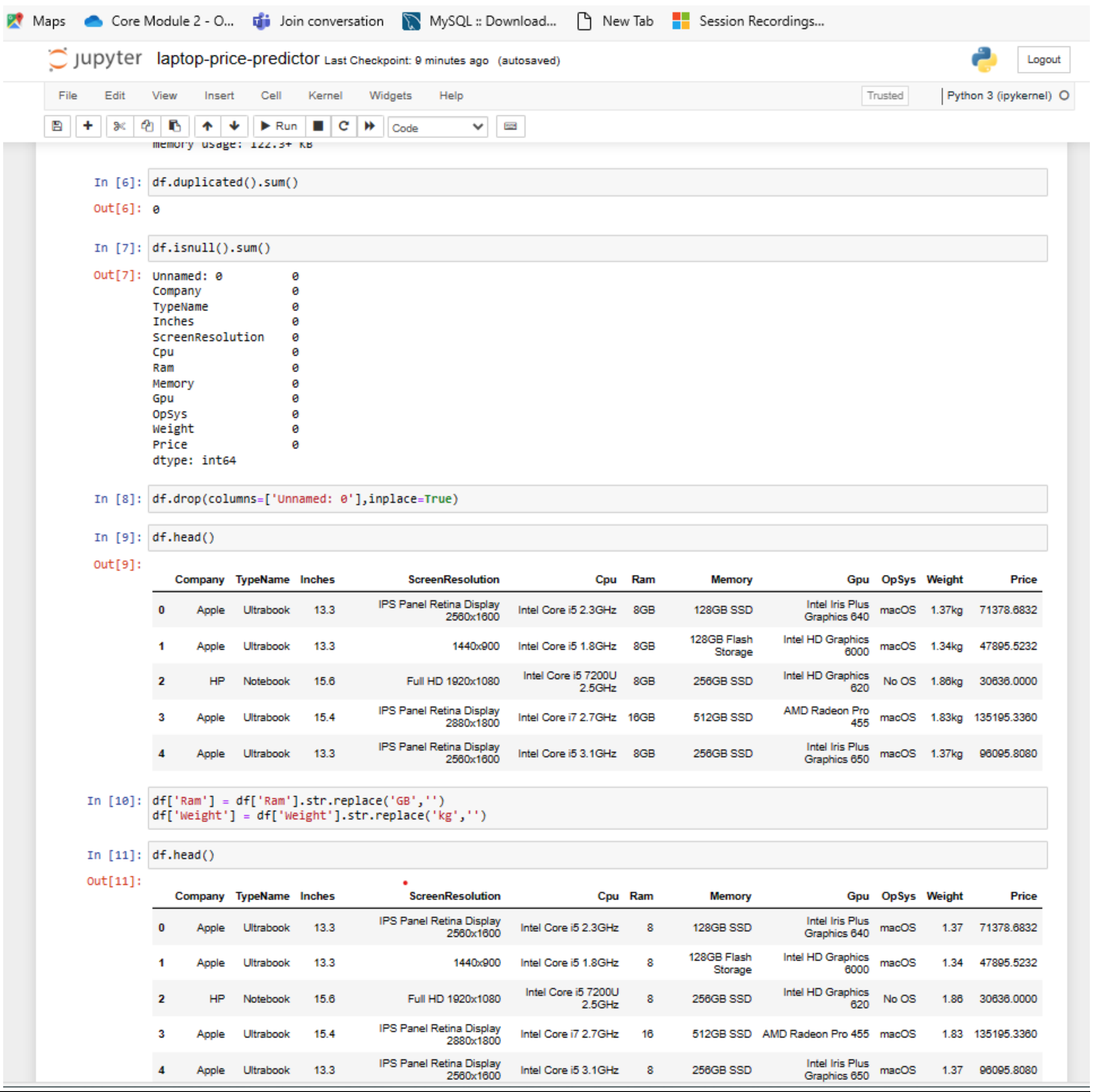


```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

In [2]: df = pd.read_csv('laptop_data.csv')

In [3]: df.head()
Out[3]:
```

	Unnamed: 0	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	71378.6832
1	1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	47895.5232
2	2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	30636.0000
3	3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	135195.3360
4	4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	96095.8080

```
In [4]: df.shape
Out[4]: (1303, 12)

In [5]: df.info()
        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 1303 entries, 0 to 1302
        Data columns (total 12 columns):
         #   Column            Non-Null Count  Dtype
        ---  ------            --------------  -----
         0   Unnamed: 0        1303 non-null   int64
         1   Company           1303 non-null   object
         2   TypeName          1303 non-null   object
         3   Inches            1303 non-null   float64
         4   ScreenResolution  1303 non-null   object
         5   Cpu               1303 non-null   object
         6   Ram               1303 non-null   object
         7   Memory            1303 non-null   object
         8   Gpu               1303 non-null   object
         9   OpSys             1303 non-null   object
         10  Weight            1303 non-null   object
         11  Price             1303 non-null   float64
        dtypes: float64(2), int64(1), object(9)
        memory usage: 122.3+ KB

In [6]: df.duplicated().sum()
Out[6]: 0

In [7]: df.isnull().sum()
Out[7]: Unnamed: 0          0
```

We have imported different types of python library and testing all the code.



```
memory usage: 122.3+ KB
```

In [6]:
```
df.duplicated().sum()
```
Out[6]:
```
0
```

In [7]:
```
df.isnull().sum()
```
Out[7]:
```
Unnamed: 0          0
Company             0
TypeName            0
Inches              0
ScreenResolution    0
Cpu                 0
Ram                 0
Memory              0
Gpu                 0
OpSys               0
Weight              0
Price               0
dtype: int64
```

