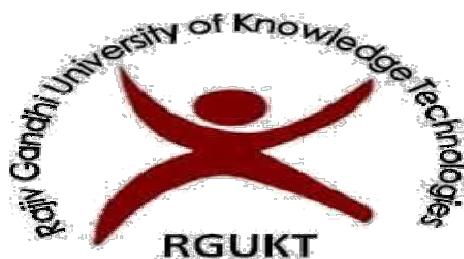


A Project Report
On
LAPTOP PRICE PREDICTION
USING
SUPERVISED LEARNING

Bachelor of Technology
in
Computer Science and Engineering



Rajiv Gandhi University of Knowledge
and Technologies
RK Valley Kadapa

Submitted By
A G DHARANI – R170672

Under the Esteemed Guidance of

Mr.Satyanandaram N
RGUKT RK Valley

DECLARATION

We hereby declare that the report of the B.Tech Major Project Work entitled "LAPTOP PRICE PREDICTION USING SUPERVISED LEARNING" which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for award of any degree.

A G DHARANI
R170672

Rajiv Gandhi University Of Knowledge Technologies



CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled “**LAPTOP PRICE PREDICTION USING SUPERVISED LEARNING**” submitted by A G DHARANI (R170672) under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic semester- - 2 2022-2023 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Project Internal Guide

Mr.N.SatyaNandaram

Assistant Professor

Head of the Department

Mr.N.SatyaNandaram

Assistant Professor

INDEX

1. Abstract.....	5
2. Existing System.....	6
3. Proposed System.....	6
4. SRS Document.....	7
4.1 Purpose.....	7
4.2 Intended Audience.....	7
4.3 Product Vision.....	7
4.4 Technologies.....	8
5. Algorithm Used.....	9 -10
6. Requirement Specification.....	11
7. Use Case Diagram.....	12
8. Exploratory Data Analysis.....	13-17
9. Code.....	18-21
10. Deployment Output.....	21
11. Conclusions.....	22
12. References.....	23

1.ABSTRACT

The design and development of laptop, computers work environment of managers, decision makers, engineers, scientists, financial analysts, office workers and even students is no more limited to their offices, research labs field or class rooms. Their work space has expanded to unlimited dimensions. However the type of problems being handled by these individual's have also become far more complex. Business needs to be focused on global competitiveness, engineering is taking its own shapes in a variety of fields, and students are designing their own course plans.

Present Project is focused on using **Supervised learning techniques in machine learning** to predict a suitable laptop out of a huge range of manufacturers and brands available worldwide with combinations of various specifications. Major components considered for selecting a laptop are **hard drive space, RAM, processing speed** and price of various brands. A different laptop brands specifications data was taken. A number of statistical tools were applied to provide the selection methodology of most suitable brand by creating an application using Streamlit.

2.EXISTING SYSTEM

In the existing system there are so many problems for laptop price prediction problem for one of the Kaggle competitions, in which they combine standard machine learning algorithms with their original ideas like residual algorithms, logit transform and neural network machine. But during data analysis show that the laptop price variation prediction results is not accurate enough.

Disadvantages

Accuracy is low

Sometimes standard deviation of the results is very high because of small datasize

3. PROPOSED SYSTEM

This project proposes a method Use of computers and laptops has been on a rise for the past few Machine learning is a branch of Artificial intelligence that deals with implementing applications that can make a future prediction based on past data. If you are a data science enthusiast or practitioner, then this article will help build your own end-to-end machine learning project from scratch. There are various steps involved in building a machine learning project but not all the steps are mandatory to use in a single project, and it all depends on the dataset. Obtaining Laptop prices dataset. Random Forest Algorithm is used. It gives more accurate results when compared with other algorithms.

Advantages

The greater number of trees in forest leads to high accuracy and prevents the problem of overfitting.

Can implement in all systems.

SRS Document

4.Introduction:

This document has the requirements of Laptop Price Prediction using Supervised Learning project. This project is used to predict the price of laptops based on the selected attributes.

4.1. Purpose:

The purpose of this document is to gather the requirements that are needed for implementing the Laptop Price prediction. It also focuses on various key features, the product, product vision and scope, product overview. The main purpose of laptop price prediction used to predict the price of laptop according to user requirements

4.2. Intended Audience:

The intended audience will be the user who can provide dataset (the dataset contains specifications of a laptop as attributes).

