Laptop Price Detector

--By Rohan Chaudhari

- For this project, I'll be utilising Jupyter notebook, and the dataset I'll be using is the laptop pricing dataset.
- There are a total of 12 features and 1303 rows. I have taken this dataset because it requires extensive feature engineering.
- Initially I have loaded all the required libraries like pandas, numpy, seaborn etc.
- Then I have loaded the dataset into the dataframe using pandas. To view the first 5 rows I have used df.head().
- After that I found the shape and the unique values that are present in the data
- Then I checked weather there are any null and duplicate values present in the dataset and found that there are no null and duplicate values.
- After checking for the info about the data I came to know that there is a column 'Unnamed' which is useless so I will drop that column.
- The numeric number for 'weights' and 'Ram' size is followed by a string, resulting in it being displayed as object datatype. So, in order to fix it, I'm going to remove those strings and change their datatypes to float and int.
- It's evident from the first graph that there are relatively few computers with a price tag of more than $150,000. As a result, we may claim that the data is biassed toward lower values, thus we'll use the log of the pricing column to fix the problem.
- I learned about the major producers via the company graph. Then I plotted company vs. price and discovered that pricing are highly dependent on the firm
- I followed the above step for other columns as well and came to know that the inches don't have a very high co-relation with the prices. So I dropped that column.
- I have used df.corr to find the correlation between prices and the other columns.

Screen Resolution column processing:

- For the'Screen Resolution'column, I first found the different values it has a string with some information like whether the laptop is touchscreen or not, whether it has an IPS panel, and so on. To extract this information, we first split the data in Screen Resolution column and then make two columns Touchscreen and IPS and put the value 1 in touchscreen when the laptop is touchscreen and 0 when the laptop is not touchscreen. For IPS panels, we follow the same procedure. Then we simply plot the price of a touch screen and an IPS panel against each other to see how much the existence of an IPS and touchscreen affects the pricing.
- After separating the touch screen and the IPS panel, we go on to the resolution. To begin, I created two new columns for X and Y Resolution, then multiplied them to create a Res column and then a PPI (pixels per inch) column.
- After completing these columns I dropped the main Screen Resolution column.

Cpu column processing:

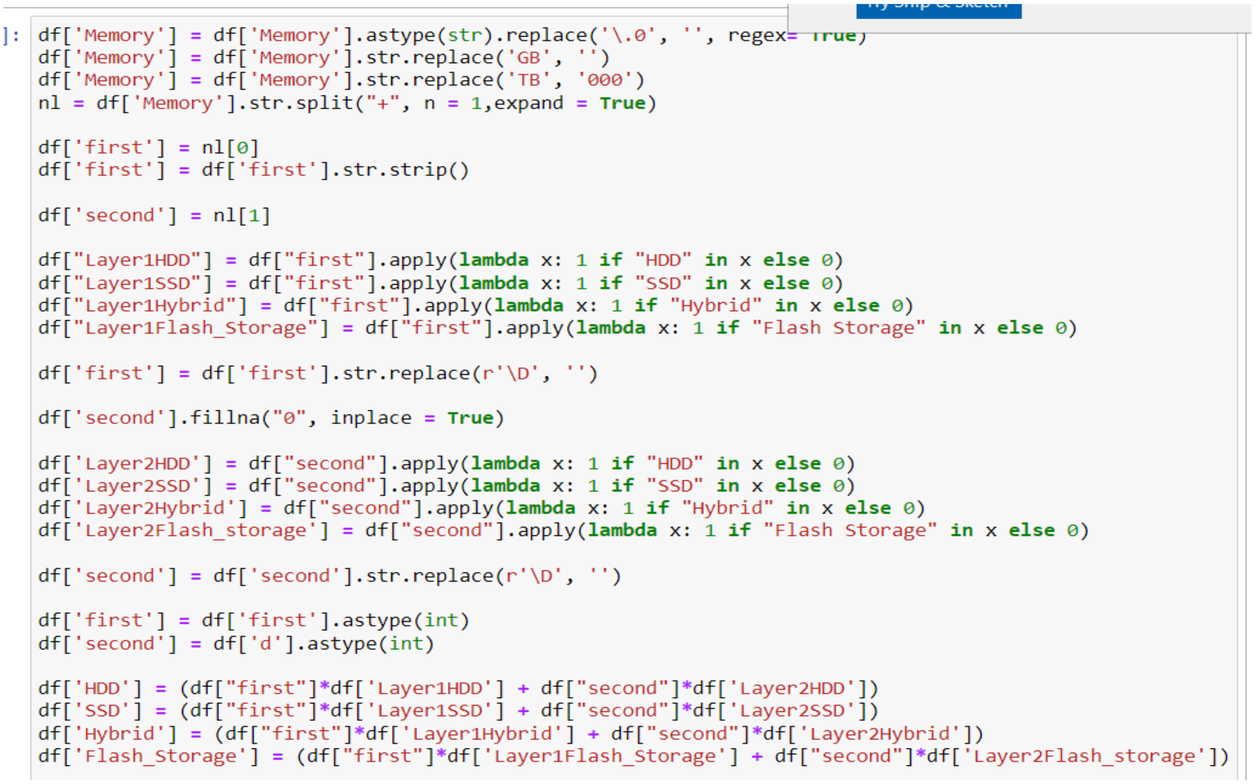
- For the CPU column initially I found the different values contained in that column .
- Then I separated the different values and made a new column containing CPU Name. Since the whole cpu name was not that useful I just took the first 3 word and made categories out of them.
- I wrote a processor function to divde the multiple CPU Names into 3 most significant categories.
- After making this column I dropped the main cpu column.