In [8]:
```
df.drop(columns=['Unnamed: 0'],inplace=True)
```

In [9]:
```
df.head()
```
Out[9]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	71378.6832
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	47895.5232
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	30636.0000
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	135195.3360
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	96095.8080

In [10]:
```
df['Ram'] = df['Ram'].str.replace('GB','')
df['Weight'] = df['Weight'].str.replace('kg','')
```

In [11]:
```
df.head()
```
Out[11]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080

DEPLOYMENT

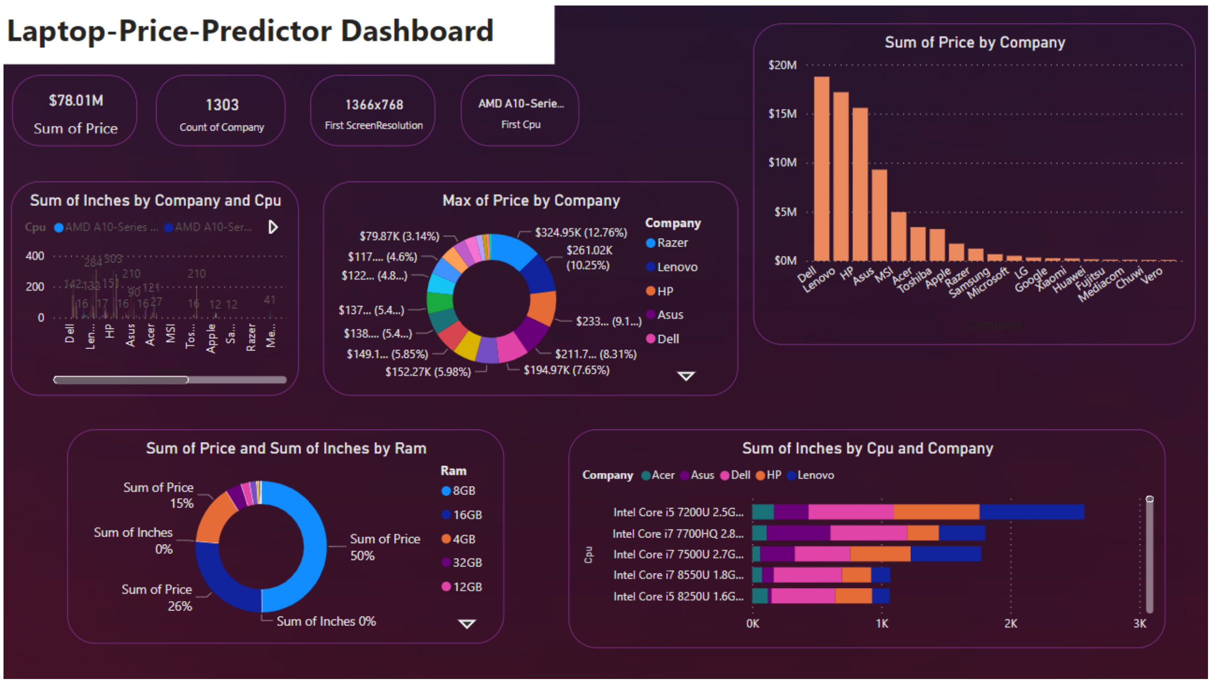


Image showing Dashboard of Laptop Price Predictor

FUTURE SCOPE

The future scope of the Laptop Price Predictor project extends beyond its initial implementation, presenting opportunities for expansion, enhancement, and integration with emerging technologies. Some potential future developments include:

Integration with E-Commerce Platforms:

Collaborate with e-commerce platforms to integrate the Laptop Price Predictor directly into product pages. This integration could provide real-time pricing predictions for users as they explore laptops online.

Incorporation of External Data Sources:

Expand the dataset by incorporating additional external data sources, such as customer reviews, market trends, and technological advancements. This would enhance the model's predictive capabilities by capturing a more comprehensive view of the laptop market.

Global Market Expansion:

Extend the scope of the project to cover a broader geographical reach, considering regional variations in pricing trends, consumer preferences, and market dynamics. This would make the tool more relevant and valuable on a global scale.

The future scope of the project is expansive and can be tailored based on emerging technologies, user feedback, and the evolving landscape of online education. Continuous improvement and adaptation to new trends will be crucial for keeping the project relevant and valuable.

Collaboration with Retailers:

Partner with retail chains to incorporate the Laptop Price Predictor into their sales processes. This could enhance the in-store or online shopping experience for customers by providing instant pricing insights.

