Project Report
On

Laptop Price Prediction

Submitted for the requirement of

Project Course

Bachelor of Engineering

Computer Science and Engineering

Submitted by:
Danish Gupta
20BCS5160

Department of Computer Science & Engineering Chandigarh University, Gharuan

DECLARATION

I hereby declare that I have completed my six weeks summer training at **INTERNSHALA** from **15/06/2022 to 27/07/2022** under the guidance of **INTERNSHALA**. I hereby undertakethat the project undertaken by me is the genuine work of mine

NAME: - DANISH GUPTA
UID: - 20BCS5160
DATE: - 09/10/2022

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Department of Computer Science Engineering at University Institute of Engineering, Chandigarh University, Gharuan Campus for providing us opportunity to implement the knowledge gained over these years as. We would also like to express our deepest sense of gratitude and thanks to our university and Intershala for providing invaluable insight and guidelines for this project. We would also like to thank all of our friends who have directly and indirectly helped us in doing this project. Last but not the least, we place a deep sense of appreciation to our family members who have been constant source of inspiration for us.
Any kind of suggestion or criticism will be highly appreciated and acknowledged.

Summer training certificate

INTERNSHALA TRAININGS

Certificate of Training

Danish Gupta

from Chandigarh University has successfully completed a 6-week online training on **Machine Learning**. The training consisted of Introduction to Machine Learning, Data, Introduction to Python, Data Exploration and Pre-processing, Linear Regression, Introduction to Dimensionality Reduction, Logistic Regression, Decision Tree, Ensemble Models, and Clustering (Unsupervised Learning) modules.

Danish is a top performer in the training.

We wish Danish all the best for future endeavours.

Sarvesh Agarwal

FOUNDER & CEO, INTERNSHALA

Date of certification: 2022-07-26 Certificate no. : CCAB67E4-189F-D986-4657-2F7B4A216C1F

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

ABSTRACT

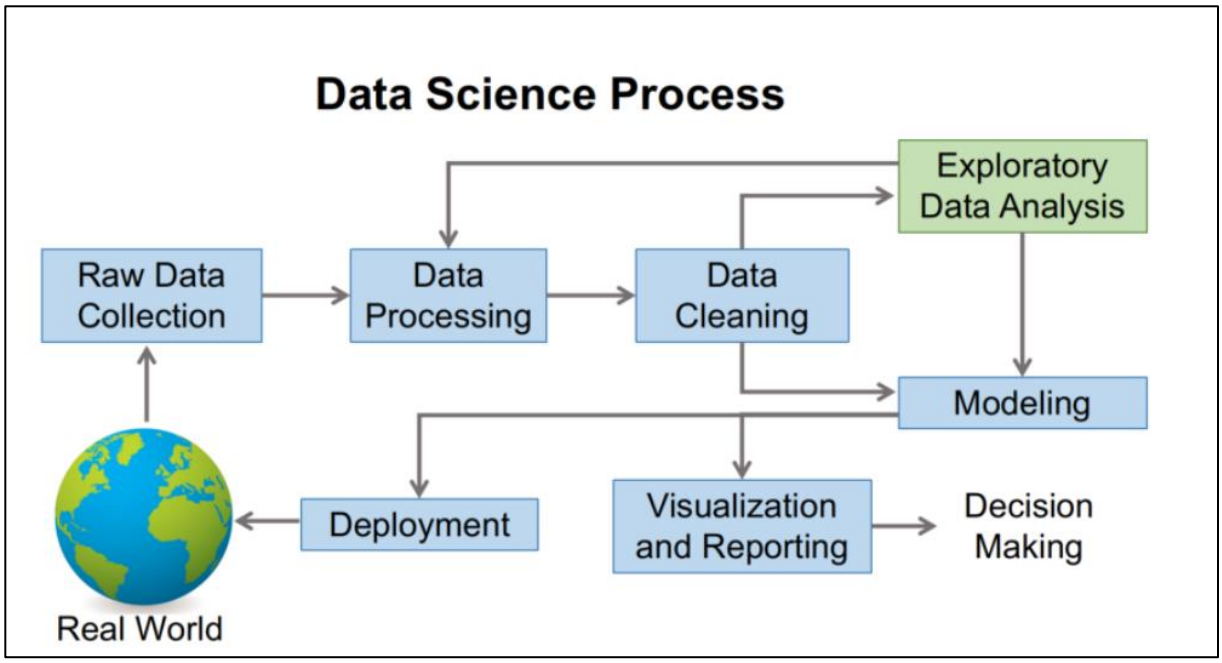
This report presents a laptop price prediction system by using the supervised machine learning technique. The research uses multiple linear regression as the machine learning prediction method which offered 81% prediction precision. Using multiple linear regression, there are multiple independent variables but one and only one dependent variable whose actual and predicted values are compared to find precision of results. This paper proposes a system where price is dependent variable which is predicted, and this price is derived from factors like Laptop's model, RAM, ROM (HDD/SSD), GPU, CPU, IPS Display, and Touch Screen. Laptop price prediction especially when the laptop is coming direct from the factory to Electronic Market/ Stores, is both a critical and important task. The mad rush that we saw in 2020 for laptops to support remote work and learning is no longer there. In India, demand of Laptops soared after the Nationwide lockdown, leading to 4.1-Million-unit shipments in the June quarter of 2021, the highest in the five years. Accurate Laptop price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, RAM, ROM, GPU, CPU, etc. In this paper, we applied different methods and techniques in order to achieve higher precision of the used laptop price prediction