Users:

1. User

- Can be a business people
- Can be a student or working professionals.

4.3 Product Vision

The product vision is to implement a prediction app to predict the price of a laptop according to user specifications. By using streamlit we can display the price of the laptop according to user need.

4.4 Technologies

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-

giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc. The biggest strength of Python is huge collection of standard library which can be used for the following.

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

2. Machine Learning

ML Tools used:

Numpy:

NumPy is a library for the Python Programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas:

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Matplotlib:

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Scikit Learn:

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Seaborn:

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

3.Streamlit

Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science apps.it is used to deploy an app without any programming languages like html, css, javascript.

5.Algorithm Used to design the Laptop price prediction System:-

RANDOM FOREST ALGORITHM:

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The Working process can be explained in the below steps

Step-1: Select random K data points from the training set.

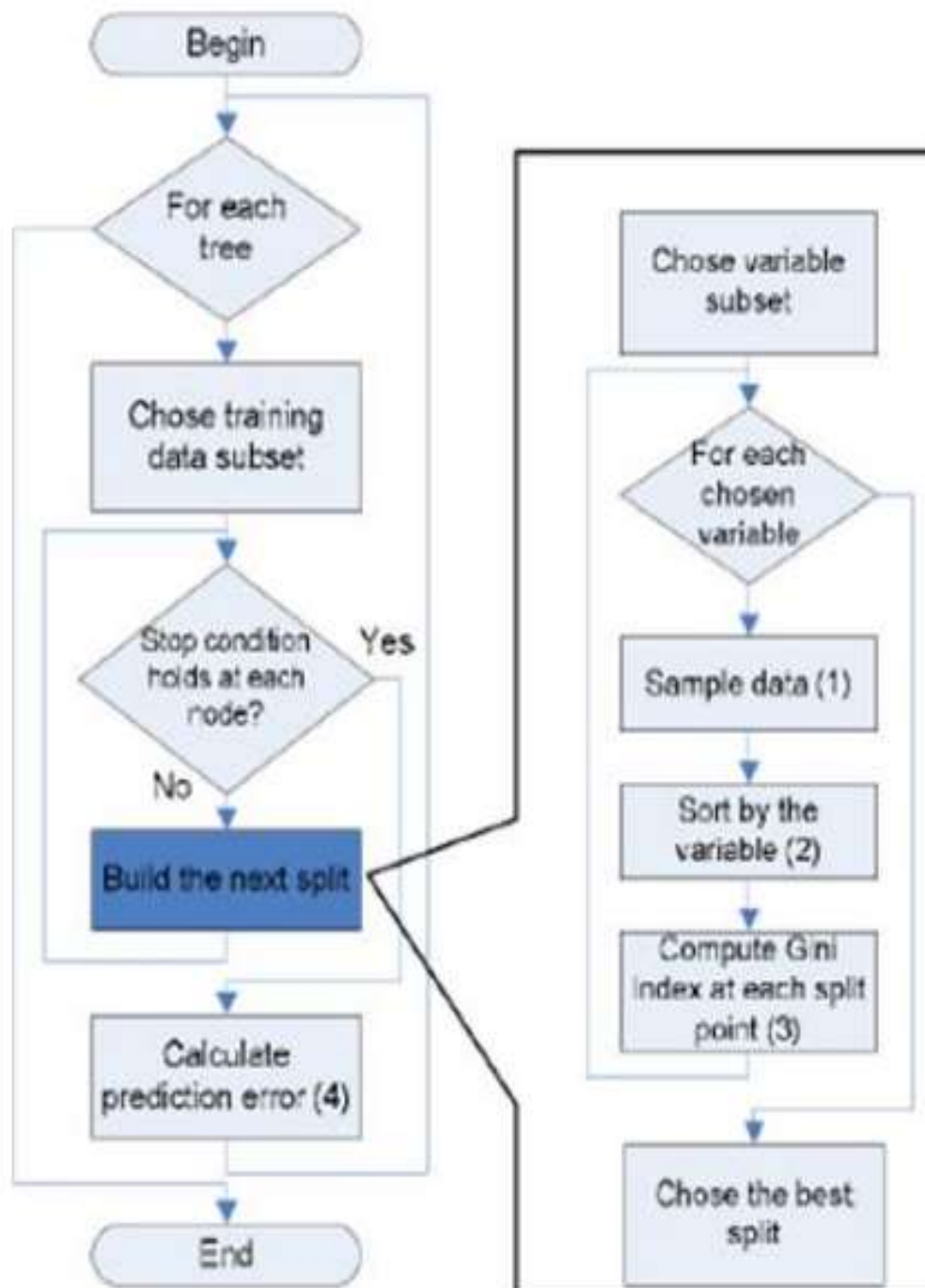
Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