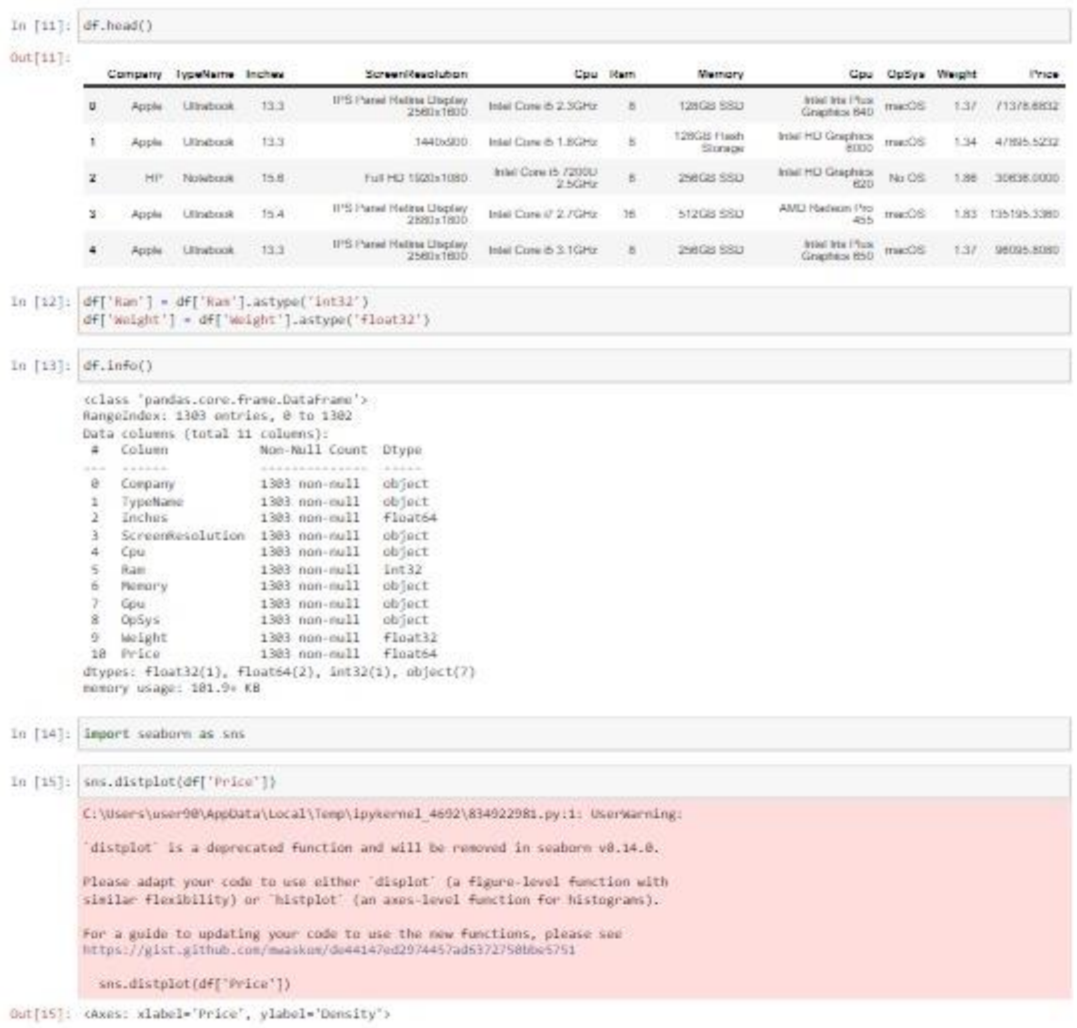
CONCLUSION

In conclusion, the Laptop Price Predictor project presents a powerful solution to address the challenges faced by consumers and manufacturers in the dynamic landscape of the consumer electronics market. By leveraging advanced data analysis techniques and machine learning algorithms, this project aims to revolutionize the way we understand, predict, and interact with laptop pricing. Key Findings:

For manufacturers, the project holds the potential to optimize pricing strategies, enhance market competitiveness, and adapt to the ever-changing demands of consumers. The insights gained from feature importance analysis can guide strategic decisions, allowing manufacturers to align their products with consumer preferences and market trends more effectively.

APPENDIX A

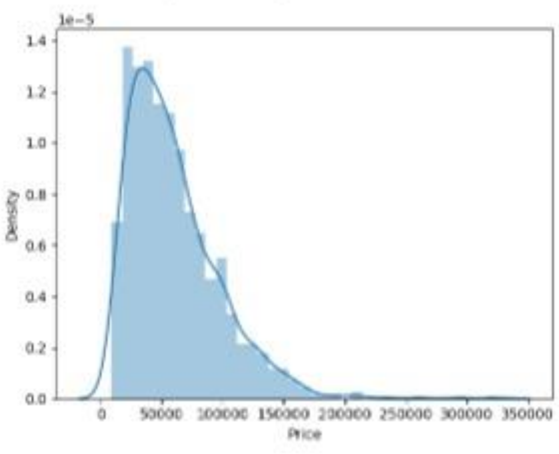
SCREENSHOT OF THE PROJECT



```
In [11]: df.head()
Out[11]:
```

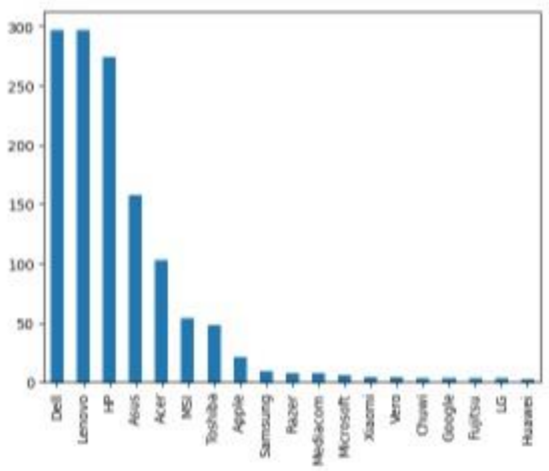
	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080

```
In [12]: df['Ram'] = df['Ram'].astype('int32')
         df['Weight'] = df['Weight'].astype('float32')

In [13]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Company           1303 non-null   object
 1   TypeName          1303 non-null   object
 2   Inches            1303 non-null   float64
 3   ScreenResolution  1303 non-null   object
 4   Cpu               1303 non-null   object
 5   Ram               1303 non-null   int32
 6   Memory            1303 non-null   object
 7   Gpu               1303 non-null   object
 8   OpSys             1303 non-null   object
 9   Weight            1303 non-null   float32
 10  Price             1303 non-null   float64
dtypes: float32(1), float64(2), int32(1), object(7)
memory usage: 101.9+ KB

In [14]: import seaborn as sns

In [15]: sns.distplot(df['Price'])

C:\Users\user90\AppData\Local\Temp\ipykernel_4692\834922981.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Price'])

Out[15]: <Axes: xlabel='Price', ylabel='Density'>
```