TABLE OF CONTENTS

1. Introduction
2. Data cleaning
3. Exploratory Data Analysis
4. Training and testing dataset
5. Decision Tree Algorithm
6. Exporting the model and Streamlit
7. Result and discussion
8. Conclusion

1. INTRODUCTION

Laptop price prediction especially when the laptop is coming direct from the factory to Electronic Market/Stores, is both a critical and important task. The mad rush that we saw in 2020 for laptops to support remote work and learning is no longer there. In India, demand of Laptops soared after the Nationwide lockdown, leading to 4.1-Million-unit shipments in the June quarter of 2021, the highest in the five years. Accurate Laptop price prediction involves expert knowledge because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, RAM, ROM, GPU, CPU, etc. In this paper, we applied different methods and techniques to achieve higher precision of the laptop price prediction. Predicting price of laptops has been studied extensively in various research. Listian discussed, in her paper written for Master thesis, that regression model that was built using Decision Tree & Random Forest Regressor can predict the price of a laptop that has been leased with better precision than multivariate regression or some simple multiple regression. This is on the grounds that Decision Tree Algorithm is better in dealing with datasets with more dimensions and it is less prone to overfitting and underfitting. The weakness of this research is that a change of simple regression with more advanced Decision Tree Algorithm regression was not shown in basic indicators like mean, variance, or standard deviation.



This is the dataset that we are going to use in our laptop price prediction model-

	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
5	5	Acer	Notebook	15.6	1366x768	AMD A9-Series 9420 3GHz	4GB	500GB HDD	AMD Radeon R5	Windows 10	2.1kg	21312.0000
6	6	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.2GHz	16GB	256GB Flash Storage	Intel Iris Pro Graphics	Mac OS X	2.04kg	114017.6016
7	7	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	256GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	61735.5360
8	8	Asus	Ultrabook	14.0	Full HD 1920x1080	Intel Core i7 8550U 1.8GHz	16GB	512GB SSD	Nvidia GeForce MX150	Windows 10	1.3kg	79653.6000
9	9	Acer	Ultrabook	14.0	IPS Panel Full HD 1920x1080	Intel Core i5 8250U 1.6GHz	8GB	256GB SSD	Intel UHD Graphics 620	Windows 10	1.6kg	41025.6000

Software and Hardware tools required:

1. JUPYTER
2. WEB BROWSER
3. STREAMLIT
4. ANY COMPUTER Having 4 GB Ram
5. ANACONDA NAVIGATOR

2.DATA CLEANING

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. But, as we mentioned above, it isn't as simple as organizing some rows or erasing information to make space for new data. Data cleaning is a lot of muscle work. There's a reason data cleaning is the most important step if you want to create a data-culture, let alone make airtight predictions. It involves:

- Fixing spelling and syntax errors
- Standardizing data sets
- Correcting mistakes such as empty fields
- Identifying duplicate data points

It's said that the majority of a data scientist's time is spent on cleaning, rather than machine learning. In fact, 45% of data scientist's time is spent on preparing data.

And to us, that makes sense—if there's data that doesn't belong in your dataset, you aren't going to get accurate results. And with so much data these days, usually combined from multiple sources, and so many critical business decisions to make, you want to be extra sure that your data is clean.

Why is Data Cleaning so Important?

Businesses have a plethora of data. But not all of it is accurate or organized. When it comes to machine learning, if data is not cleaned thoroughly, the accuracy of your model stands on shaky grounds.

These are the kinds of benefits you will see:

- Better decision making
- Boost in revenue
- Save time
- Increase productivity
- Streamline business practice

How to Clean Your Data?

Once you know what to look out for, it's easier to know what to look for when prepping your data. While the techniques used for data cleaning may vary depending on the type of data you're working with, the steps to prepare your data are fairly consistent.
Here are some steps you can take to properly prepare your data.

