Sri Lanka Institute of Information Technology

Fundamentals of Data Mining - IT3051

Mini Project - 2022

Machine Learning Model Deployed Web Application to Predict Laptop Prices

Group 15

Group members:

Name	ID Number
Thisaranga B.V.Y.L	IT20640170
Pulasinghe T.K.	IT20641474
Perera M.L.D.	IT20661410
Masakorala W.P.	IT20660284
De Silva A.M.A.R	IT20642778

Batch: 3rd Year 1st Semester (Data Science)

Table of Contents

1. **Introduction 03**

1.1 Background and the problem definition 03

1.2 Aim and objectives 04

2. **Approach and Implementation 05**

2.1 Collecting Dataset..... 05

2.2 Data Preprocessing 06

3. **Model building 12**

3.1 Model building..... 12

3.2 Evaluating Models and choosing the best fitting model 13

4. **Front End Development 14**

5. **Technologies 23**

6. **Conclusion..... 24**

7. **Roles and Responsibilities..... 25**

1. Introduction

1.1 Background and the problem definition

Laptop price prediction especially when the laptop is coming direct from the factory to Electronic Market/Stores, is both a critical and important task. 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. we applied different methods and techniques in order to achieve higher precision of the laptop price prediction.

The problem statement is that if any user wants to buy a laptop then our application should be compatible to provide a tentative price of laptop according to the user configurations. Although it looks like a simple project or just developing a model, the dataset we have is noisy and needs lots of feature engineering, and preprocessing that will drive the interest in developing this project. Most of the columns in a dataset are noisy and contain lots of information. But with feature engineering will get more good results. we will obtain a good accuracy over dataset. we will develop a web-based application that could predict a tentative price of a laptop based on user configuration.

This selected data set includes the data of different types of laptop models. By using this data set, we planned to build a model that can be used to predict laptop prices over different kinds of features. And the laptop sellers can deploy this model on their websites in order to give reliable service for their customers. So, it will help the customers of the laptop store, to plan their budget for the laptop they want, before going to the market without the help of a sales representative.

1.2 Aim and Objectives

- The major objective of this project is to develop a machine learning model to accurately predict laptop prices and develop a web application to utilize the prediction model.

- Deploying the prediction model into a web application to make it easier even for users without and statistical or data science knowledge.

- Benefit the laptop price prediction system designing and operations by allowing them to predict essential measurements.

2. Approach and Implementation

2.1 Collecting the dataset

About the dataset:

This dataset contains different features and specifications of the laptops and it contains 12 columns
and 1303 records.

The fields:

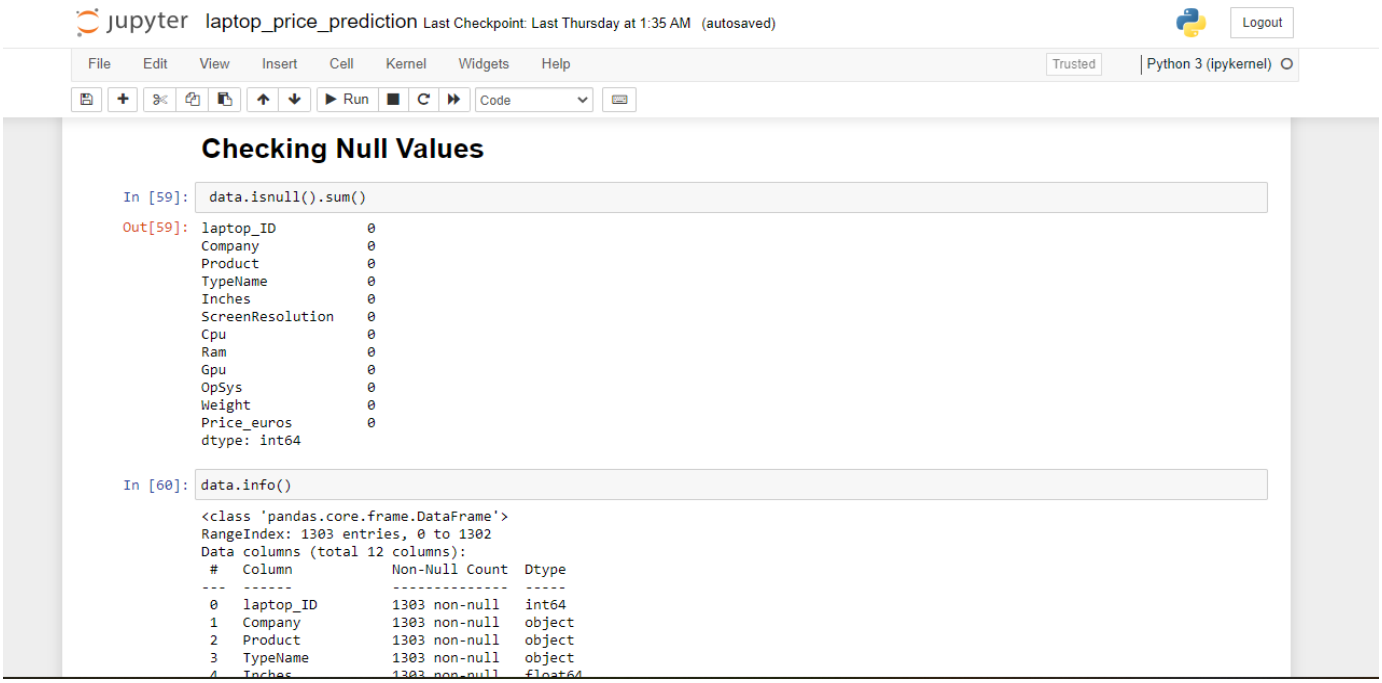
Column name	Data type	Description
Laptop_ID	Int64	Laptop ID
Company	Object	Laptop Manufacturer
Product	Object	Brand and Model
TypeName	Object	Laptop Type (Notebook, Ultrabook, gaming, etc.)
Inches	Float64	Screen Size of Laptop
ScreenResolution	Object	Screen Resolution
Cpu	Object	Central Processing Unit
Ram	Object	Laptop RAM size
Gpu	Object	Graphics Processing Units
OpSys	Object	Operating System (windows, mac, Linux, etc.)
Weight	Object	Laptop Weight
Price_euros	Float64	Laptop Price (Euro)