6. Generating Model



7.Requirement Specification:

Hardware Configuration:

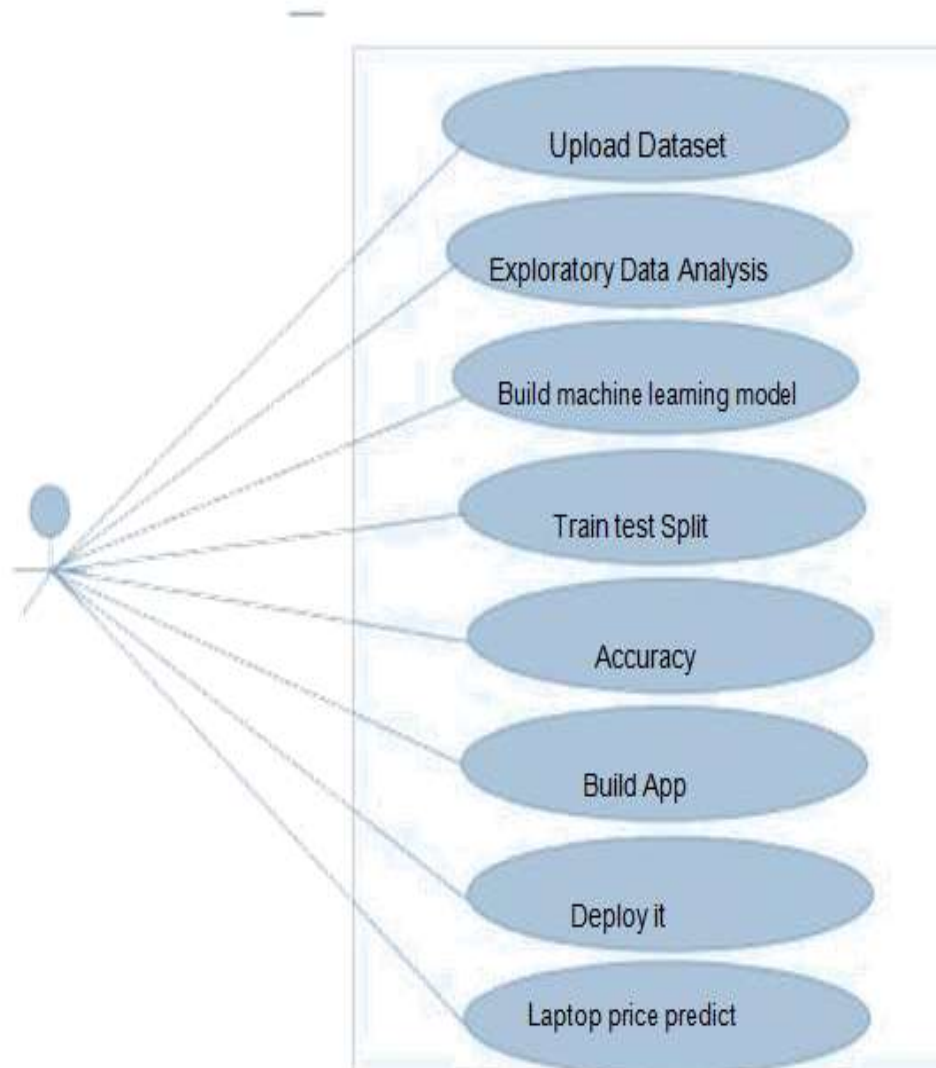
- i)Ram At least 4 GB
- ii)Hard disk 10 GB
- iii)Processor Intel i3 or more 2.00FHZ

Software Requirement:

- i)Front end Streamlit
- ii)Web Browser Operating System Firefox , Google Chrome or any compatible browser
- iii)Ubuntu,Windows or any equivalent OS

8.Use Case Diagram:

LAPTOP PRICE PREDICTION



DATA COLLECTION

The Data set used in this project is downloaded from **Kaggle website**. Which is a free source of Data sets for machine Learning and Data Science.