1. Remove duplicate observations
Duplicate data most often occurs during the data collection process. This typically happens when you combine data from multiple places or receive data from clients or multiple departments. You want to remove any instances where duplicate data exists.
You also want to remove any irrelevant observations from your dataset. This is where you data doesn't fit into the specific problem you're trying to analyze. This will help you make your analysis more efficient.

2. Filter unwanted outliers
Outliers are unusual values in your dataset. They're significantly different from other data point and can distort your analysis and violate assumptions. Removing them is a subjective practice and depends on what you're trying to analyze. Generally speaking, removing unwanted outliers will help improve the performance of the data you're working with.
Remove an outlier if:

You know that it's wrong. For example, if you have a really good sense of what range the data should fall in, like people's ages, you can safely drop values that are outside of that range.
You have a lot of data. Your sample won't be hurt by dropping a questionable outlier.
You can go back and recollect. Or, you can verify the questionable data point.
Remember: just because an outlier exists, doesn't mean it is incorrect. Sometimes an outlier will help, for instance, prove a theory you're working on. If that's the case, keep the outlier.

3. Fix structural errors
Structural errors are things like strange naming conventions, typos, or incorrect capitalization. Anything that is inconsistent will create mislabeled categories.
A good example of this is when you have both "N/A" and "Not Applicable." Both are going to appear in separate categories, but they should both be analyzed as the same category.

4. Fix missing data
Make sure that any data that's missing is filled in.
A lot of algorithms won't accept missing values. You may either drop the observations that have missing values, or you may input the missing value based on other observations.

5. Validate your data

Once you've thoroughly prepped your data, you should be able to answer these questions to validate it:

Does your data make complete sense now?

Does the data follow the relevant rules for its category or class?

Does it prove/disprove your working theory?

Some of the information about our data is given in the following image

```
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
```

Removing all the null values from our data –

1. DATA CLEANING

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)
```

Changing the datatype of the variables –

```
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
```

3.EXPLORATORY DATA ANALYSIS (EDA)

Exploratory data analysis (EDA) involves different statistical procedures that are available to provide a researcher with a view of the data in terms of distribution and general characteristics of a given data. While analyzing data using SPSS, there are a variety of statistical procedures available to carry out exploratory data analysis. There include descriptive statistics such as median, mean, mode, maximum, minimum, variance, range, and standard deviation among other numerical summaries. In addition, there are graphical procedures that help in exploring data visually. These include scatter plots, bar charts, histograms, pie charts, stem, and leaf plots among others.

There are several reasons why it is critical to perform exploratory data analysis during data analysis. Any set of data is prone to errors that may be introduced during collection or data entry. Using EDA, it is possible to identify such errors with outliers being among the best indicators of errors in data. It is possible to identify outliers in a data set using box plots. Since this is a visual form of exploratory data analysis, it is easy to clearly identify values that are abnormal in the data. When collecting data, it is impossible to identify some features such as pattern of distribution without performing an analysis of the data. This is where exploratory data analysis proves invaluable. Using the skewness measure, it is possible to identify how data distributed from the mean. When data is skewed to the left, this indicates that most of the values lie on the left side of the mean whereas when data is skewed to the right, more data lies on the right side of the mean. When most data values are concentrated at the mean, the graph assumes a dome shape. Scatter plots are useful for displaying the distribution of data along the X- and Y-axes. The scatter plot is can be enriched by a regression line which indicates deviation of data from the line of best fit (from the area where most data lie). The regression line can have a positive or a negative gradient thus indicating a proportional or an inverse proportion in relationship between variables (Field, 2009).

Exploratory data analysis is also helpful in testing for normal distribution in data. Using normal probability plots, a researcher is able to define whether the mean, mode and median are the same. If these are the same, this is defined as perfect normal distribution. To identify differences in distributions, one can utilize the Kolmogorov-Smirnov & Shapiro-Wilk tests. Performing EDA helps in choosing between parametric and non-parametric tests for further data analysis. For numerical data with normal distribution, one can perform analysis using t-test or ANOVA. On the other hand, non-parametric methods include Chi-square test or Spearman correlation coefficient (Field, 2009). Missing data can also be dealt with using pairwise or listwise deletion during exploratory data analysis.

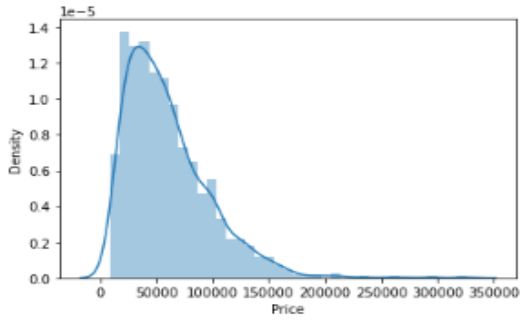
2. EDA

In [14]:
```
import seaborn as sns
```

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