2.2 Data Pre-processing

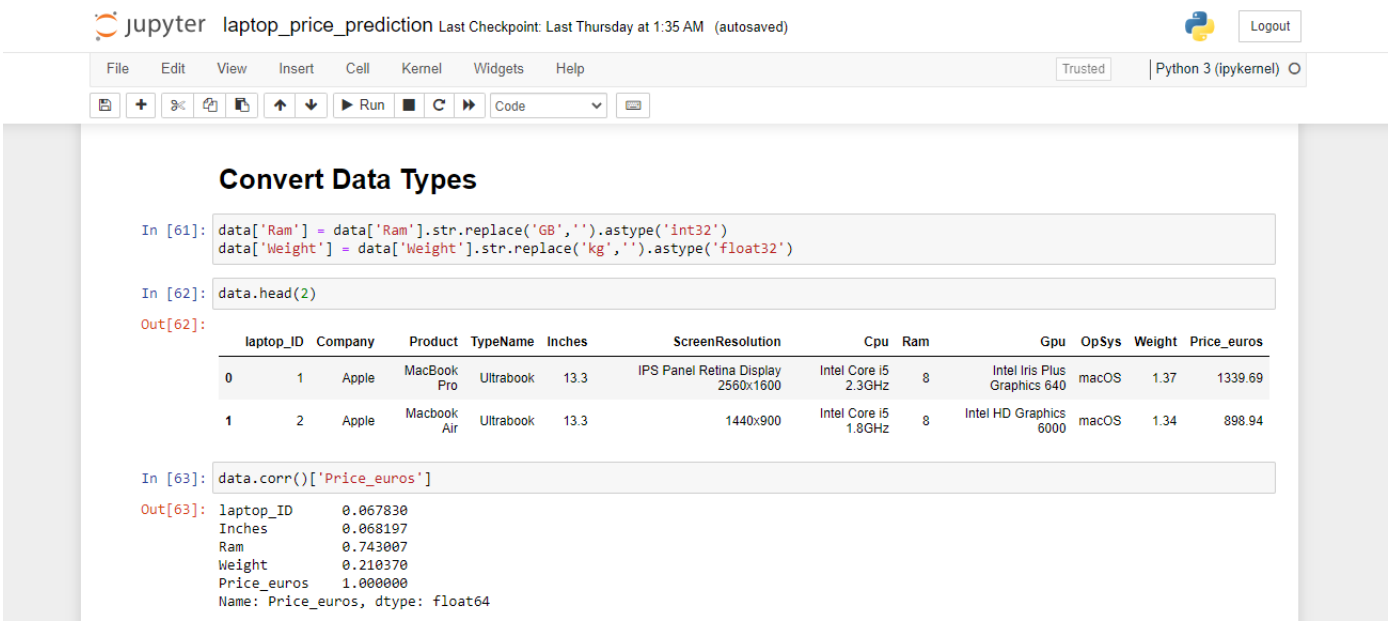
Data pre-processing is important to increase the learning capability and the accuracy of the machine learning model.

The following methods were used to pre-process the data.

1. Checking and removing null values



2. Convert Data Types



3. Update Columns according data types

jupyter laptop_price_prediction Last Checkpoint: Last Thursday at 1:35 AM (autosaved)

Logout

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

Run Code

Update columns according data types

In [65]:
```
def add_company(inpt):
    if inpt == 'Samsung' or inpt == 'Razer' or inpt == 'Mediacom' or inpt == 'Microsoft'or inpt == 'Xiaomi'or inpt == 'Vero'or ir
        return 'Other'
    else:
        return inpt
```

In [66]:
```
data['Company'] = data['Company'].apply(add_company)
```

In [67]:
```
data['Company'].value_counts()
```

Out[67]:
```
Dell       297
Lenovo     297
HP         274
Asus       158
Acer       103
MSI         54
Other       51
Toshiba     48
Apple       21
Name: Company, dtype: int64
```

In [68]:
```
len(data['Product'].value_counts())
```

Out[68]: 618

jupyter laptop_price_prediction Last Checkpoint: Last Thursday at 1:35 AM (autosaved)