It is a reliable source,so we took data from Kaggle .The step of gathering data is the foundation of the machine learning process

Dataset: laptop_data.csv"

Some of the attributes are Company name, Typename, screen size, Screen Resolution, CPU, RAM, Memory, GPU, OpSys, Weight, After downloading it to the PC can be directly upload to the Jupyter Notebook

```
df = pd.read_csv('laptop_data.csv')
```

Import Libraries

Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns

Exploratory Data Analysis

```
df= pd.read_csv('laptop_data.csv')
```

```
In [4]: df.head(5)
```

Out[4]:

	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

df.shape

(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
```

```
df['Ram']=df['Ram'].str.replace('GB','')
```

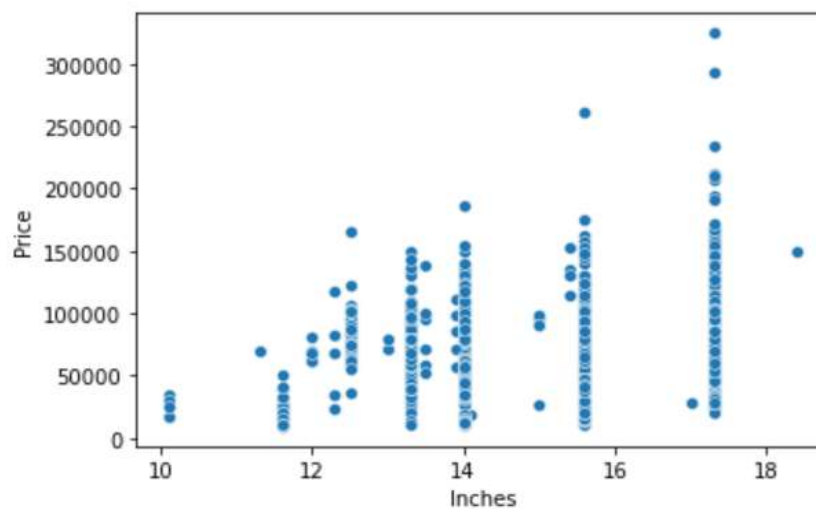
```
df['Weight']=df['Weight'].str.replace('kg','')
```

```
df.head()
```

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

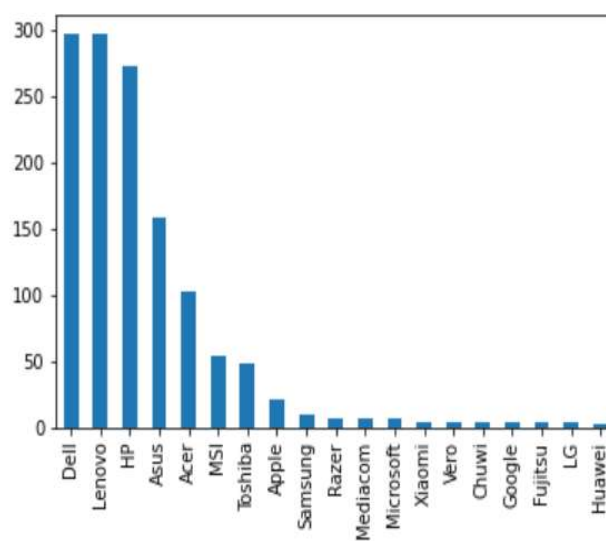
```
Out[23]: <AxesSubplot:xlabel='Inches', ylabel='Price'>
```



Price

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

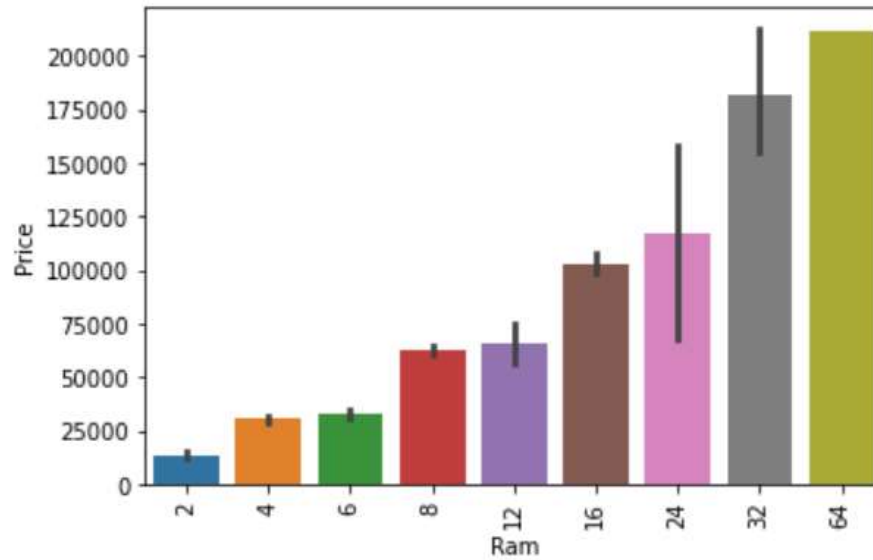
```
Out[18]: <AxesSubplot:>
```



