University of Sulaimani

Computer department

30/1/2024

(Computer models for price prediction)

Prepared by :

Komar abdulla

Mabast hiwa

Sarhang kamal

Barez baxtiar

Content

Introduction

The "Laptop Price Prediction Project" is a pioneering endeavor in the realm of data-driven decision-making, utilizing advanced machine learning techniques to predict laptop prices. As laptops have become indispensable in our daily lives, the intricacies of their pricing mechanisms have gained paramount importance. This project was conceived to address the challenges posed by the lack of transparency in laptop pricing, providing a predictive model that empowers consumers, manufacturers, and retailers a like. The ubiquity of laptops in various aspects of life, from work to education and entertainment, makes understanding their pricing dynamics crucial. Transparent pricing not only facilitates informed consumer choices but also fosters healthy competition within the laptop industry. By offering a solution to the intricate pricing problem, the project aims to contribute to the efficiency and fairness of the laptop market.

Problem Statement

In delving into the heart of the matter, the project seeks to address the absence of a standardized pricing model for laptops. The diverse array of specifications and features in the laptop market poses a significant challenge for consumers attempting to evaluate the true value of a device. This lack of transparency not only complicates the purchasing process but also hampers market efficiency.

The significance of this problem lies in its far-reaching consequences. Transparent pricing is not only a matter of consumer satisfaction but also a driver of fair competition. An informed consumer is an empowered consumer, and fair competition is the cornerstone of a healthy market. By shedding light on the intricacies of laptop pricing, the project endeavors to foster a market environment where both buyers and sellers can make decisions based on a clear understanding of value.

Solution Method

The chosen solution method involves the application of a Decision Tree Regression model, a machine learning algorithm renowned for its ability to predict numerical values. The design process is meticulous, starting with the thoughtful selection of features that are likely to influence laptop prices. The methodology extends to data cleaning procedures to ensure the integrity of the dataset and visualization techniques to gain a comprehensive understanding of the data's patterns.

The decision tree model is the linchpin of the project, trained on a carefully curated subset of the data. This process is integral in ensuring the model's ability to generalize to new, unseen data—a critical aspect for practical application. The choice of a decision tree model aligns with the project's commitment to transparency, as these models are not only powerful predictors but also offer interpretability, allowing stakeholders to understand the reasoning behind predictions.

Implementation

Moving beyond the theoretical aspects, this page is dedicated to showcasing the tangible implementation of the project. Capturing the screen, the audience will be guided through the actual code and processes involved in bringing the project to life. Screenshots and code snippets will be interspersed, providing a visual narrative of the project's journey.

Starting with the loading of the dataset, the audience will witness the intricate steps of data cleaning and the subsequent visualization of key insights. The project's heartbeat, the decision tree model, will be presented in its training phase, emphasizing the transformation of data into a predictive tool. The

implementation segment aims not only to illustrate the technical prowess of the project but also to demystify the complex processes for a diverse audience.

Import the libraries :

In [4]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt
import seaborn as sns