Logout

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

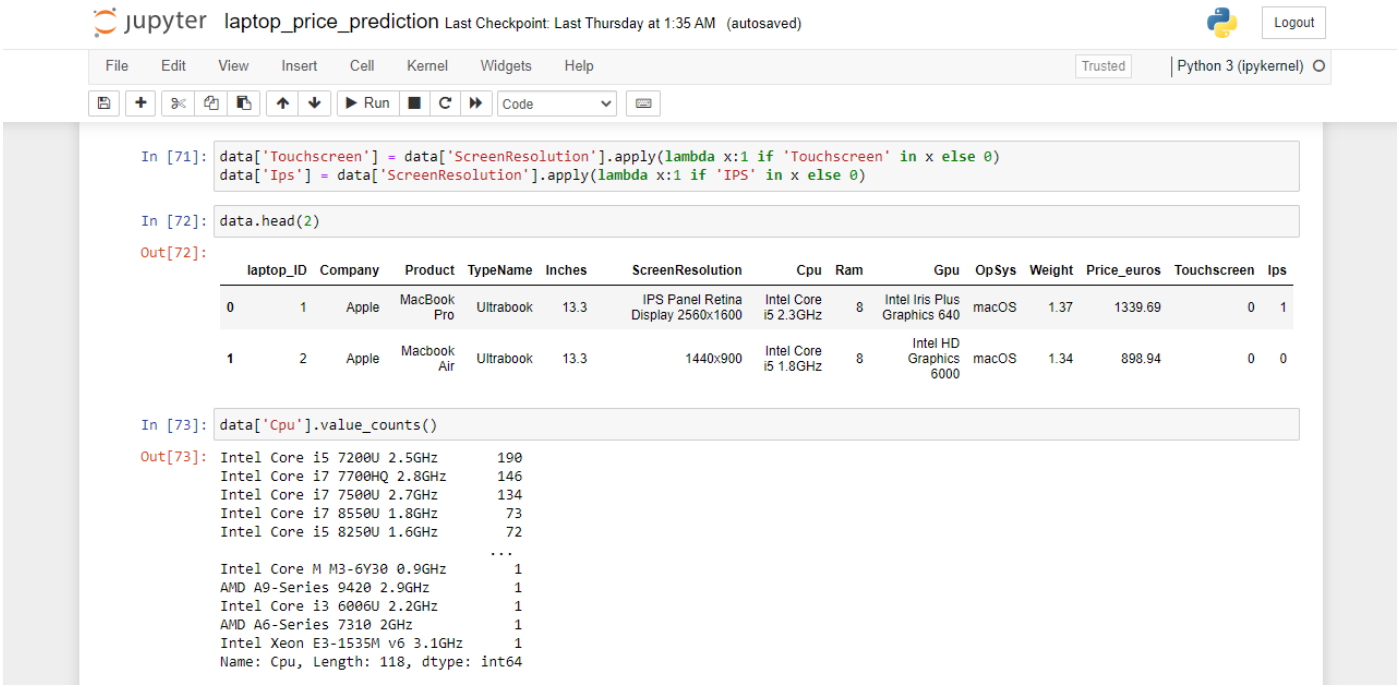
Run Code

In [69]:
```
data['TypeName'].value_counts()
```

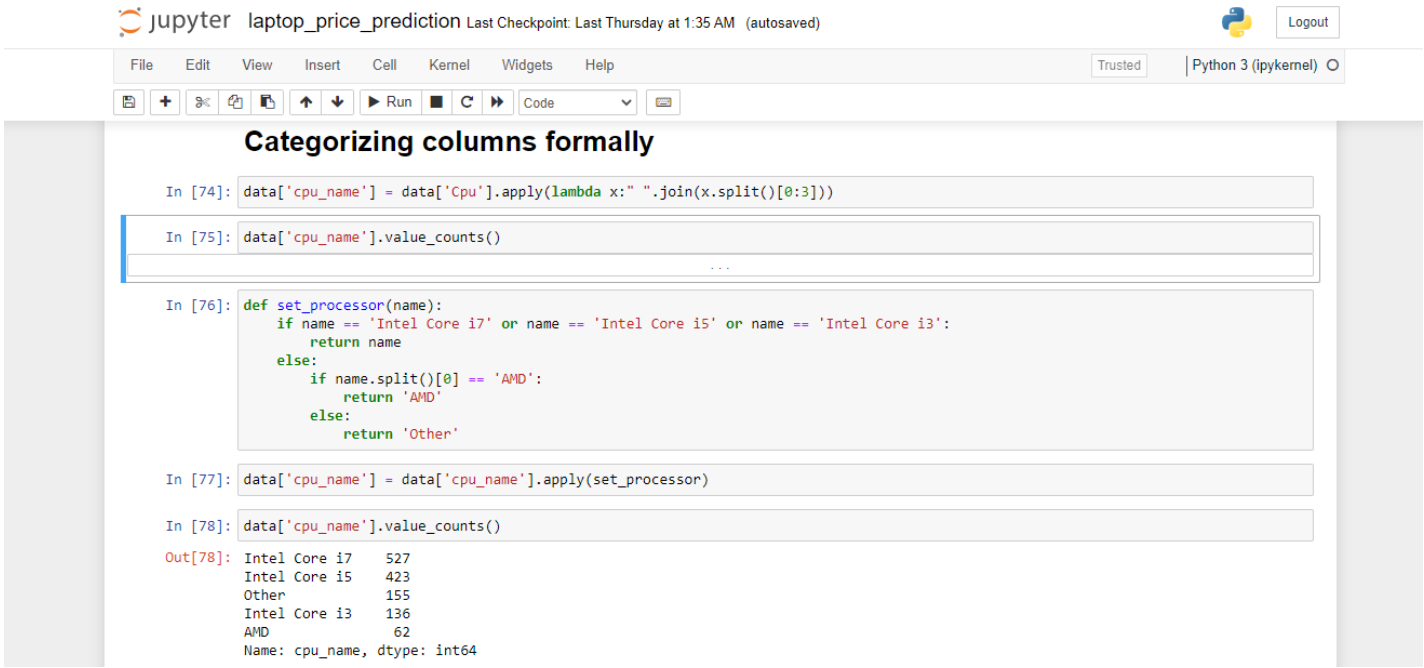
Out[69]:
```
Notebook              727
Gaming                205
Ultrabook             196
2 in 1 Convertible    121
Workstation            29
Netbook                25
Name: TypeName, dtype: int64
```

In [70]:
```
data['ScreenResolution'].value_counts()
```

Out[70]:
```
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
```



4. Categorizing Columns formally



jupyter laptop_price_prediction Last Checkpoint: Last Thursday at 1:35 AM (autosaved)

Logout

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

Run Code

In [81]:
```
data['gpu_name'] = data['Gpu'].apply(lambda x:" ".join(x.split()[0:1]))
```

In [82]:
```
data['gpu_name'].value_counts()
```

Out[82]:
```
Intel     722
Nvidia    400
AMD       180
ARM         1
Name: gpu_name, dtype: int64
```

In [83]:
```
data.shape
```

Out[83]:
```
(1303, 16)
```

In [84]:
```
data = data[data['gpu_name'] != 'ARM']
```

In [85]:
```
data.shape
```

Out[85]:
```
(1302, 16)
```

In [86]:
```
data.head(2)
```

Out[86]:

	laptop_ID	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Gpu	OpSys	Weight	Price_euros	Touchscreen	Ips	cpu_name	gp
0	1	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	Intel Iris Plus Graphics 640	macOS	1.37	1339.69	0	1	Intel Core i5	

jupyter laptop_price_prediction Last Checkpoint: Last Thursday at 1:35 AM (autosaved)

Logout

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

Run Code

In [88]:
```
def set_os(inpt):
    if inpt == 'Windows 10' or inpt == 'Windows 7' or inpt == 'Windows 10 S':
        return 'Windows'
    elif inpt == 'macOS' or inpt == 'Mac OS X':
        return 'Mac'
    elif inpt == 'Linux':
        return inpt
    else:
        return 'Other'
```