```

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

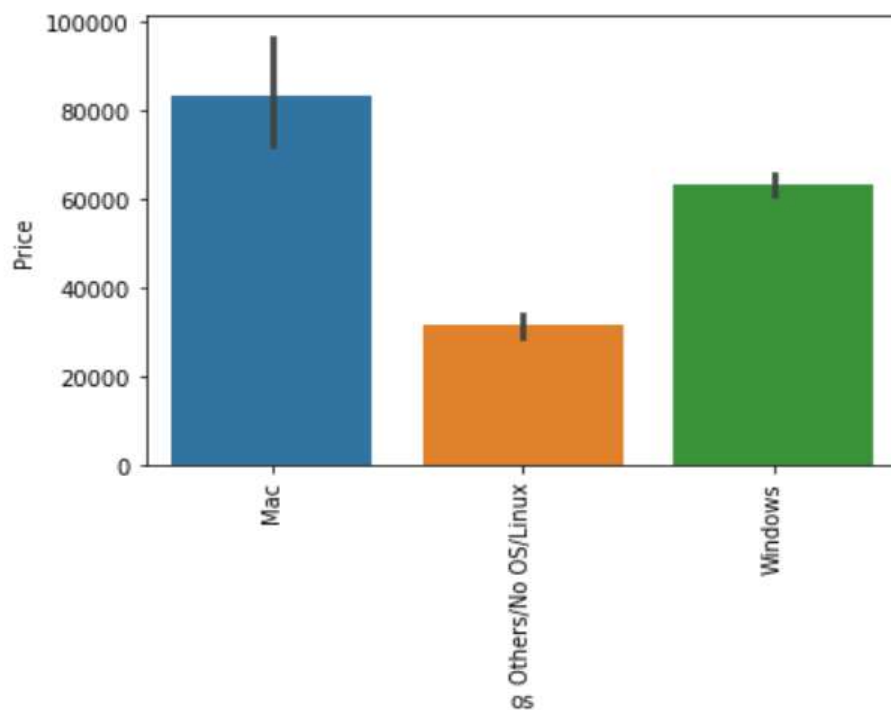
```



```

▶ sns.barplot(x=df['os'],y=df['Price'])
  plt.xticks(rotation='vertical')
  plt.show()