from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder



df = pd.read_csv(r"C:\Users\DELL\Desktop\laptop.csv")
df = df.drop('Unnamed: 0', axis=1)
```

Showing the 5 first and 5 last rows of the dataset:

In [5]:
```
df.head()
```
Out[5]:

	Manufacturer	Category	Screen	GPU	OS	CPU_core	Screen_Size_cm	CPU_frequency	RAM_GB	Storage_GB_SSD	Weight_kg	Price
0	Acer	4	IPS Panel	2	1	5	35.560	1.6	8	256	1.60	978
1	Dell	3	Full HD	1	1	3	39.624	2.0	4	256	2.20	634
2	Dell	3	Full HD	1	1	7	39.624	2.7	8	256	2.20	946
3	Dell	4	IPS Panel	2	1	5	33.782	1.6	8	128	1.22	1244
4	HP	4	Full HD	2	1	7	39.624	1.8	8	256	1.91	837

In [6]:
```
df.tail()
```
Out[6]:

	Manufacturer	Category	Screen	GPU	OS	CPU_core	Screen_Size_cm	CPU_frequency	RAM_GB	Storage_GB_SSD	Weight_kg	Price
233	Lenovo	4	IPS Panel	2	1	7	35.560	2.6	8	256	1.70	1891
234	Toshiba	3	Full HD	2	1	5	33.782	2.4	8	256	1.20	1950
235	Lenovo	4	IPS Panel	2	1	5	30.480	2.6	8	256	1.36	2236
236	Lenovo	3	Full HD	3	1	5	39.624	2.5	6	256	2.40	883
237	Toshiba	3	Full HD	2	1	5	35.560	2.3	8	256	1.95	1499

In [7]:
```
df.shape
```

Showing some statistical information on the dataset :

In [10]:
```
df.describe()
```
Out[10]:

	Category	GPU	OS	CPU_core	Screen_Size_cm	CPU_frequency	RAM_GB	Storage_GB_SSD	Weight_kg	Price
count	238.000000	238.000000	238.000000	238.000000	234.000000	238.000000	238.000000	238.000000	233.000000	238.000000
mean	3.205882	2.151261	1.058824	5.630252	37.269615	2.360084	7.882353	245.781513	1.862232	1462.344538
std	0.776533	0.638282	0.235790	1.241787	2.971365	0.411393	2.482603	34.765316	0.494332	574.607699
min	1.000000	1.000000	1.000000	3.000000	30.480000	1.200000	4.000000	128.000000	0.810000	527.000000
25%	3.000000	2.000000	1.000000	5.000000	35.560000	2.000000	8.000000	256.000000	1.440000	1066.500000
50%	3.000000	2.000000	1.000000	5.000000	38.100000	2.500000	8.000000	256.000000	1.870000	1333.000000
75%	4.000000	3.000000	1.000000	7.000000	39.624000	2.700000	8.000000	256.000000	2.200000	1777.000000
max	5.000000	3.000000	2.000000	7.000000	43.942000	2.900000	16.000000	256.000000	3.600000	3810.000000

In [11]:
```
df.isna().sum()
```
Out[11]:
```
Manufacturer      0
Category          0
Screen            0
GPU               0
OS                0
CPU_core          0
Screen_Size_cm    4
CPU_frequency     0
RAM_GB            0
Storage_GB_SSD    0
Weight_kg         5
Price             0
dtype: int64
```

Filling the null columns by average and then check the null values :

```
In [13]: avg_screen_size = df['Screen_Size_cm'].mean()
         df['Screen_Size_cm'] = df['Screen_Size_cm'].fillna(avg_screen_size)
```

```
In [14]: avg_Weight_kg = df['Weight_kg'].mean()
         df['Weight_kg'] = df['Weight_kg'].fillna(avg_Weight_kg)
```

```
In [15]: df.isna().sum()
```

```
Out[15]: Manufacturer      0
         Category          0
         Screen            0
         GPU               0
         OS                0
         CPU_core          0
         Screen_Size_cm    0
         CPU_frequency     0
         RAM_GB            0
         Storage_GB_SSD    0
         Weight_kg         0
         Price             0
         dtype: int64
```

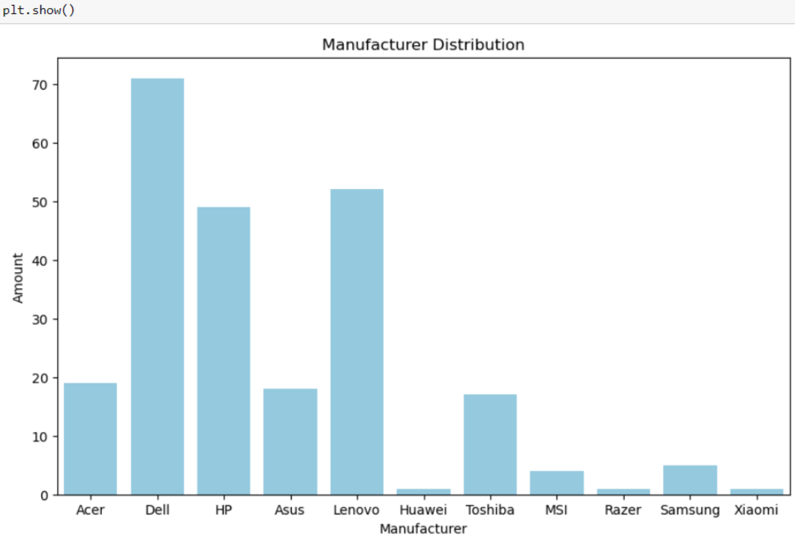
```
In [16]: plt.figure(figsize=(10,6))
         sns.countplot(x='Manufacturer', data=df, color='skyblue')
         plt.xlabel('Manufacturer')
         plt.ylabel('Amount')
         plt.title('Manufacturer Distribution')