In [89]:
```
data['OpSys'] = data['OpSys'].apply(set_os)
```

In [90]:
```
data['OpSys'].value_counts()
```

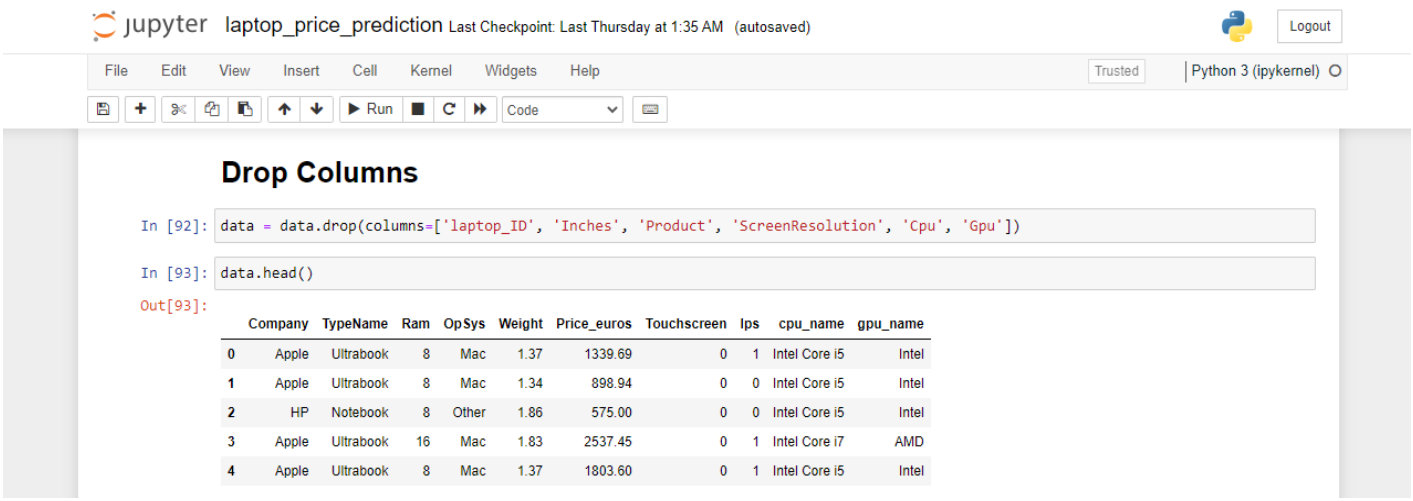
Out[90]:
```
Windows    1125
Other        94
Linux        62
Mac          21
Name: OpSys, dtype: int64
```

In [91]:
```
data.head(2)
```

Out[91]:

	laptop_ID	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Gpu	OpSys	Weight	Price_euros	Touchscreen	Ips	cpu_name	gp
0	1	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	Intel Iris Plus Graphics 640	Mac	1.37	1339.69	0	1	Intel Core i5	
1	2	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	Intel HD Graphics 6000	Mac	1.34	898.94	0	0	Intel Core i5	

5. Drop Unwanted Columns



Drop Columns

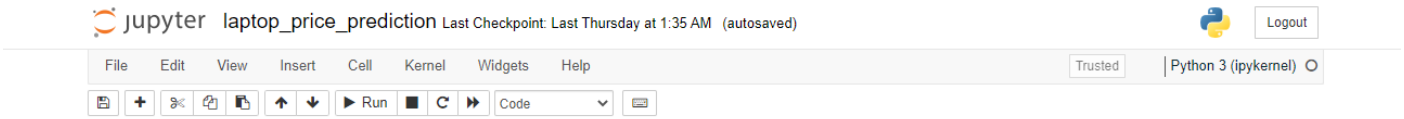
```
In [92]: data = data.drop(columns=['laptop_ID', 'Inches', 'Product', 'ScreenResolution', 'Cpu', 'Gpu'])
```

```
In [93]: data.head()
```

Out[93]:

	Company	TypeName	Ram	OpSys	Weight	Price_euros	Touchscreen	Ips	cpu_name	gpu_name
0	Apple	Ultrabook	8	Mac	1.37	1339.69	0	1	Intel Core i5	Intel
1	Apple	Ultrabook	8	Mac	1.34	898.94	0	0	Intel Core i5	Intel
2	HP	Notebook	8	Other	1.86	575.00	0	0	Intel Core i5	Intel
3	Apple	Ultrabook	16	Mac	1.83	2537.45	0	1	Intel Core i7	AMD
4	Apple	Ultrabook	8	Mac	1.37	1803.60	0	1	Intel Core i5	Intel

6. Split the data into training and testing sets.



Split Data Set into training and testing set

```
In [97]: X = data.drop('Price_euros', axis=1)
         y = data['Price_euros']
```

```
In [98]: !pip install sklearn
```

```
Requirement already satisfied: sklearn in c:\users\parame\anaconda3\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\parame\anaconda3\lib\site-packages (from sklearn) (1.0.2)
Requirement already satisfied: numpy>=1.14.6 in c:\users\parame\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.21.
5)
Requirement already satisfied: joblib>=0.11 in c:\users\parame\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\parame\anaconda3\lib\site-packages (from scikit-learn->sklearn)
(2.2.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\parame\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.9.1)
```

```
In [99]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
```

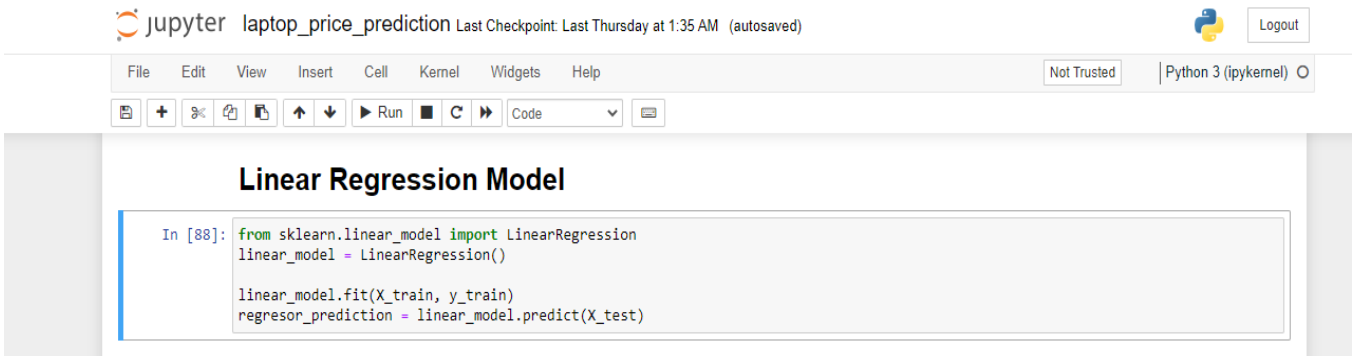
```
In [100]: X_train.shape, X_test.shape
```

```
Out[100]: ((976, 31), (326, 31))
```