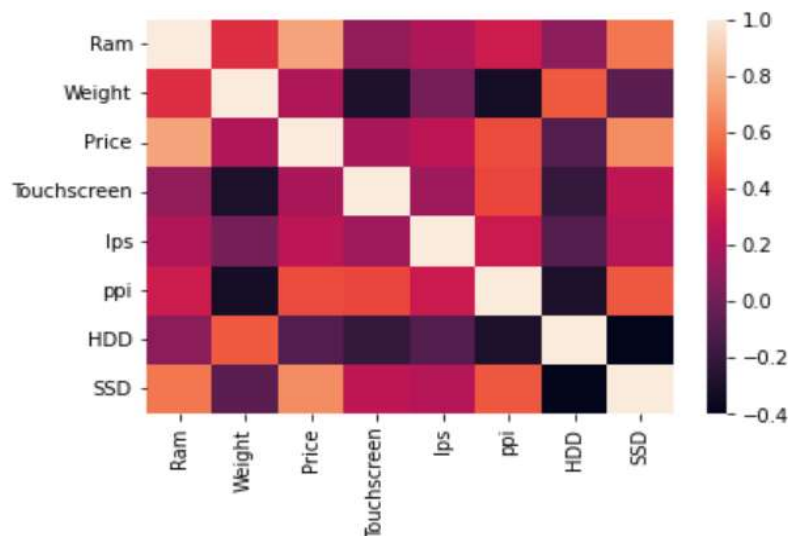
```



Correlation Matrix

```
sns.heatmap(df.corr())
```

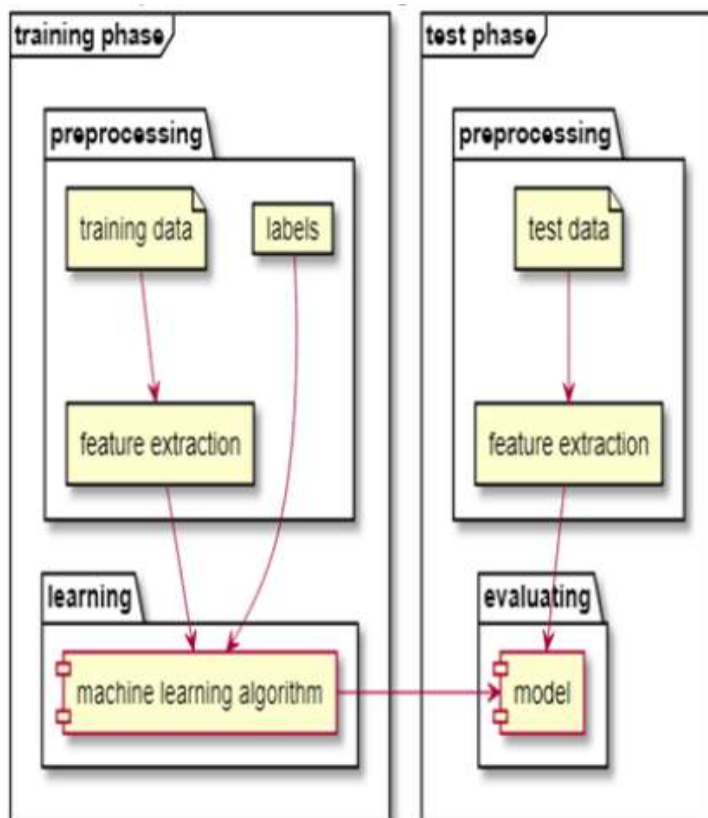
]: <Axes: >



Train,test ,Split

```
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)
```



Import libraries for model training

```
➤ from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import r2_score, mean_absolute_error

➤ from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForest
from sklearn.svm import SVR
```

Random Forest Algorithm:

Random Forest

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

step2 = RandomForestRegressor(n_estimators=100,
                             random_state=3,
                             max_samples=0.5,
                             max_features=0.75,
                             max_depth=15)

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))
```

/usr/local/lib/python3.9/dist-packages/sklearn/preprocessing/_encoders.py:868: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be removed in 1.4. `sparse_output` is ignored unless you leave `sparse` to its default value.

```
warnings.warn(
```

R2 score 0.8873402378382488

MAE 0.15860130110457718

Export the Model

Exporting the Model

```
➤ import pickle
pickle.dump(df,open('df_options.pkl','wb'))
pickle.dump(df,open('df_transformed.pkl','wb'))
pickle.dump(pipe,open('df_laptop_price_pred.pkl','wb'))
```

Deploy the App

```
app.py X
C:\Users\agdha> OneDrive\Desktop\Laptop-Price\Laptop-Price-Prediction-ML-main> app.py
1 import streamlit as st
2 import pickle
3 import numpy as np
4 import pandas as pd
5
6 model = pickle.load(open("laptop_price_pred_model.pkl", "rb"))
7 df = pickle.load(open("df_transformed.pkl", "rb"))
8 df_options = pickle.load(open("df_options.pkl", "rb"))
9 st.title("Laptop Price Predictor")
10
11 #company
12 company = st.selectbox("Which company's laptop would you like to purchase?", df_options['Company'].unique())
13
14 #type
15 typename = st.selectbox("What type of laptop are you looking for?", df_options['TypeName'].unique())
16
17 #ram
18 ram = st.selectbox("How many GB of RAM is needed?", df_options['Ram'].unique())
19 ram = int(ram)
20
21 #weight
22 weight = st.number_input("How many kgs should your laptop approximately weigh?")
23 weight = int(weight)
24
```