         plt.show()
```

Create countplot graph to explain amount if manufacture distribution :

```
In [16]: plt.figure(figsize=(10,6))
         sns.countplot(x='Manufacturer', data=df, color='skyblue')
         plt.xlabel('Manufacturer')
         plt.ylabel('Amount')
         plt.title('Manufacturer Distribution')

         plt.show()
```



Create correlation between the columns of the dataset :

In [17]:
```python
numeric_columns = df.select_dtypes(include=['number'])
correlation_matrix = numeric_columns.corr()
print(correlation_matrix)
```

```
                Category       GPU        OS  CPU_core  Screen_Size_cm  \
Category        1.000000 -0.114174 -0.043378  0.232425       -0.305035
GPU            -0.114174  1.000000 -0.199549  0.145388        0.152979
OS             -0.043378 -0.199549  1.000000  0.016954        0.150835
CPU_core        0.232425  0.145388  0.016954  1.000000        0.037293
Screen_Size_cm -0.305035  0.152979  0.150835  0.037293        1.000000
CPU_frequency  -0.053414  0.291439  0.050407  0.242722       -0.002262
RAM_GB          0.030127  0.218973 -0.074625  0.473075        0.017651
Storage_GB_SSD  0.038246  0.094288  0.007751  0.400015        0.116368
Weight_kg      -0.381032  0.262853  0.120858  0.068599        0.810703
Price           0.286243  0.288298 -0.221730  0.459398       -0.126672

                CPU_frequency    RAM_GB  Storage_GB_SSD  Weight_kg     Price
Category            -0.053414  0.030127        0.038246  -0.381032  0.286243
GPU                  0.291439  0.218973        0.094288   0.262853  0.288298
OS                   0.050407 -0.074625        0.007751   0.120858 -0.221730
CPU_core             0.242722  0.473075        0.400015   0.068599  0.459398
Screen_Size_cm      -0.002262  0.017651        0.116368   0.810703 -0.126672
CPU_frequency        1.000000  0.226736        0.035557   0.066522  0.366666
RAM_GB               0.226736  1.000000        0.361469   0.055068  0.549297
Storage_GB_SSD       0.035557  0.361469        1.000000   0.112519  0.243421
Weight_kg            0.066522  0.055068        0.112519   1.000000 -0.050312
Price                0.366666  0.549297        0.243421  -0.050312  1.000000
```

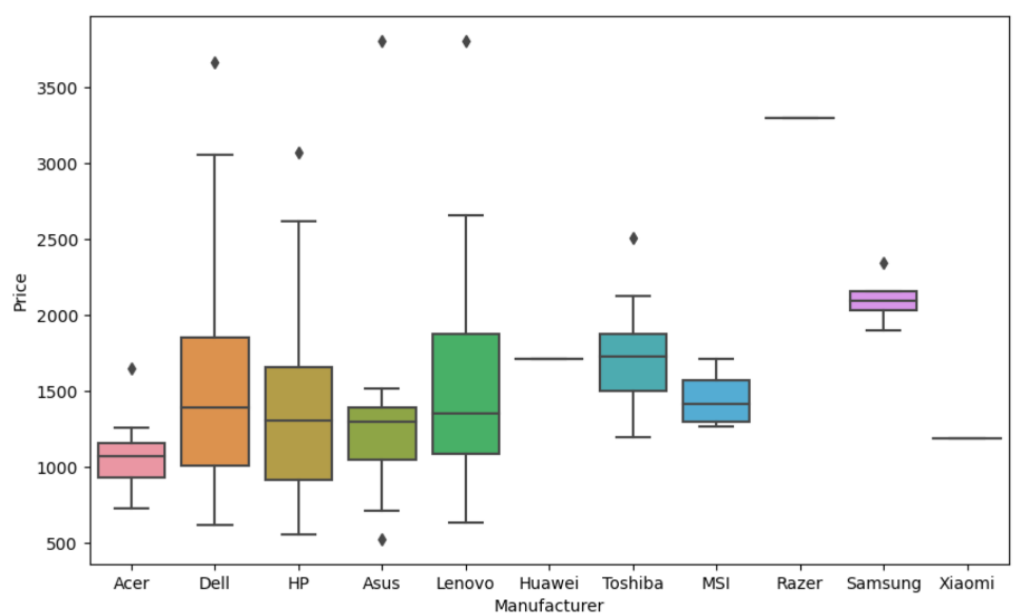
In [18]:
```python
plt.figure(figsize=(10,6))
sns.histplot(df['Price'], color='skyblue')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.title('Price Distribution Distribution')