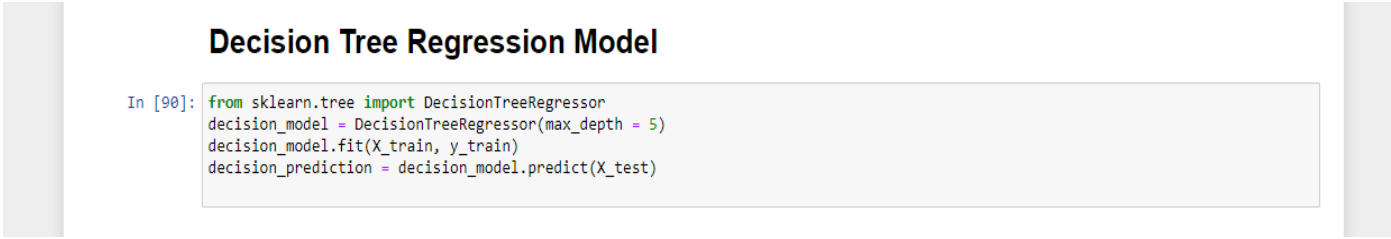
3. Model Building

3.1 Model Building in Python

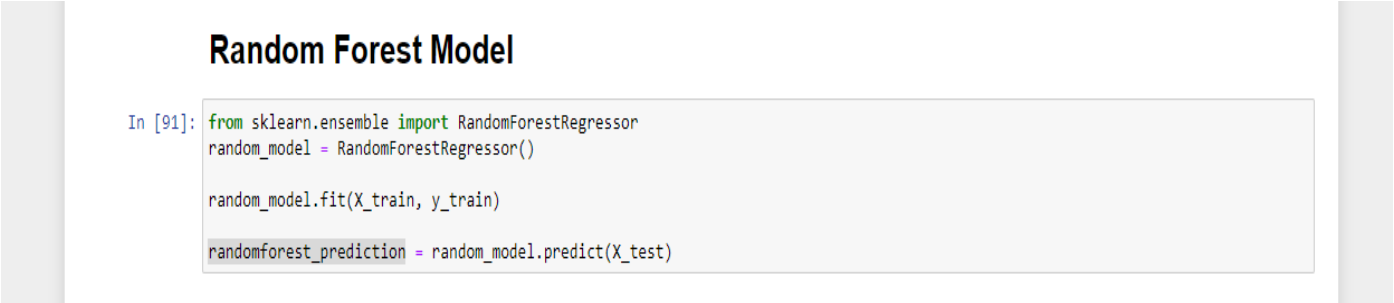
1. Linear Regression Model



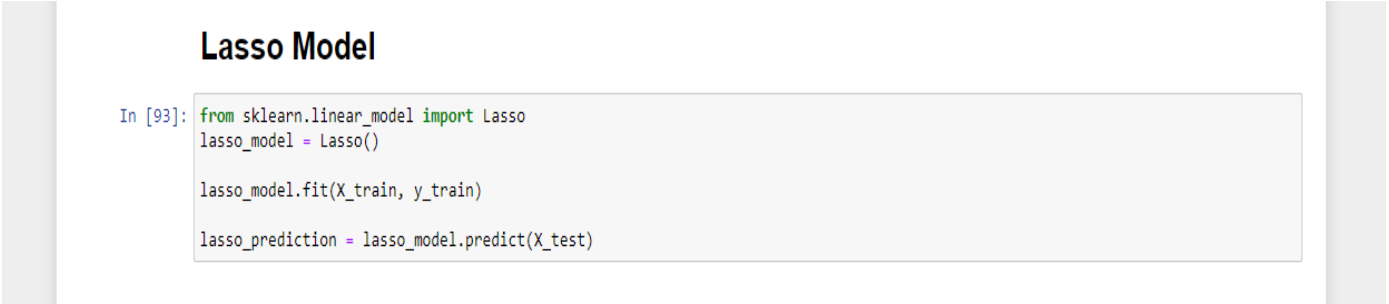
2. Decision Tree Regression Model



3. Random Forest Model



4. Lasso Model



3.2 Evaluating Models and choosing the best fitting model

The models were evaluated using R squared valued to choose the best prediction model. As the result the **Random Decision Forest** is chosen as the best model with **0.75 of accuracy rate.**

- Considering R2 Regression Score

R^2 Regression Score

In [94]:
```
from sklearn.metrics import r2_score

linear_R2 = r2_score(y_test, regresor_prediction)
decision_R2 = r2_score(y_test, decision_prediction)
random_R2 = r2_score(y_test, randomforest_prediction)
lasso_R2 = r2_score(y_test, lasso_prediction)

print('R2 - Score of Linear Regression model is : ', linear_R2)
print('R2 - Score of Dicision Tree Regression model is : ', decision_R2)
print('R2 - Score of Random Forest Regression model is : ', random_R2)
print('R2 - Score of Lasso Model is : ', lasso_R2)
```

```
R2 - Score of Linear Regression model is :  0.7144376547326752
R2 - Score of Dicision Tree Regression model is :  0.6735947597463422
R2 - Score of Random Forest Regression model is :  0.7549981593984256
R2 - Score of Lasso Model is :  0.7150659827523432
```

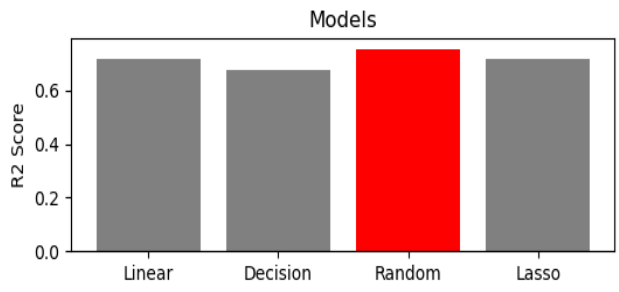
- Comparing models to find the best solution