Ram column processing:

- Here I didn’t do any processing as initially only I have removed the GB string attached to it.
- I just plotted Ram vs prices to see the type of relation and found that as Ram size increased price increases.

Memory column processing:

- Here also I first found the number of different values.
- After looking at the values I came to know that we have a string ‘GB’ & ‘TB’ attached to the integers. We also have a ‘+’ sign indicating that the there is some additional hard-disk present in the system.
- So first I removed the GB & TB string and replace the TB with 000 as 1TB = 1000GB. Then taking the ‘+’ as separator I separated the values.
- After separating I placed them in 2 temporary column namely first and second. I also made some temporary column names ‘Layer’ for indicatind weather there is a HDD, SSD, Hybrid or Flash_storage present.
- In column named first we will have the numeric value and in the ‘layer’ column will tell us weather it’s a HDD, SSD, Hybrid or Flash_Storage. I did the same for the ‘second’ column
- Once this was done I created new column namely HDD, SSD, Hybrid, Flash_storage. I multipled the ‘first’ column with the ‘layer’ column and stored the values in this column.
- After completing this I deleted the original memory and all the temporary columns.



```
]: df['Memory'] = df['Memory'].astype(str).replace('\.0', '', regex= True)
   df['Memory'] = df['Memory'].str.replace('GB', '')
   df['Memory'] = df['Memory'].str.replace('TB', '000')
   nl = df['Memory'].str.split("+", n = 1,expand = True)

   df['first'] = nl[0]
   df['first'] = df['first'].str.strip()

   df['second'] = nl[1]

   df["Layer1HDD"] = df["first"].apply(lambda x: 1 if "HDD" in x else 0)
   df["Layer1SSD"] = df["first"].apply(lambda x: 1 if "SSD" in x else 0)
   df["Layer1Hybrid"] = df["first"].apply(lambda x: 1 if "Hybrid" in x else 0)
   df["Layer1Flash_Storage"] = df["first"].apply(lambda x: 1 if "Flash Storage" in x else 0)

   df['first'] = df['first'].str.replace(r'\D', '')

   df['second'].fillna("0", inplace = True)

   df['Layer2HDD'] = df["second"].apply(lambda x: 1 if "HDD" in x else 0)
   df['Layer2SSD'] = df["second"].apply(lambda x: 1 if "SSD" in x else 0)
   df['Layer2Hybrid'] = df["second"].apply(lambda x: 1 if "Hybrid" in x else 0)
   df['Layer2Flash_storage'] = df["second"].apply(lambda x: 1 if "Flash Storage" in x else 0)

   df['second'] = df['second'].str.replace(r'\D', '')

   df['first'] = df['first'].astype(int)
   df['second'] = df['d'].astype(int)

   df['HDD'] = (df["first"]*df['Layer1HDD'] + df["second"]*df['Layer2HDD'])
   df['SSD'] = (df["first"]*df['Layer1SSD'] + df["second"]*df['Layer2SSD'])
   df['Hybrid'] = (df["first"]*df['Layer1Hybrid'] + df["second"]*df['Layer2Hybrid'])
   df['Flash_Storage'] = (df["first"]*df['Layer1Flash_Storage'] + df["second"]*df['Layer2Flash_storage'])
```

GPU column processing:

- Here also I first found the different values present in gpu column.
- After that I segmented these based on the GPU producer and stored these into new column 'GPU Name '.
- For Arm their was only one row so I dropped it.
- Then I plotted the price vs GPU Name to see the relation.
- After finishing this I dropped the gpu column.

OS column processing:

- Here also I first found the different values present.
- I came to know that the values can be divided into three major categories 'Windows', 'Mac' and 'Others'.
- So to do this I wrote a function and applied it on the 'Opsys' column and made a new column 'OS_Name' from it.
- After this I plotted the OS_Name with price to see the relation and then deleted the original column.

- After completing this data cleaning and processing I splitted the data into training and test. Then I used a column transfer function to encode the categorical data.

- After this using a pipeline structure I applied the Linear Regression model on the data and found the r2 score and the mean_absolute_error.