plt.show()
```

Create boxplot between manufacturer and price to explain between two columns :

In [19]:
```python
plt.figure(figsize=(10,6))
sns.boxplot(x='Manufacturer', y='Price', data=df)
plt.xlabel('Manufacturer')
plt.ylabel('Price')

plt.show()
```



Results Discussion

The crux of any data-driven project lies in its results, and this page serves as the stage for presenting and dissecting the outcomes. Performance metrics, with a particular focus on the Mean Absolute Error, will be unveiled. To provide a comprehensive understanding, relevant graphs and visualizations will accompany the discussion, shedding light on the intricate relationship between various features and laptop prices. The insights gained from the results are pivotal in drawing conclusions about the project's success in addressing the pricing problem. This section goes beyond mere presentation, encouraging an interactive dialogue with the audience. Questions about the significance of specific features, outliers in the dataset, and the real-world implications of the findings will be welcomed, fostering a dynamic exchange o f ideas.

Project Conclusion

the project reaches its denouement, the concluding page serves as a reflective space. A succinct yet comprehensive summary of the achievements, challenges, and learnings will be presented. This is not merely a recollection but an opportunity to distill the essence of the "Laptop Price Prediction Project."

Acknowledging the achievements, such as the successful implementation of a predictive model, is crucial. However, the conclusion goes beyond the immediate victories, delving into the broader impact and implications of the project. It serves as a platform to discuss the societal, economic, and technological ramifications of a transparent pricing model for laptops.

Looking forward, the conclusion is not merely an endpoint but a threshold to future possibilities. Plans and considerations for future developments or improvements will be outlined, providing a roadmap for the continuous evolution of the project. Whether it involves expanding the scope to include more features, exploring advanced machine learning techniques, or collaborating with industry stakeholders, the future considerations aim to invigorate the project's trajectory.

In essence, the "Laptop Price Prediction Project" is not just a technical feat; it is a manifestation of the potential of data science in reshaping industries and empowering individuals. By addressing the intricacies of laptop pricing, the project contributes to the broader discourse on technology, transparency, and market dynamics. It stands as a testament to the collaborative efforts of individuals striving to bring about positive change in the world of technology economics.

The references:

1) DecisionTreeRegressor Documentation. Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

2) Seaborn Documentation. https://seaborn.pydata.org/

3) Matplotlib Documentation. https://matplotlib.org/

4) Pandas Documentation. https://pandas.pydata.org/

5) NumPy Documentation. https://numpy.org/