Compare Models

In [95]:
```
models = ('Linear','Decision','Random','Lasso')
n = [linear_R2,decision_R2,random_R2,lasso_R2]
y_pos = np.arange(len(models))
highlights = ['gray' if (x< max(n)) else 'red' for x in n]
```

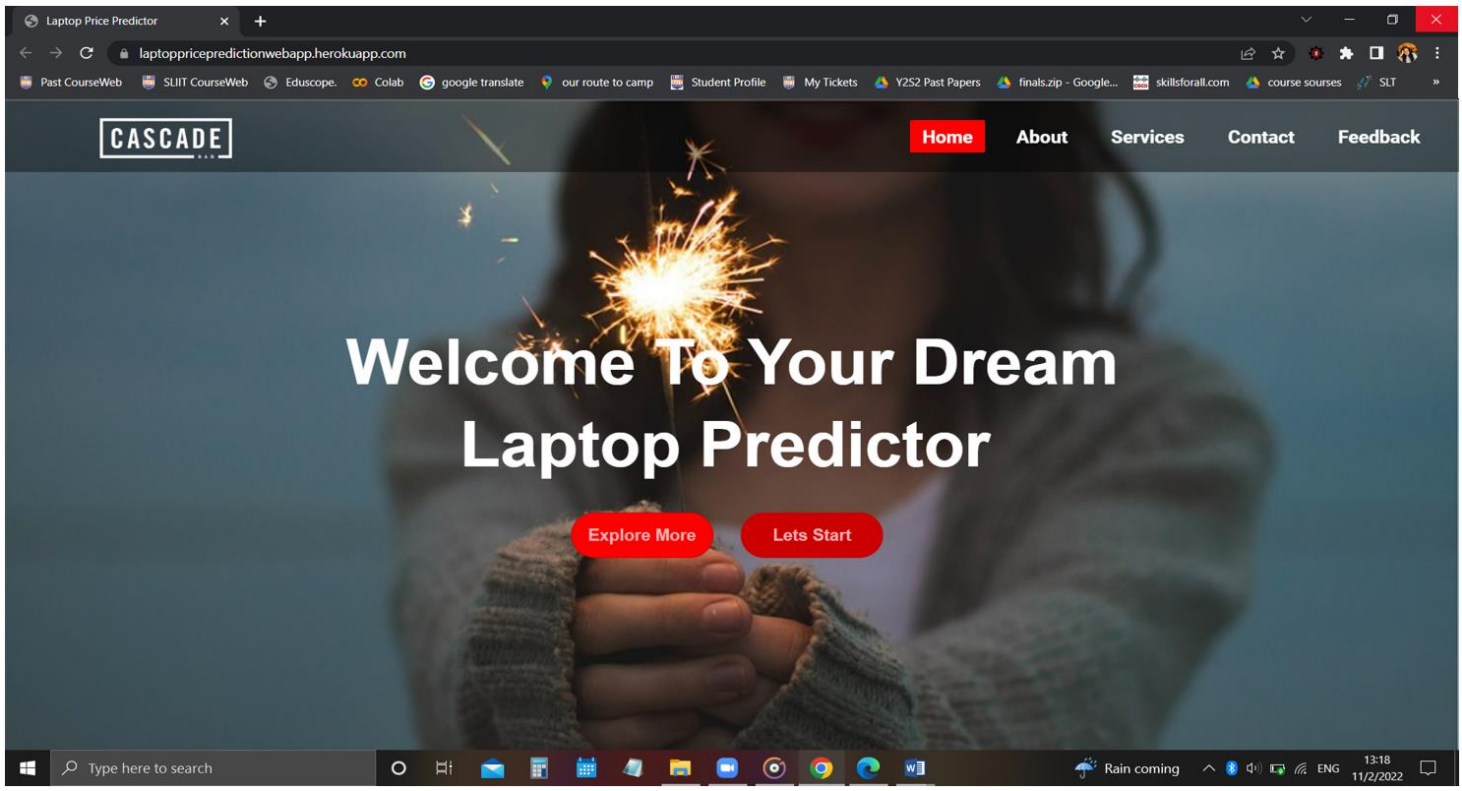
In [101]:
```
from matplotlib import pyplot as plt

plt.figure(figsize=(6,2))
plt.xticks(y_pos,models)
plt.ylabel('R2 Score')
plt.title('Models')
plt.bar(y_pos, n, color = highlights)
```