```
C:\Users\danis\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarn
d will be removed in a future version. Please adapt your code to use either `displot
xibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

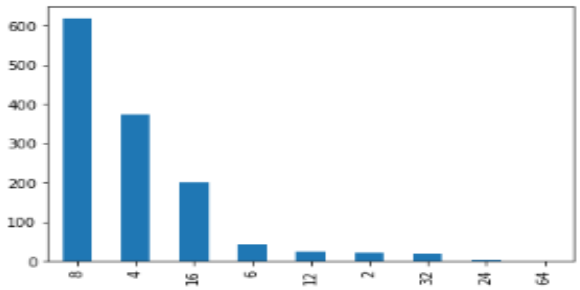
Out[15]: <AxesSubplot:xlabel='Price', ylabel='Density'>



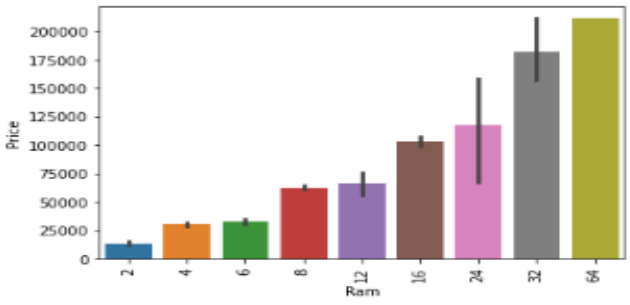
RAM

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

Out[55]: <AxesSubplot:>



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



Just like this we will perform EDA on all the other columns to get their best benefit.

4. TRAINING AND TESTING DATASET

Machine Learning is one of the booming technologies across the world that enables computers/machines to turn a huge amount of data into predictions. However, these predictions highly depend on the quality of the data, and if we are not using the right data for our model, then it will not generate the expected result. In machine learning projects, we generally divide the original dataset into training data and test data. We train our model over a subset of the original dataset, i.e., the training dataset, and then evaluate whether it can generalize well to the new or unseen dataset or test set. Therefore, train and test datasets are the two key concepts of machine learning, where the training dataset is used to fit the model, and the test dataset is used to evaluate the model.

In this topic, we are going to discuss train and test datasets along with the difference between both of them. So, let's start with the introduction of the training dataset and test dataset in Machine Learning.

What is Training Dataset?

The training data is the biggest (in -size) subset of the original dataset, which is used to train or fit the machine learning model. Firstly, the training data is fed to the ML algorithms, which lets them learn how to make predictions for the given task.

What is Test Dataset?

Once we train the model with the training dataset, it's time to test the model with the test dataset. This dataset evaluates the performance of the model and ensures that the model can generalize well with the new or unseen dataset. The test dataset is another subset of original data, which is independent of the training dataset. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world. Usually, the test dataset is approximately 20-25% of the total original data for an ML project.

At this stage, we can also check and compare the testing accuracy with the training accuracy, which means how accurate our model is with the test dataset against the training dataset. If the accuracy of the model on training data is greater than that on testing data, then the model is said to have overfitting.

The testing data should:

- Represent or part of the original dataset.
- It should be large enough to give meaningful predictions.

Need of Splitting dataset into Train and Test set

Splitting the dataset into train and test sets is one of the important parts of data pre-processing, as by doing so, we can improve the performance of our model and hence give better predictability.
We can understand it as if we train our model with a training set and then test it with a completely different test dataset, and then our model will not be able to understand the correlations between the features.

Train and Test datasets in Machine Learning

Therefore, if we train and test the model with two different datasets, then it will decrease the performance of the model. Hence it is important to split a dataset into two parts, i.e., train and test set.
In this way, we can easily evaluate the performance of our model. Such as, if it performs well with the training data, but does not perform well with the test dataset, then it is estimated that the model may be overfitted.

For splitting the dataset, we can use the train_test_split function of scikit-learn.

How do training and testing data work in Machine Learning?

Machine Learning algorithms enable the machines to make predictions and solve problems based on past observations or experiences. These experiences or observations an algorithm can take from the training data, which is fed to it. Further, one of the great things about ML algorithms is that they can learn and improve over time on their own, as they are trained with the relevant training data.

Once the model is trained enough with the relevant training data, it is tested with the test data. We can understand the whole process of training and testing in three steps, which are as follows:

Feed: Firstly, we need to train the model by feeding it with training input data.
Define: Now, training data is tagged with the corresponding outputs (in Supervised Learning), and the model transforms the training data into text vectors or several data features.
Test: In the last step, we test the model by feeding it with the test data/unseen dataset. This step ensures that the model is trained efficiently and can generalize well.