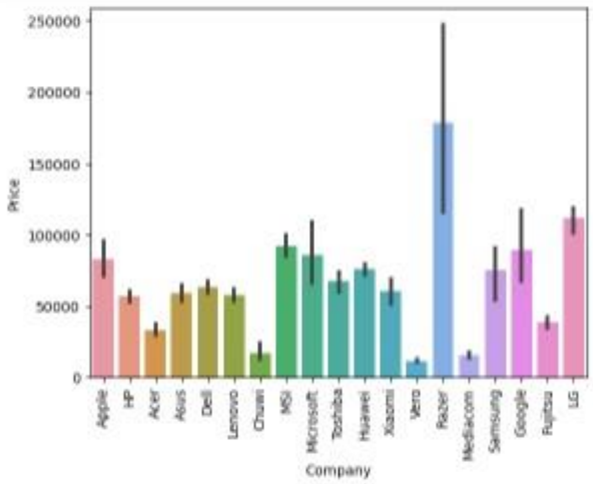


Representing data from the graph

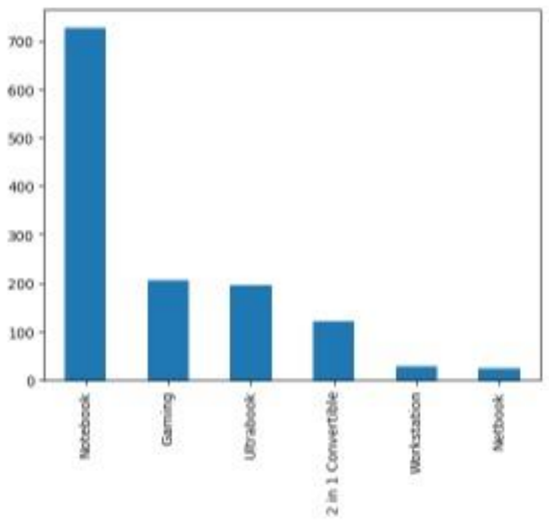
```
In [16]: df['Company'].value_counts().plot(kind='bar')

Out[16]: <Axes: >
```



```
In [17]: sns.barplot(x=df['Company'],y=df['Price'])
         plt.xticks(rotation='vertical')
         plt.show()
```

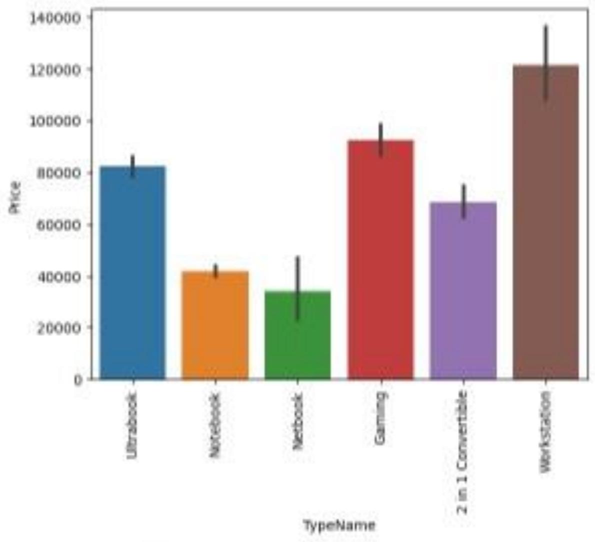


```
In [18]: df['TypeName'].value_counts().plot(kind='bar')

Out[18]: <Axes: >
```

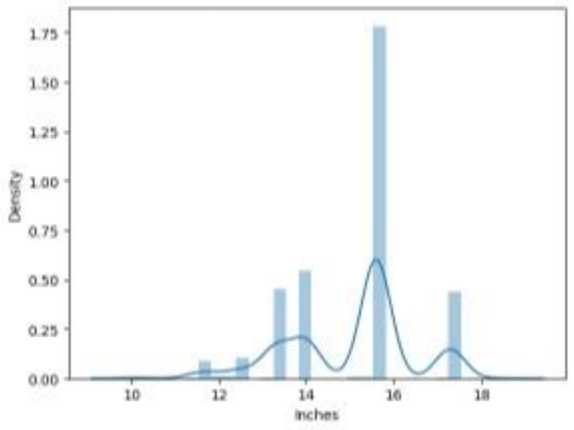


Representing data from the graph

```
In [19]: sns.barplot(x=df['TypeName'],y=df['Price'])
         plt.xticks(rotation='vertical')
         plt.show()
```