Out[101]: <BarContainer object of 4 artists>



4.Front End Development

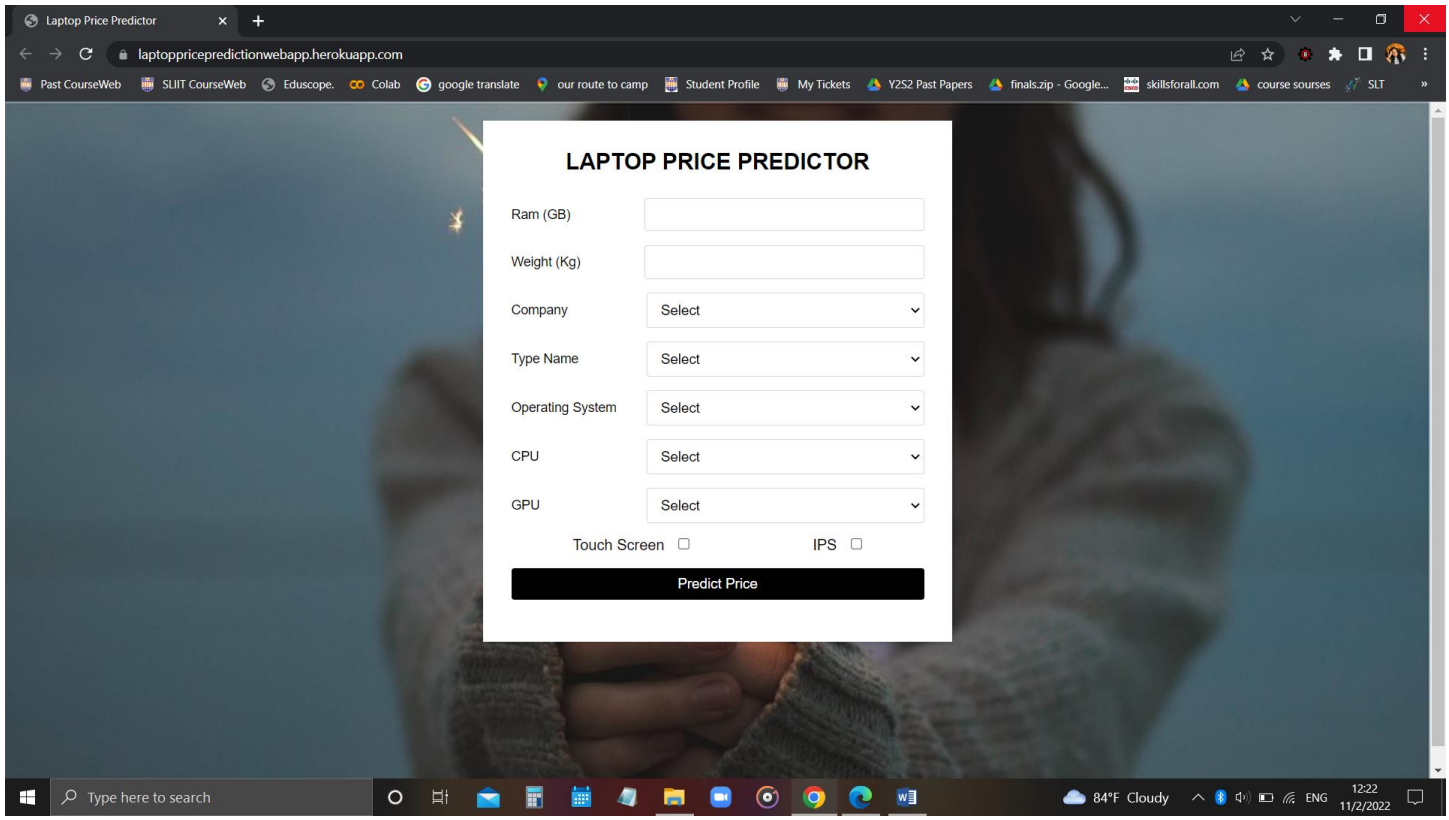
First Page



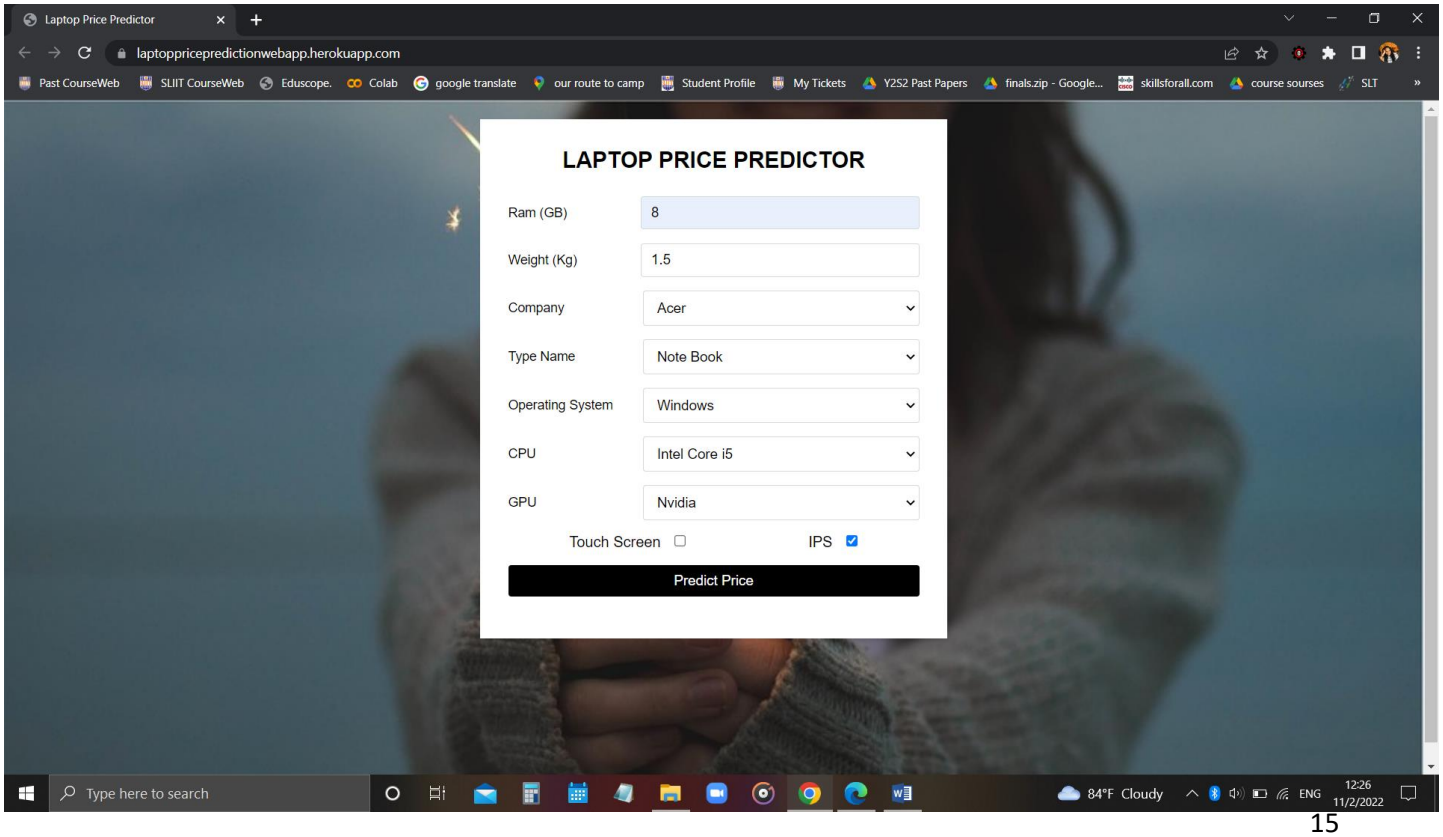
Second Page

The dashboard allows the user to input previously unknown data as

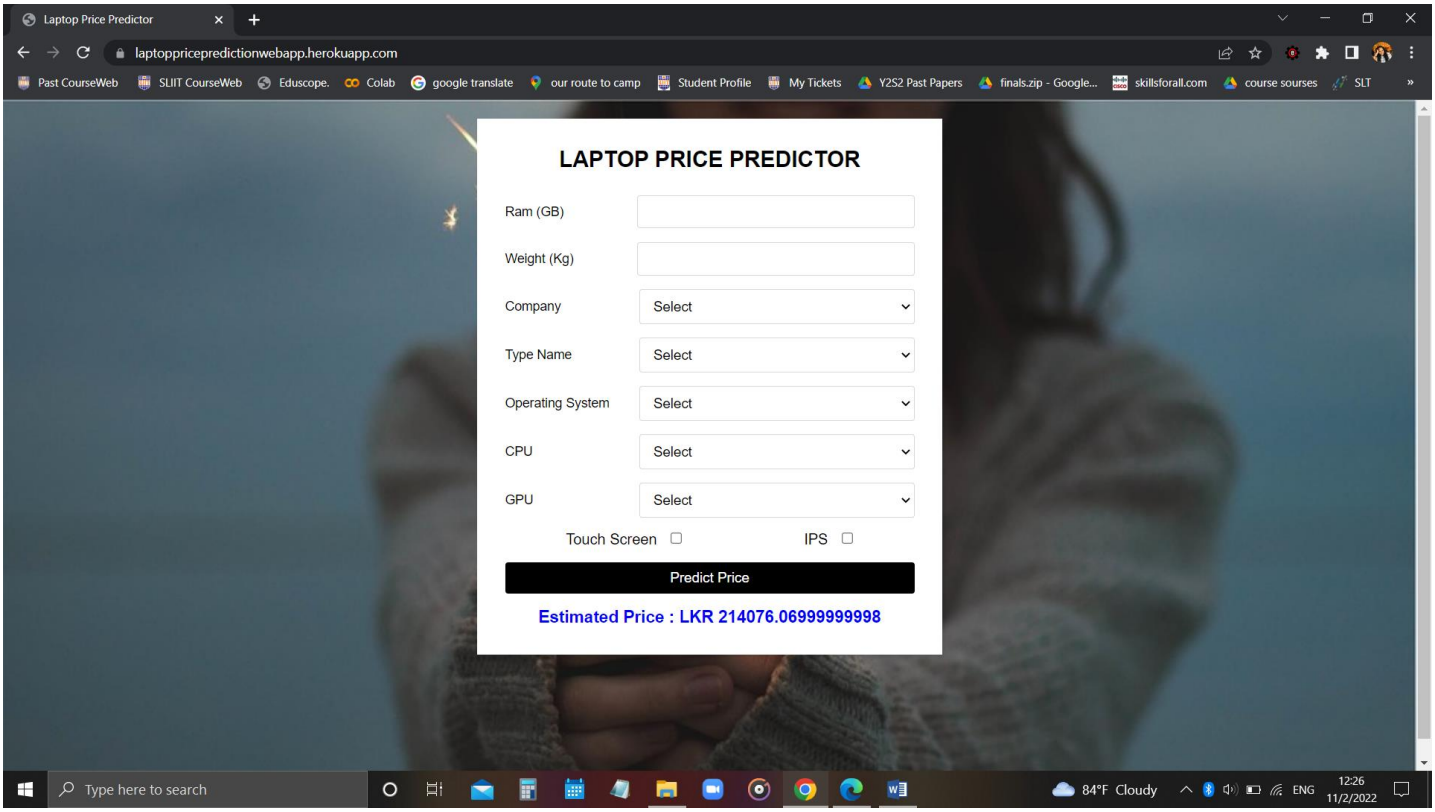
RAM(GB)
Weight (Kg),
Company
Type Name
Operating System
CPU
GPU
Touch Screen (IPS or not) and gives the predicted laptop price as the output.



Add values to the input fields



Output


Laptop Price Predictor
laptoppricepredictionwebapp.herokuapp.com
LAPTOP PRICE PREDICTOR
Ram (GB)
Weight (Kg)
Company
Select
Type Name
Select
Operating System
Select
CPU
Select
GPU
Select
Touch Screen
IPS
Predict Price
Estimated Price : LKR 214076.06999999998
Type here to search
84°F Cloudy
ENG
12:26
11/2/2022

5. Technologies

Language – Python
IDE – Jupyter Notebook, VS Code
Front End – HTML, CSS
Back End - Flask
Deployment Server – Heroku

- To implement this machine learning project Python language was used, that offers concise and readable code which makes it easier to understand complex algorithm codes.
- As the IDES Jupyter Notebook and VS Code were used.
- Jupyter Notebook was used implement data pre- processing methods as it is an open-source web application that allows us to create and share codes.
- Working as a group, was made easy by using Jupyter Notebook.
- VS Code was used to develop the front-end dashboard using Python.
- VS Code was useful to manage our version controlling with Github.
- HTML, CSS used to develop front end of the application.
- Flask which is an open-source Python library was used to develop the backend of the application.