3. TRAIN TEST SPLIT

In [124]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.15,random_state=2)
```

In [91]:
```
X_train
```

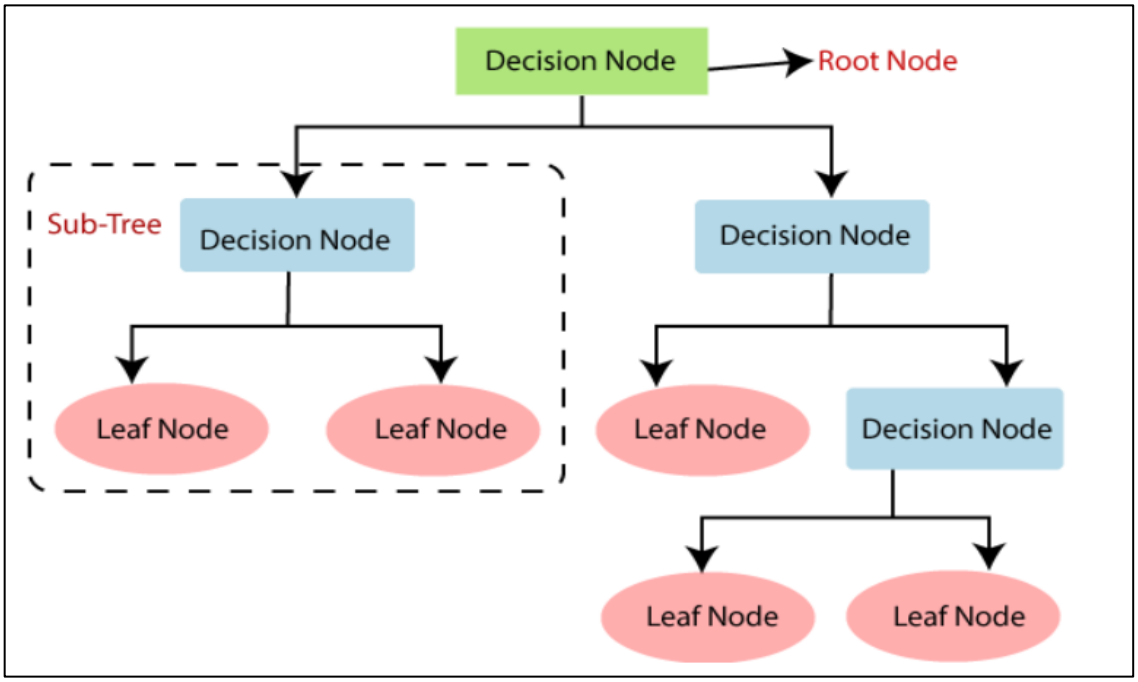
Out[91]:

	Company	TypeName	Ram	Weight	Touchscreen	Ips	ppi	Cpu brand	HDD	SSD	Gpu brand	os
183	Toshiba	Notebook	8	2.00	0	0	100.454670	Intel Core i5	0	128	Intel	Windows
1141	MSI	Gaming	8	2.40	0	0	141.211998	Intel Core i7	1000	128	Nvidia	Windows
1049	Asus	Netbook	4	1.20	0	0	135.094211	Other Intel Processor	0	0	Intel	Others/No OS/Linux
1020	Dell	2 in 1 Convertible	4	2.08	1	1	141.211998	Intel Core i3	1000	0	Intel	Windows
878	Dell	Notebook	4	2.18	0	0	141.211998	Intel Core i5	1000	128	Nvidia	Windows
...	...	...	...	...	...	...	...	...	...	...	...	...
466	Acer	Notebook	4	2.20	0	0	100.454670	Intel Core i3	500	0	Nvidia	Windows
299	Asus	Ultrabook	16	1.63	0	0	141.211998	Intel Core i7	0	512	Nvidia	Windows
493	Acer	Notebook	8	2.20	0	0	100.454670	AMD Processor	1000	0	AMD	Windows
527	Lenovo	Notebook	8	2.20	0	0	100.454670	Intel Core i3	2000	0	Nvidia	Others/No OS/Linux
1193	Apple	Ultrabook	8	0.92	0	1	226.415547	Other Intel Processor	0	0	Intel	Mac

1106 rows × 12 columns

5. DECISION TREE ALGORITHM

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome.
In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
The decisions or the test are performed based on features of the given dataset.
It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
It is called a decision tree because, like a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
A decision tree simply asks a question and based on the answer (Yes/No), it further split the tree into subtrees.



Applying DECISION TREE ALGORITHM,

Decision Tree

In [100]:

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = DecisionTreeRegressor(max_depth=8)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.841298665327169
MAE 0.18096109736609037
```