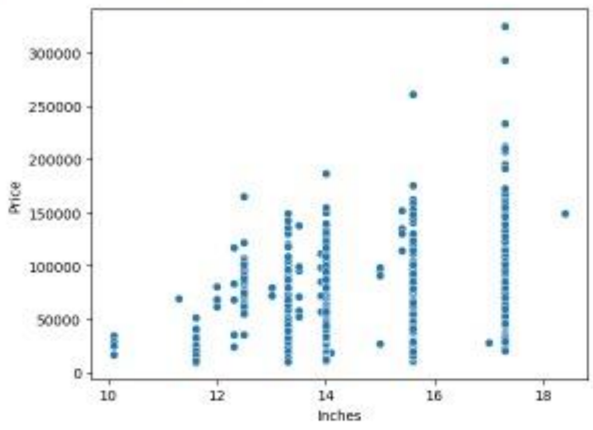
```
In [20]: sns.distplot(df['Inches'])

C:\Users\user90\AppData\Local\Temp\ipykernel_4692\1439577752.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Inches'])

Out[20]: <Axes: xlabel='Inches', ylabel='Density'>
```



```
In [21]: sns.scatterplot(x=df['Inches'],y=df['Price'])

Out[21]: <Axes: xlabel='Inches', ylabel='Price'>
```

Representing data from the graph

```
In [21]: sns.scatterplot(x=df['Inches'],y=df['Price'])

Out[21]: <Axes: xlabel='Inches', ylabel='Price'>
```



```
In [22]: df['ScreenResolution'].value_counts()

Out[22]: Full HD 1920x1080                                    507
         1366x768                                             281
         IPS Panel Full HD 1920x1080                          230
         IPS Panel Full HD / Touchscreen 1920x1080             53
         Full HD / Touchscreen 1920x1080                       47
         1600x900                                              23
         Touchscreen 1366x768                                  16
         Quad HD+ / Touchscreen 3200x1800                      15
         IPS Panel 4K Ultra HD 3840x2160                       12
         IPS Panel 4K Ultra HD / Touchscreen 3840x2160         11
         4K Ultra HD / Touchscreen 3840x2160                   10
         4K Ultra HD 3840x2160                                  7
         Touchscreen 2560x1440                                  7
         IPS Panel 1366x768                                     7
         IPS Panel Quad HD+ / Touchscreen 3200x1800             6
         IPS Panel Retina Display 2560x1600                     6
         IPS Panel Retina Display 2304x1440                     6
         Touchscreen 2256x1504                                  6
         IPS Panel Touchscreen 2560x1440                        5
         IPS Panel Retina Display 2880x1800                     4
         IPS Panel Touchscreen 1920x1200                        4
         1440x900                                               4
         IPS Panel 2560x1440                                    4
         IPS Panel Quad HD+ 2560x1440                           3
         Quad HD+ 3200x1800                                     3
         1920x1080                                              3
         Touchscreen 2400x1600                                  3
         2560x1440                                              3
         IPS Panel Touchscreen 1366x768                         3
         IPS Panel Touchscreen / 4K Ultra HD 3840x2160          2
         IPS Panel Full HD 2160x1440                            2
         IPS Panel Quad HD+ 3200x1800                           2
         IPS Panel Retina Display 2736x1824                     1
         IPS Panel Full HD 1920x1200                            1
         IPS Panel Full HD 2560x1440                            1
         IPS Panel Full HD 1366x768                             1
         Touchscreen / Full HD 1920x1080                        1
         Touchscreen / Quad HD+ 3200x1800                       1
         Touchscreen / 4K Ultra HD 3840x2160                    1
         IPS Panel Touchscreen 2400x1600                        1
         Name: ScreenResolution, dtype: int64

In [23]: df['Touchscreen'] = df['ScreenResolution'].apply(lambda x:1 if 'Touchscreen' in x else 0)

In [24]: df.sample(5)

Out[24]:
```

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen
832	Lenovo	Notebook	15.6	Full HD 1920x1080	Intel Core i5 8250U 1.6GHz	4	256GB SSD	Intel UHD Graphics 620	Windows 10	1.80	44169.1200	0
369	HP	Ultrabook	14.0	IPS Panel Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	8	256GB SSD	Intel HD Graphics 620	Windows 10	1.36	93240.0000	0
1298	Lenovo	2 in 1 Convertible	14.0	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6500U 2.5GHz	4	128GB SSD	Intel HD Graphics 520	Windows 10	1.80	33992.6400	1
665	Dell	Gaming	15.6	Full HD 1920x1080	Intel Core i7 7820HK 2.9GHz	16	256GB SSD + 1TB HDD	Nvidia GeForce GTX 1070	Windows 10	3.49	149916.6000	0
1063	Dell	Notebook	15.6	Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	16	2TB HDD	AMD Radeon R7 M445	Windows 10	2.32	52746.6672	0

Representing data from the graph

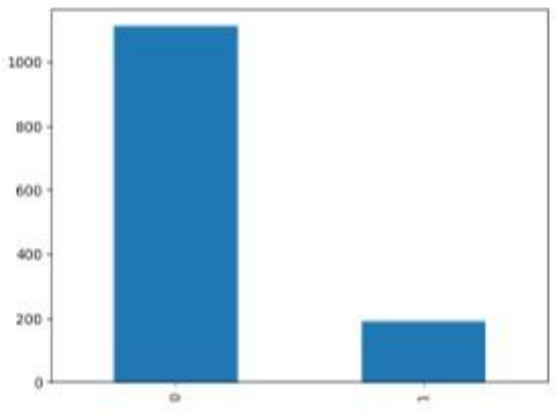
```
In [23]: df['Touchscreen'] = df['ScreenResolution'].apply(lambda x:1 if 'Touchscreen' in x else 0)
```

```
In [24]: df.sample(5)
```