- Heroku is the deployment environment and it is a container-based cloud Platform as a Service (PaaS).

6. Conclusion

Today, using Machine Learning techniques can provide a proper estimate for the Laptop Price prediction. Since the above data set consists of many complex factors related to the laptop price, with nonlinear characteristics, using traditional data modeling techniques would not lead to appropriate results. However, Machine Learning models can handle such complexities easily.

Since, Machine Learning models have various advantages over other types of models, they can model nonlinearities in data more accurately. Due to this, various machine learning models can be used for laptop price prediction. Linear Regression Model, Random Forest Regression Model, Lasso Regression Model and Decision Tree Regression Model are used here for the predictions. The Random Forest Regression model is a rarely used model, and also in this situation, it showed good results. Considering the data set, the prediction results shown from the Random Forest Regression Model stands as the best proof of concept. This model provides more accurate results under periods of higher variations.

7. Roles and responsibilities

Student Number	Student Name	Responsibilities
IT20640170	Thisaranga B.V.Y.L.	• Data Preprocessing • Model Building • Building front-end for Models
IT20641474	Pulasinghe T.K.	• Data Preprocessing • Model Building • Building front-end for Models
IT20661410	Perera M.L.D.	• Data Preprocessing • Model Building • Building front-end for Models
IT20660284	Masakorala W.P.	• Data Preprocessing • Model Building • Building front-end for Models
IT20642778	De Silva A.M.A.R.	• Data Preprocessing • Model Building • Building front-end for Models