6. EXPORTING THE MODEL and STREAMLIT

Exporting the Model

In [119]:
```
import pickle

pickle.dump(df,open('df.pkl','wb'))
pickle.dump(pipe,open('pipe.pkl','wb'))
```

In [120]:
```
df
```

Out[120]:

	Company	TypeName	Ram	Weight	Price	Touchscreen	Ips	ppi	Cpu brand	HDD	SSD	Gpu brand	os
0	Apple	Ultrabook	8	1.37	71378.6832	0	1	226.983005	Intel Core i5	0	128	Intel	Mac
1	Apple	Ultrabook	8	1.34	47895.5232	0	0	127.677940	Intel Core i5	0	0	Intel	Mac
2	HP	Notebook	8	1.86	30636.0000	0	0	141.211998	Intel Core i5	0	256	Intel	Others/No OS/Linux
3	Apple	Ultrabook	16	1.83	135195.3360	0	1	220.534624	Intel Core i7	0	512	AMD	Mac
4	Apple	Ultrabook	8	1.37	96095.8080	0	1	226.983005	Intel Core i5	0	256	Intel	Mac
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1298	Lenovo	2 in 1 Convertible	4	1.80	33992.6400	1	1	157.350512	Intel Core i7	0	128	Intel	Windows
1299	Lenovo	2 in 1 Convertible	16	1.30	79866.7200	1	1	276.053530	Intel Core i7	0	512	Intel	Windows
1300	Lenovo	Notebook	2	1.50	12201.1200	0	0	111.935204	Other Intel Processor	0	0	Intel	Windows
1301	HP	Notebook	6	2.19	40705.9200	0	0	100.454670	Intel Core i7	1000	0	AMD	Windows
1302	Asus	Notebook	4	2.20	19660.3200	0	0	100.454670	Other Intel Processor	500	0	Intel	Windows