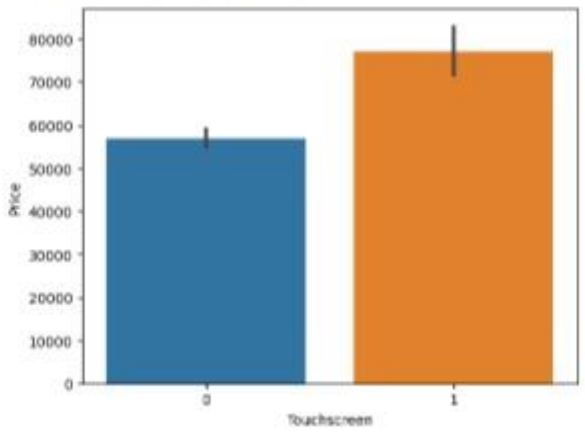
Out[24]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen
632	Lenovo	Notebook	15.6	Full HD 1920x1080	Intel Core i5 8250U 1.6GHz	4	256GB SSD	Intel UHD Graphics 620	Windows 10	1.80	44169.1200	0
389	HP	Ultrabook	14.0	IPS Panel Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	8	256GB SSD	Intel HD Graphics 620	Windows 10	1.38	93240.0000	0
1298	Lenovo	2 in 1 Convertible	14.0	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6500U 2.5GHz	4	128GB SSD	Intel HD Graphics 520	Windows 10	1.80	33992.6400	1
665	Dell	Gaming	15.6	Full HD 1920x1080	Intel Core i7 7820HK 2.9GHz	16	256GB SSD + 1TB HDD	Nvidia GeForce GTX 1070	Windows 10	3.49	149916.6000	0
1063	Dell	Notebook	15.6	Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	16	2TB HDD	AMD Radeon R7 M445	Windows 10	2.32	52746.6672	0

```
In [25]: df['Touchscreen'].value_counts().plot(kind='bar')
Out[25]: <Axes: >
```



```
In [26]: sns.barplot(x=df['Touchscreen'],y=df['Price'])
Out[26]: <Axes: xlabel='Touchscreen', ylabel='Price'>
```



```
In [27]: df['Ips'] = df['ScreenResolution'].apply(lambda x:1 if 'IPS' in x else 0)
```

```
In [28]: df.head()
```

Out[28]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen	Ips
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1

```
In [29]: df['Ips'].value_counts().plot(kind='bar')
Out[29]: <Axes: >
```

```
In [53]: df.drop(columns=['Cpu','Cpu Name'],inplace=True)
```

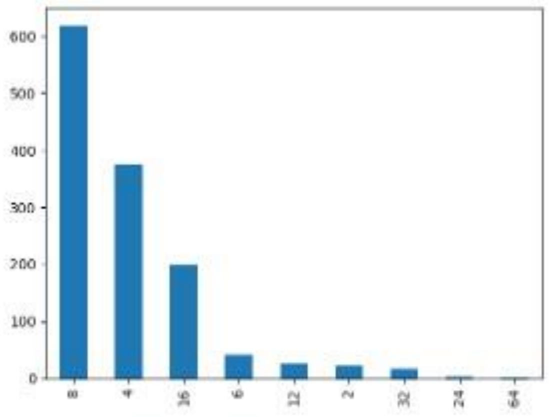
```
In [54]: df.head()
```

Out[54]:

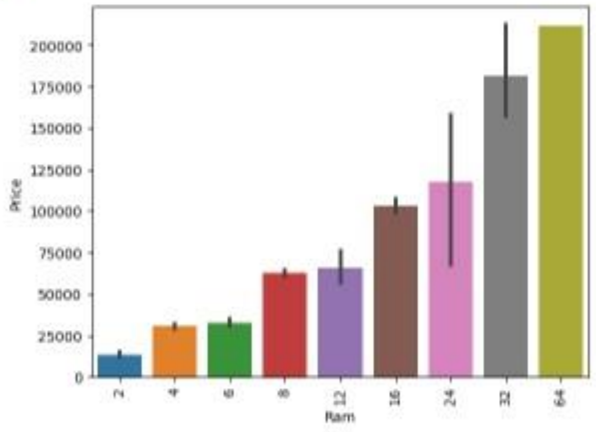
	Company	TypeName	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen	Ips	ppi	Cpu brand
0	Apple	Ultrabook	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	226.983005	Intel Core i5
1	Apple	Ultrabook	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	127.677940	Intel Core i5
2	HP	Notebook	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	141.211998	Intel Core i5
3	Apple	Ultrabook	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	220.534624	Intel Core i7
4	Apple	Ultrabook	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	226.983005	Intel Core i5

```
In [55]: df['Ram'].value_counts().plot(kind='bar')
```

```
Out[55]: <Axes: >
```



```
In [56]: sns.barplot(x=df['Ram'],y=df['Price'])
         plt.xticks(rotation='vertical')
         plt.show()
```

