Laptop Price Prediction

A Data Science & ML Project

Submitted By
Abhikriti Moti, 12007073

Certification by the ALLSOFT SOLUTIONS – IBM CE

Some Credly Badges Earned during the learning..

[Click Here to Verify]

[Click Here to Verify]

ACHIEVEMENT

Allsoft Solutions
An ISO 9001 : 2008 Certified Company

PROJECT COMPLETION CERTIFICATE

In recognition of the commitment to achieve professional excellence this is to certify that Ms./Mr.

Abhikriti Moti

has successfully completed an Industry-oriented project.

Project Name	Laptop Price Prediction
Technologies Used	seaborn, pickle, streamlit, numpy, pandas, matplotlib, Decision Tree, Linear Reg, Ridge Reg, KNN
Reference No.	AIP/CEP2021/IN/13149
Training Date	June 2022 – July 2022
Training Duration	6 Weeks
Training Location	Online Live Mode

Program Co-ordinator
Industry/Academic Alliance

Business Partner

Director
Training and Development
Allsoft Solutions and Services

BIG DATA - ANALYTICS | IoT | ORACLE | J2EE | PHP | CLOUD COMPUTING

Google Data Analytics Professional Certificate

This additional course was done in continuation with the summer training project.

Verify at https://coursera.org/verify/professional-cert/D6YB6XXCWSKJ

Problem Statement

- If any user wants to buy a laptop, then the application should be compatible to provide a **tentative price of laptop** according to the user configurations.

- Data Set is noisy and needs lots of feature engineering, and pre-processing.

Project Lifecycle

Step 01	Step 02	Step 03	Step 04	Step 05
Data Pre-processing	Exploratory Data Analysis	Feature Engineering	Machine learning modeling	Deployment

Libraries Used

Streamlit

develop web apps and deploy them easily

Numpy

Fundamental package for scientific computing

Pandas

High-performance data manipulation

Seaborn

Library is for making Python statistical graphics

Matplotlib

Comprehensive library for creating static, animated, and interactive visualizations

Data Pre-processing Steps

Download the dataset from **Kaggle**:
https://www.kaggle.com/code/danielbethell/laptop-prices-prediction/data

1. Data quality assessment

Idea of its overall quality, relevance to project, and consistency.

```
In 195   1  df.shape

Out 195      (1303, 12)
```

```
In 196   1  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
```

2. Data cleaning and Transformation

- Remove data that does not belong in your dataset.
- Converting data from one format or structure into another.

```
In 199   1  df.drop(columns=['Unnamed: 0'],inplace=True)
         2  df.head()
```

```
In 200   1  df['Ram'] = df['Ram'].str.replace('GB','')
         2  df['Weight'] = df['Weight'].str.replace('kg','')
```

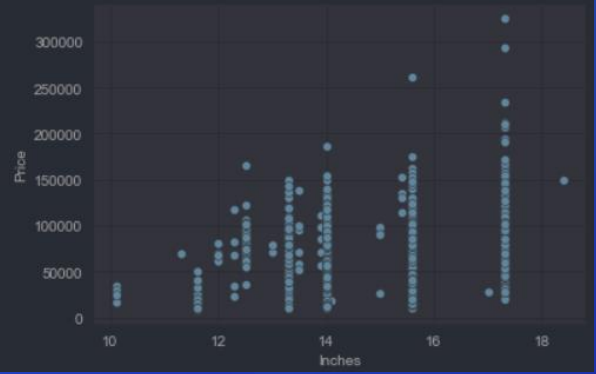
```
In 202   1  df['Ram'] = df['Ram'].astype('int32')
         2  df['Weight'] = df['Weight'].astype('float32')
```

Exploratory Data Analysis

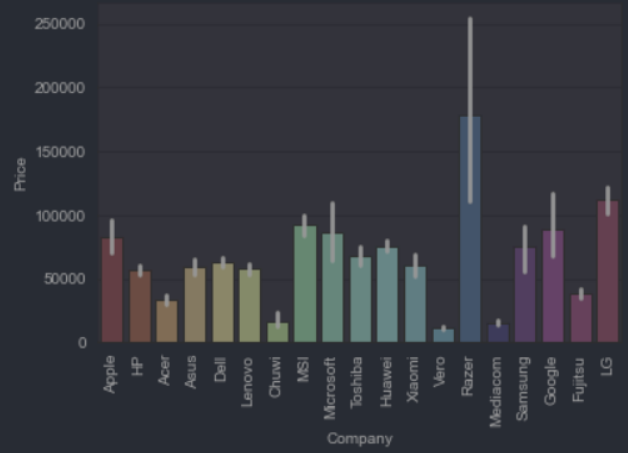
- Initial investigations on data to discover patterns, spot anomalies, test hypothesis and check assumptions

```
In 210  1  sns.scatterplot(x=df['Inches'],y=df['Price'])

Out 210    <AxesSubplot:xlabel='Inches', ylabel='Price'>
```




```
In 206  1  import matplotlib.pyplot as plt
        2  sns.barplot(x=df['Company'],y=df['Price'])
        3  plt.xticks(rotation='vertical')
        4  plt.show()
```



Feature Engineering

- Leverages data to create new variables
- Extracting relevant features from the data to train ML algorithms

In 229
```
df['ppi'] = (((df['X_res']**2) + (df['Y_res']**2))**0.5/df['Inches']).astype('float')
```

In 236
```
def fetch_processor(text):
    if text == 'Intel Core i7' or text == 'Intel Core i5' or text == 'Intel Core i3':
        return text
    else:
        if text.split()[0] == 'Intel':
            return 'Other Intel Processor'
        else:
            return 'AMD Processor'
```

In 237
```
df['Cpu brand'] = df['Cpu Name'].apply(fetch_processor)
```

Machine learning modeling

Linear Regression

- Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables.
- Finds how the value of the dependent variable is changing according to the value of the independent variable.

```
In 283
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = LinearRegression()

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))
```

Random Forest

- 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.