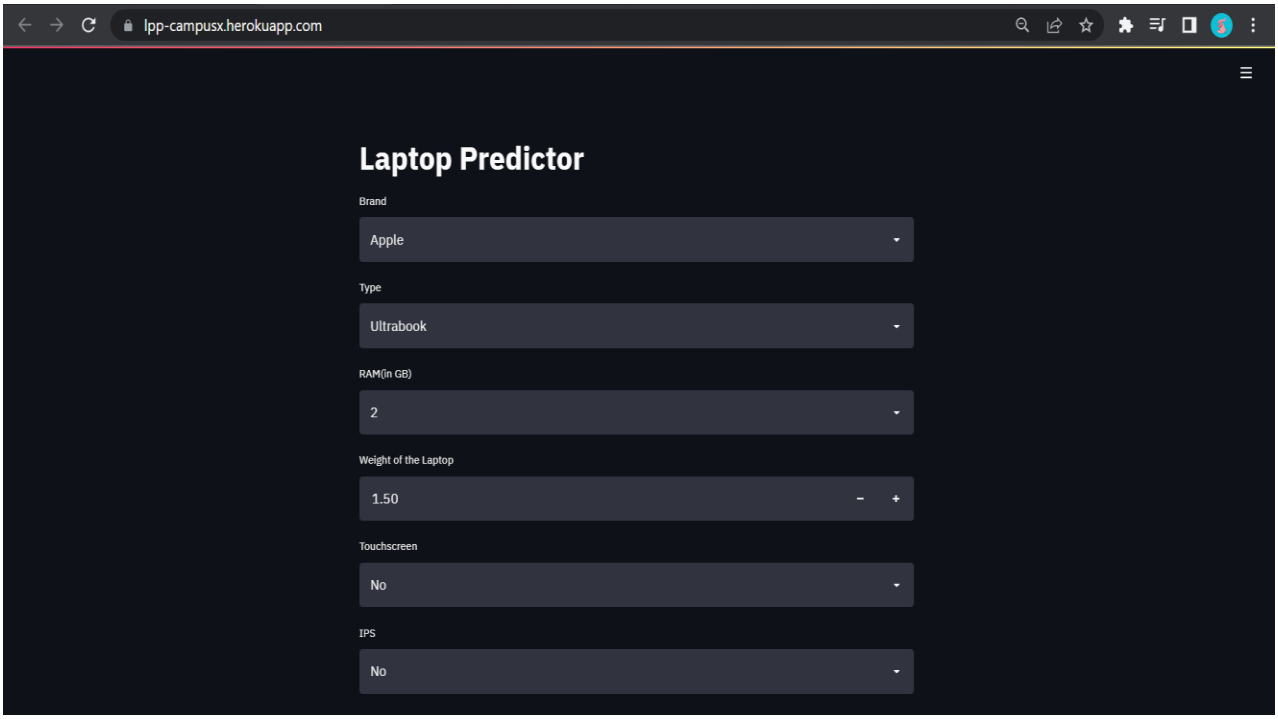
1302 rows × 13 columns

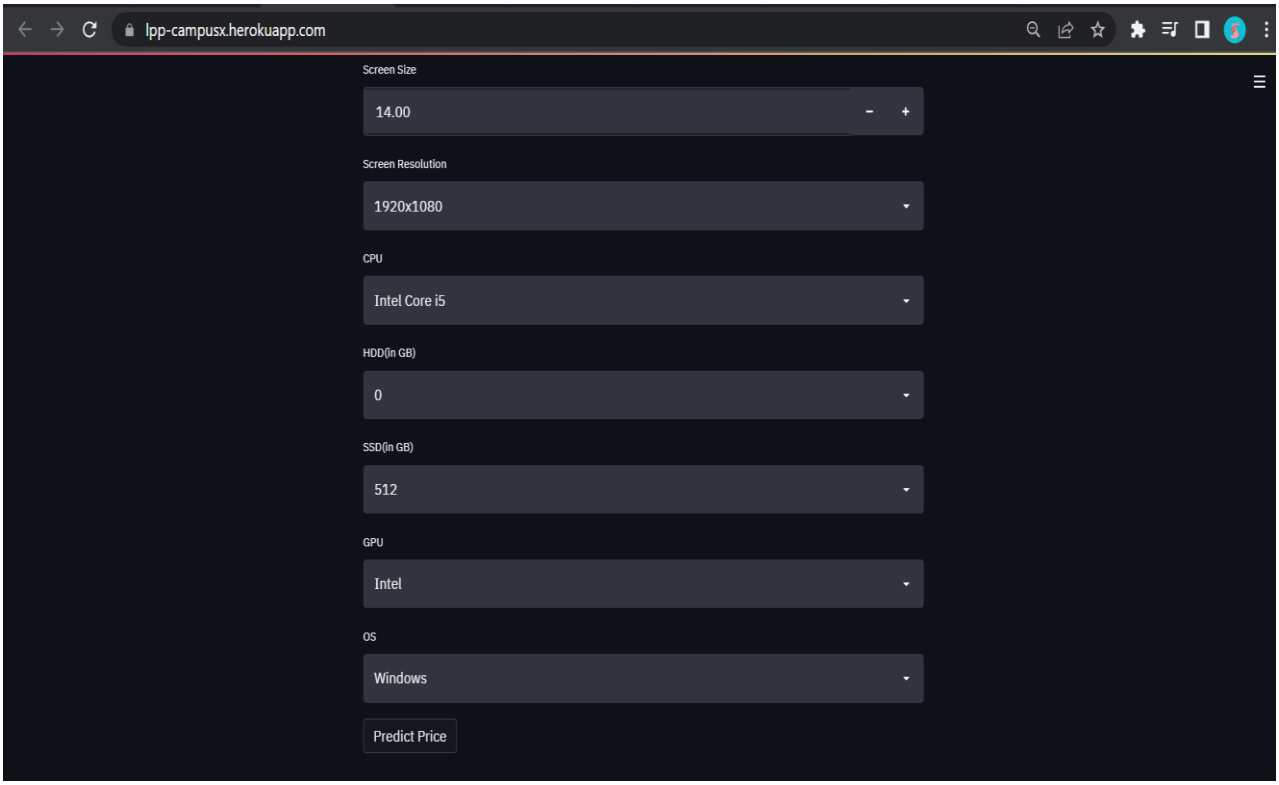
What is Streamlit?