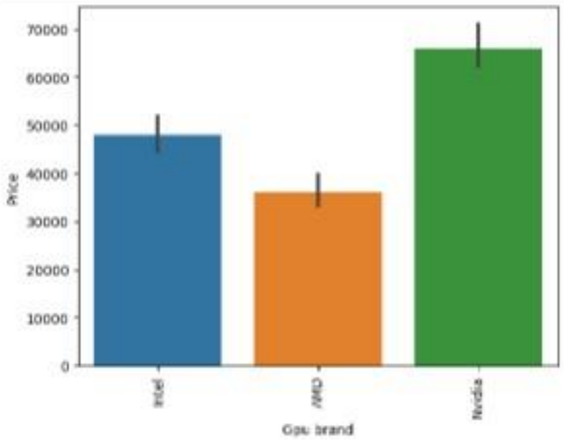


```
In [57]: df['Memory'].value_counts()
```

```
Out[57]: 256GB SSD                        412
         1TB HDD                          223
         500GB HDD                        132
         512GB SSD                        118
         128GB SSD +  1TB HDD              94
         128GB SSD                         76
         256GB SSD +  1TB HDD              73
         32GB Flash Storage                38
         2TB HDD                           16
         64GB Flash Storage                15
         512GB SSD +  1TB HDD              14
         1TB SSD                           14
         256GB SSD +  2TB HDD              10
         1.0TB Hybrid                       9
         256GB Flash Storage                8
         16GB Flash Storage                 7
         32GB SSD                           6
         180GB SSD                          5
         128GB Flash Storage                4
         512GB SSD +  2TB HDD               3
         16GB SSD                           3
         512GB Flash Storage                2
         1TB SSD +  1TB HDD                 2
         256GB SSD +  500GB HDD             2
         128GB SSD +  2TB HDD               2
         256GB SSD +  256GB SSD             2
```

```
Name: Gpu brand, dtype: int64
```

In [71]:
```
sns.barplot(x=df['Gpu brand'],y=df['Price'],estimator=np.median)
plt.xticks(rotation='vertical')
plt.show()
```



In [72]:
```
df.drop(columns=['Gpu'],inplace=True)
```

In [73]:
```
df.head()
```

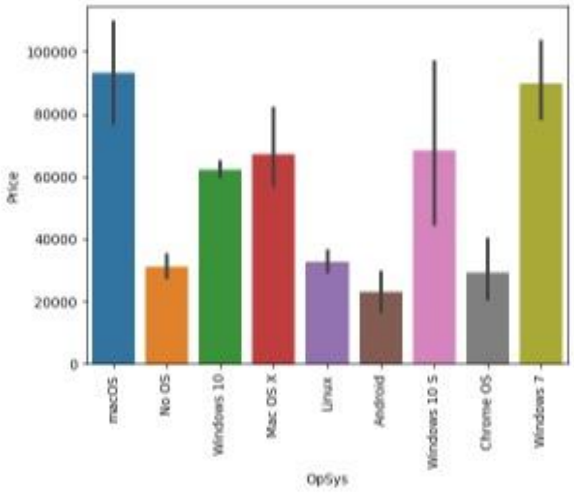
Out[73]:

	Company	TypeName	Ram	OpSys	Weight	Price	Touchscreen	Ips	ppi	Cpu brand	HDD	SSD	Gpu brand
0	Apple	Ultrabook	8	macOS	1.37	71378.6832	0	1	226.983005	Intel Core i5	0	128	Intel
1	Apple	Ultrabook	8	macOS	1.34	47895.5232	0	0	127.677940	Intel Core i5	0	0	Intel
2	HP	Notebook	8	No OS	1.86	30636.0000	0	0	141.211998	Intel Core i5	0	256	Intel
3	Apple	Ultrabook	16	macOS	1.83	135195.3360	0	1	220.534624	Intel Core i7	0	512	AMD
4	Apple	Ultrabook	8	macOS	1.37	96095.8080	0	1	226.983005	Intel Core i5	0	256	Intel

In [74]:
```
df['OpSys'].value_counts()
```

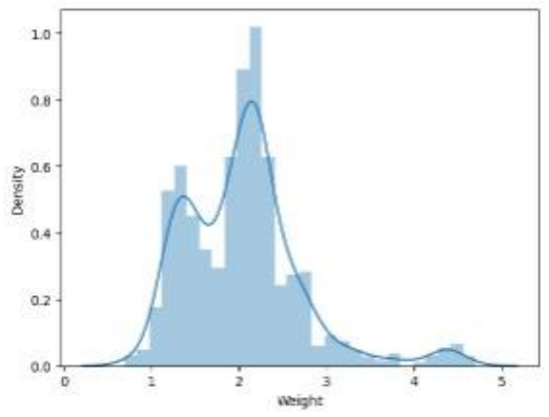
Out[74]:
```
Windows 10      1072
No OS             66
Linux             62
Windows 7         45
Chrome OS         26
macOS             13
Mac OS X           8
Windows 10 S       8
Android            2
Name: OpSys, dtype: int64
```