- It predicts output with high accuracy, even for the large dataset it runs efficiently.

In 289

```
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))
```

```
R2 score 0.8873402378382488
MAE 0.15860130110457718
```

Exporting the Model

- Save the pipeline object for the development of the project website
- Save the ML model using pickel

```
In 291  1  import pickle
        2
        3  pickle.dump(df,open('df.pkl','wb'))
        4  pickle.dump(pipe,open('pipe.pkl','wb'))
```

Creating a Web Application

- Use streamlit to create a web app

```
import streamlit as st
```

- create a file named main.py in the same working directory where we will write code for streamlit.

- Import the model

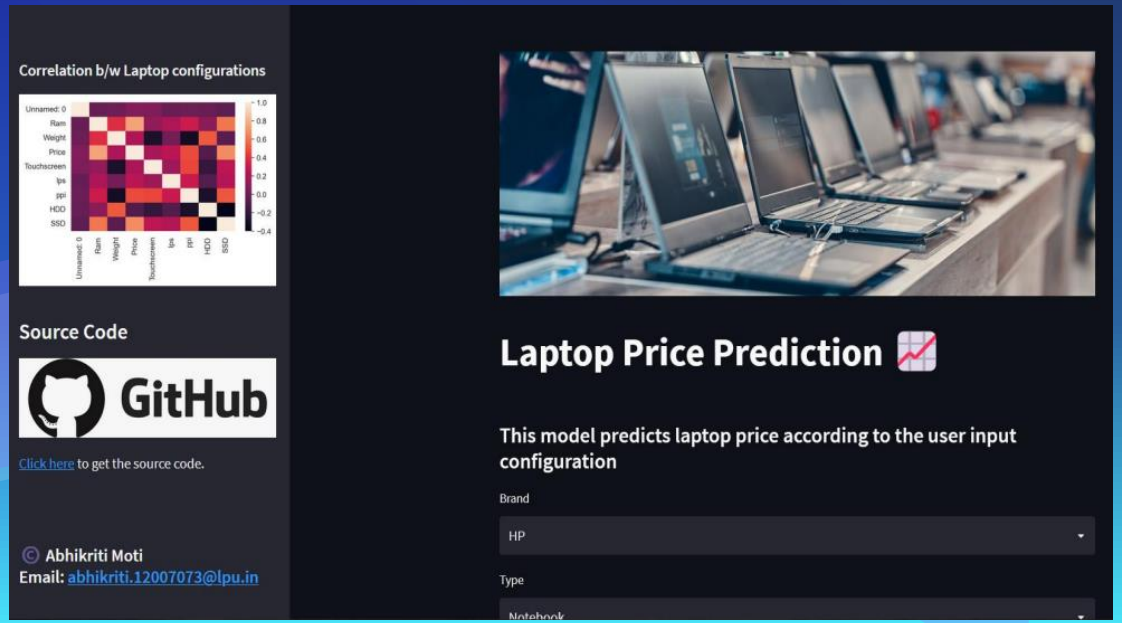
```
# import the model
pipe = pickle.load(open('pipe.pkl','rb'))
df = pickle.load(open('df.pkl','rb'))
```

- Run the app file using the below command

```
PS D:\OneDrive - Lovely Professional University\Laptop_Price_Prediction> py -m streamlit run main.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.28.10.17:8501
```



Web Application

Laptop Price Prediction 📈

This model predicts laptop price according to the user input configuration

Brand: HP

Type: Notebook

RAM (in GB): 8

Weight of the Laptop: 1.00

Touchscreen: Yes

IPS: No

Screen Size: 1.00

Screen Resolution: 1920x1080

CPU: Intel Core i5

HDD(in GB): 512

SSD(in GB): 256

GPU: Intel

OS: Windows

Predict Price

The predicted price of this configuration is

₹ 72475

Dataset Used 🔎

kaggle link

	Unnar	Company	TypeName	Inches	ScreenResolution	Cpu
0	0	Apple	Ultrabook	13.3000	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz
1	1	Apple	Ultrabook	13.3000	1440x900	Intel Core i5 1.8GHz
2	2	HP	Notebook	15.6000	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz
3	3	Apple	Ultrabook	15.4000	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz
4	4	Apple	Ultrabook	13.3000	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz
5	5	Acer	Notebook	15.6000	1366x768	AMD A9-Series 9420 3GHz
6	6	Apple	Ultrabook	15.4000	IPS Panel Retina Display 2880x1800	Intel Core i7 2.2GHz
7	7	Apple	Ultrabook	13.3000	1440x900	Intel Core i5 1.8GHz
8	8	Asus	Ultrabook	14.0000	Full HD 1920x1080	Intel Core i7 8550U 1.8GHz
9	9	Acer	Ultrabook	14.0000	IPS Panel Full HD 1920x1080	Intel Core i5 8250U 1.6GHz

Learning Outcomes

- Save Demonstrate proficiency with statistical analysis of data.
- Develop the ability to build and assess data-based models.
- Investigate data and designs by loading, extracting, transforming, and analyzing data from various sources.
- Implement histograms, classifiers, decision trees, sampling, linear regression, and projectiles in a scripting language.

Thanks!

Do you have any questions?
abhikritimoti@outlook.com