C:\Users> agdha > OneDrive > Desktop > Laptop-Price > Laptop-Price-Prediction-ML-main > app.py

```
31
32 #ips panel
33 ips = st.selectbox("Do you want IPS Display?", ['Yes','No'])
34 if ips == 'Yes':
35     ips = 1
36 else:
37     ips = 0
38
39 #screen size in inches
40 screen_size = st.number_input('Enter screen size [in inches]')
41 if screen_size == 0:
42     screen_size = 14
43
44 # resolution
45 resolution = st.selectbox('Select Screen Resolution',['1920x1080','1366x768','1600x900','3840x2160','3200x1800','2880x1800','2560x1600','25
46
47 #cpu
48 cpu_brand = st.selectbox("Which processor do you need?", df_options['Cpu brand'].unique())
49
50 #hdd
51 hdd = st.number_input("How many GB of HDD is required?")
52 hdd = int(hdd)
53
54 #ssd
55 ssd = st.number_input("How many GB of SSD is required?")
56 ssd = int(ssd)
57
```

ted Mode is intended for safe code browsing. Trust this window to enable all features. [Manage](#) [Learn More](#)

app.py X

C:\Users> agdha > OneDrive > Desktop > Laptop-Price > Laptop-Price-Prediction-ML-main > app.py

```
66 Y_res = int(resolution.split('x')[1])
67 ppi = ((X_res**2) + (Y_res**2))**0.5/screen_size
68
69
70 test = pd.DataFrame(columns=['Ram', 'Weight', 'Touchscreen', 'Ips', 'ppi', 'HDD', 'SSD', 'Company_Acer', 'Company_Apple', 'Company_Asus', 'C
71
72 # fill it with 0s
73 test.loc[0] = 0
74
75 # update relevant columns
76 test['Company_'+company] = 1
77 test['Type_'+typename] = 1
78 test['Ram'] = ram
79 test['Weight'] = weight
80 test['Touchscreen'] = touchscreen
81 test['Ips'] = ips
82 test['ppi'] = ppi
83 test['Cpu_brand_'+cpu_brand] = 1
84 test['HDD'] = hdd
85 test['SSD'] = ssd
86 test['Gpu_brand_'+gpu] = 1
87 test['os_'+os] = 1
88
89
90 if st.button('Predict'):
91     prediction = model.predict(test)
92     st.success(int(np.exp(prediction[0])))
```

DEPLOYMENT OUTPUT:

Laptop Price Predictor

Which company's laptop would you like to purchase?

Dell ▼

What type of laptop are you looking for?

Notebook ▼

How many GB of RAM is needed?

16 ▼

How many kgs should your laptop approximately weigh?

1.58 - +

Touchscreen? Y/N

No ▼

Do you want IPS Display?

Which processor do you need?

Intel Core i5 ▼

How many GB of HDD is required?

1024.00 - +

How many GB of SSD is required?

0.00 - +

Which brand's GPU are you looking for in your laptop?

Intel ▼

Which operating system will suit your needs best?

Windows ▼

Predict

63950

Conclusion:

In this project I predict the price of laptop on the basis of its specification. I predict the range from different online ecommerce websites. I predict for different brands and types of different laptop. I trained with different model among that Random Forest Algorithm Accuracy is more.

Accuracy of the model :**0.887**

I exported the predictive model by using the pickle library. And deployed the app using Streamlit. By using any web browser user or any clients can predict the price the laptop according to their requirements.

References

[1] International Journal of Computer Science and Mobile Computing.
Laptop Price Prediction using Machine Learning..

[2] https://www.researchgate.net/publication/50946368_Exploratory_data_analysis_in_the_context_of_data_mining_and_resampling

[3] <https://medium.com/analyticsvidhya/predicting-laptop-prices-using-mle60a0315b45a>

[4] <https://www.kaggle.com/code/danielbethell/laptop-prices-prediction>

[5] https://issuu.com/pricedetailsindia/docs/blog_compare_laptop_price_and_its_functions

Github Link

<https://github.com/Dharani2905/majorproject>