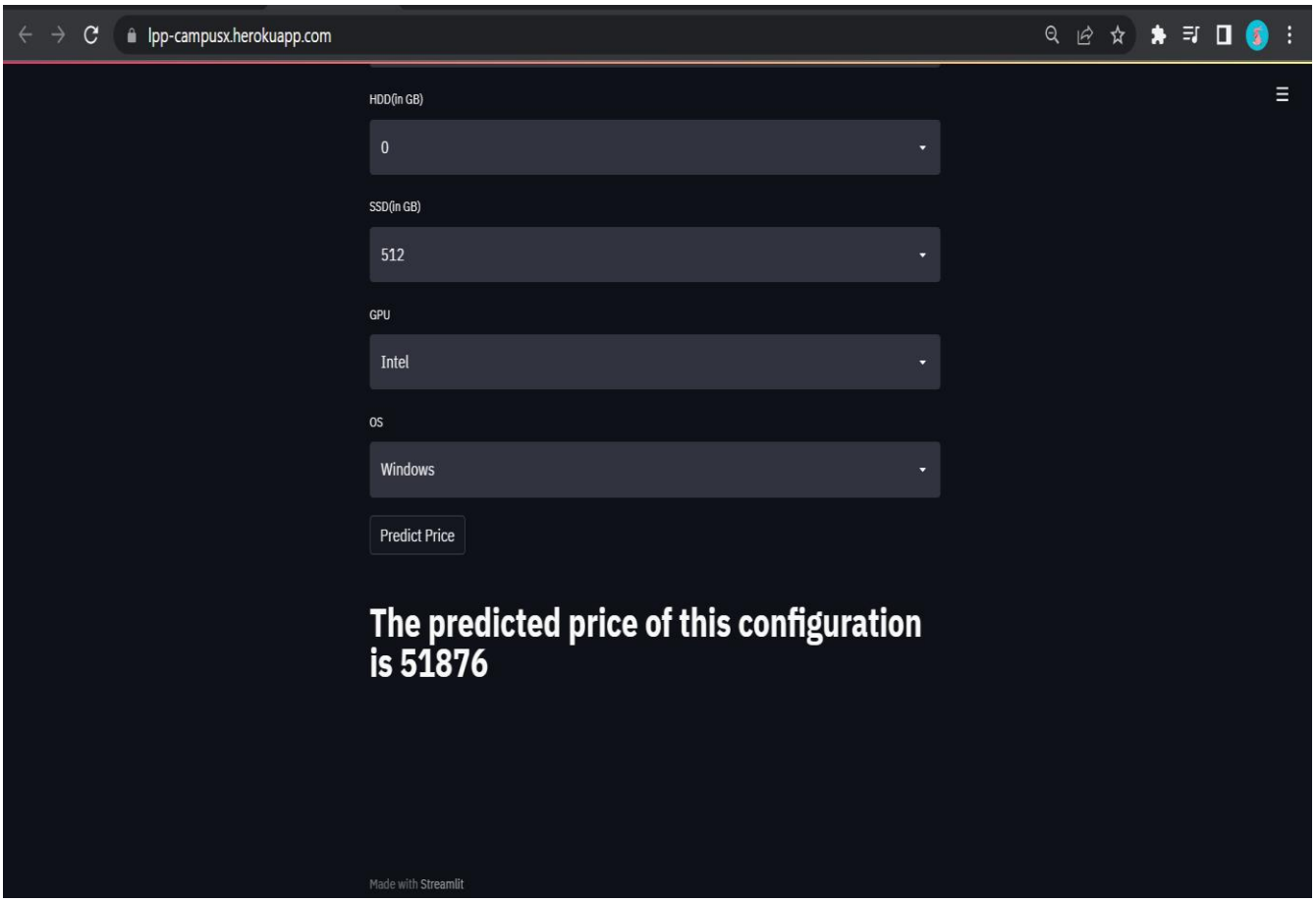
Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

The trend of Data Science and Analytics is increasing day by day. From the data science pipeline, one of the most important steps is model deployment. We have a lot of options in python for deploying our model. Some popular frameworks are Flask and Django. But the issue with using these frameworks is that we should have some knowledge of HTML, CSS, and JavaScript. Keeping these prerequisites in mind, Adrien Treuille, Thiago Teixeira, and Amanda Kelly created “Streamlit”. Now using streamlit you can deploy any machine learning model and any python project with ease and without worrying about the frontend. Streamlit is very user-friendly.

7. RESULT AND DISCUSSION






lpp-campusx.herokuapp.com
HDD(in GB)
0
SSD(in GB)
512
GPU
Intel
OS
Windows
Predict Price
The predicted price of this configuration is 51876
Made with Streamlit

8. Conclusions

Predicting something through the application of machine learning using the Decision Tree algorithm makes it easy for students, especially in determining the choice of laptop specifications that are most desirable for students to meet student needs and in accordance with the purchasing power of students. Students no longer need to look for various sources to find laptop specifications that are needed by students in meeting the needs of students, because the laptop specifications from the results of the machine learning application have provided the most desirable specifications with their prices of laptops.

9. REFRENCE

Project Learning: https://internshala.com

Report data: https://www.analyticsvidhya.com/blog/2021/11/laptop-price-prediction-practical-understanding-of-machine-learning-project-lifecycle/
https://www.kaggle.com/code/danielbethell/laptop-prices-prediction
https://www.youtube.com/watch?v=BgpM2IiCH6k
https://medium.com/analytics-vidhya/laptop-price-prediction-by-machine-learning-7e1211bb96d1