ML Project

Exploratory Data Analysis & Price Prediction of Laptops dataset (kaggle)

Submission By :

Ansh Avi Khanna

IMT2021038

Laptop Price Prediction

1. Original Dataset
2. Cleaned Dataset
3. Data Analysis
4. Data Modeling
5. Model Deployment

1. Original Dataset

I have worked on a dataset on laptops (from kaggle). The dataset contains details & specifications of laptops. Considering all the specs entered by the user, the model returns the predicted price (in euros) of the given configuration.

```
lp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Unnamed: 0        1303 non-null   int64  
 1   Company           1303 non-null   object 
 2   Product           1303 non-null   object 
 3   TypeName          1303 non-null   object 
 4   Inches            1303 non-null   float64
 5   ScreenResolution  1303 non-null   object 
 6   Cpu               1303 non-null   object 
 7   Ram               1303 non-null   object 
 8   Memory            1303 non-null   object 
 9   Gpu               1303 non-null   object 
 10  OpSys             1303 non-null   object 
 11  Weight            1303 non-null   object 
 12  Price_euros       1303 non-null   float64
dtypes: float64(2), int64(1), object(10)
memory usage: 132.5+ KB
```

2. Cleaned Dataset

Explored the cleaned data to add more columns to the dataset for better prediction.

```
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Company           1303 non-null   object
 1   Product           1303 non-null   string
 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(GB)           1303 non-null   int32
 7   Memory            1303 non-null   object
 8   Gpu               1303 non-null   object
 9   OpSys             1303 non-null   object
 10  Weight(kg)        1303 non-null   float64
 11  Price_euros       1303 non-null   float64
 12  CPU_Company       1303 non-null   object
 13  Weight_Category   1303 non-null   object
 14  SSD               1303 non-null   int64
 15  HDD               1303 non-null   int64
 16  Hybrid            1303 non-null   int64
 17  Flash             1303 non-null   int64
 18  IPS Display       1303 non-null   int64
 19  TouchScreen       1303 non-null   int64
 20  ppi               1303 non-null   float64
 21  Resolution        1303 non-null   object
 22  CPU_brand         1303 non-null   object
 23  ClockSpeed(GHz)   1303 non-null   float64
 24  GPU_Company       1303 non-null   object
 25  OS                1303 non-null   object
```

3. Data Analysis

Used seaborn and matplotlib for data analysis and visualization.

The attached pdf contains all the plots and observations.

4. Data Modeling

Used scikit learn to train and test the dataset. Applied linear regression on the dataset.

Training Data

```
: from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4,random_state=2)
```

Applying Linear Regression to the dataset for Price Prediction

[130]:
```
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
```

[131]:
```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse=False,drop='first',handle_unknown='ignore'),[0,1,2,3,4])
],remainder='passthrough')

step2 = LinearRegression()

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

lm.fit(X_train,Y_train)

Y_hat = lm.predict(X_test)

print('R2 score : ',r2_score(Y_test,Y_hat))
print('MAE : ',mean_absolute_error(Y_test,Y_hat))
```

```
R2 score :  0.8053019186480996
MAE :  0.20289500270563213
```

5. Model Deployment

Used pickle to export the model.

Used streamlit to deploy the model as a website (local host).

- Command to run website.py : streamlit run <file path>