In [75]:
```
sns.barplot(x=df['OpSys'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



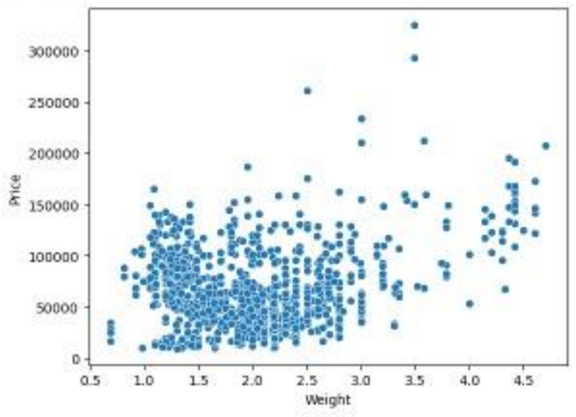
In [76]:
```
def cat_os(inp):
    if inp == 'Windows 10' or inp == 'Windows 7' or inp == 'Windows 10 S':
        return 'Windows'
    elif inp == 'macOS' or inp == 'Mac OS X':
        return 'Mac'
    else:
        return 'Others/No OS/Linux'
```

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 (ipykernel)



In [82]:
```
sns.scatterplot(x=df['Weight'],y=df['Price'])
```

Out[82]:
```
<Axes: xlabel='Weight', ylabel='Price'>
```



In [83]:
```
df.corr()['Price']
```

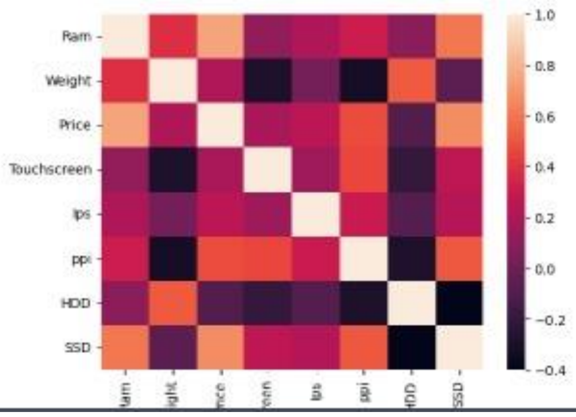
```
C:\Users\user90\AppData\Local\Temp\ipykernel_4692\815546952.py:1: FutureWarning: The default value of numeric_only in DataFram
e.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  df.corr()['Price']
```

Out[83]:
```
Ram            0.742985
Weight         0.209867
Price          1.000000
Touchscreen    0.192917
Ips            0.253320
ppi            0.475368
HDD           -0.096891
SSD            0.670660
Name: Price, dtype: float64
```

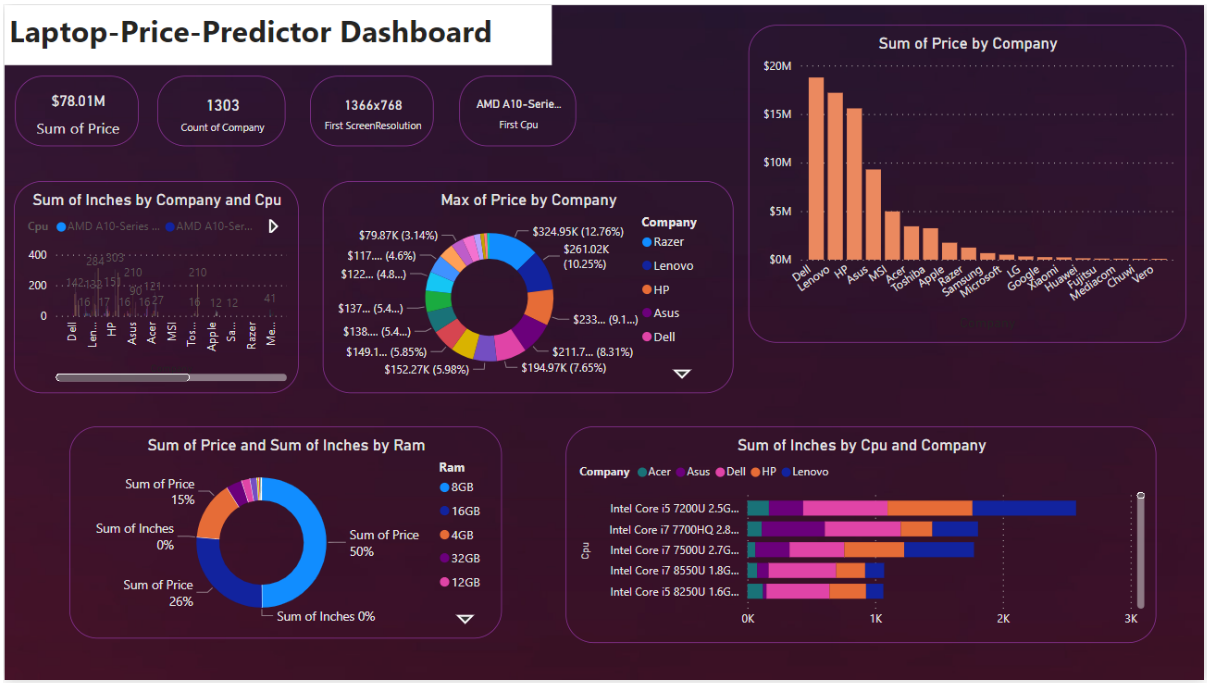
In [84]:
```
sns.heatmap(df.corr())
```

```
C:\Users\user90\AppData\Local\Temp\ipykernel_4692\58359773.py:1: FutureWarning: The default value of numeric_only in DataFrame.
corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_on
ly to silence this warning.
  sns.heatmap(df.corr())
```

Out[84]:
```
<Axes: >
```



Dashboard using Power BI


Laptop-Price-Predictor Dashboard
$78.01M
Sum of Price
1303
Count of Company
1366x768
First ScreenResolution
AMD A10-Serie...
First Cpu
Sum of Inches by Company and Cpu
Max of Price by Company
Company
Razer
Lenovo
HP
Asus
Dell
Sum of Price by Company
Sum of Price and Sum of Inches by Ram
Ram
8GB
16GB
4GB
32GB
12GB
Sum of Inches by Cpu and Company
Intel Core i5 7200U 2.5G...
Intel Core i7 7700HQ 2.8...
Intel Core i7 7500U 2.7G...
Intel Core i7 8550U 1.8G...
Intel Core i5 8250U 1.6G...

REFERENCES

[1] https://www.google.com

[2] https://www.w3schools.com/python/numpy/numpy_intro.asp

[3] https://seaborn.pydata.org/

[4] pandas - Python Data Analysis Library (pydata.org)

[5] https://jupyter.org/

[6] https://www.kaggle.com