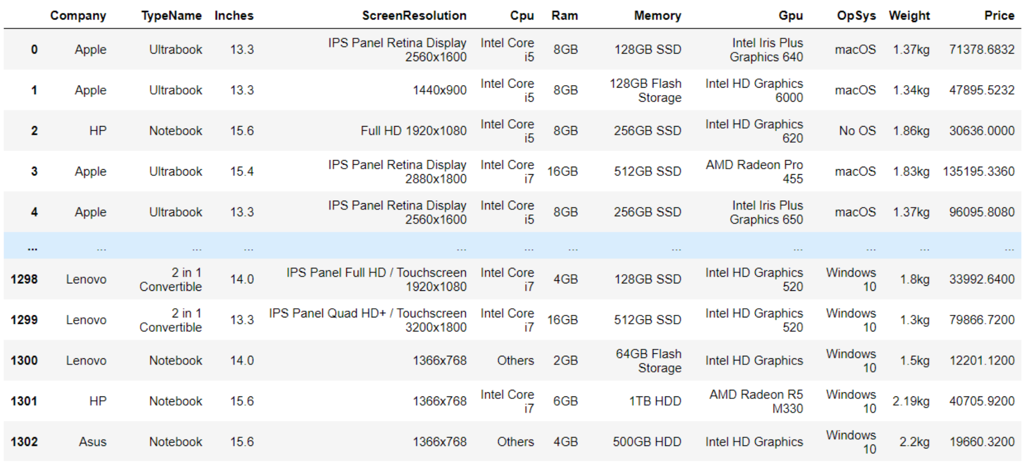
Flask Deployment of Laptop Price Prediction System

Report date: 26 August 2023
Internship Batch: LISUM24
Version: 1.0
Data intake by: Riwaj Neupane
Data intake reviewer: Data Glacier
Data storage location: https://github.com/Riwaj22/Data-Glacier/tree/main/week%204

Laptop_Data Details:

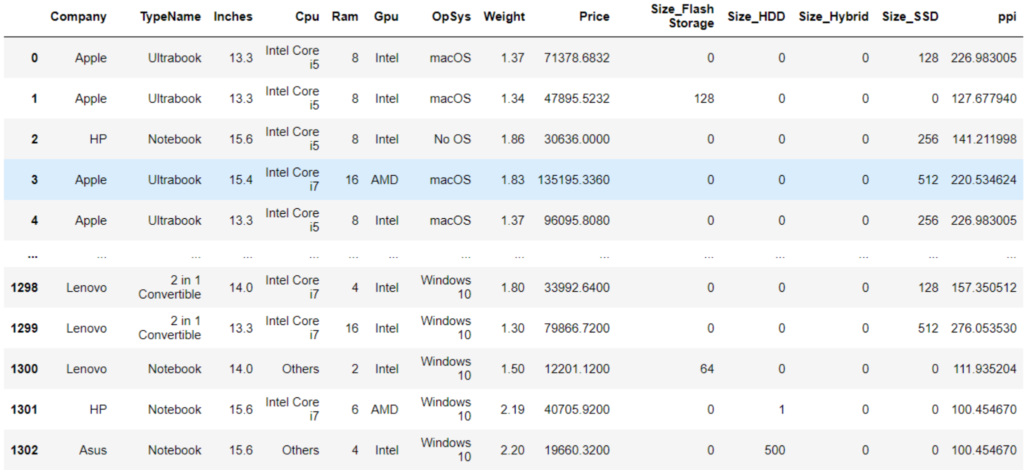
From kaggle: https://www.kaggle.com/datasets/mohidabdulrehman/laptop-price-dataset

Before cleaning:



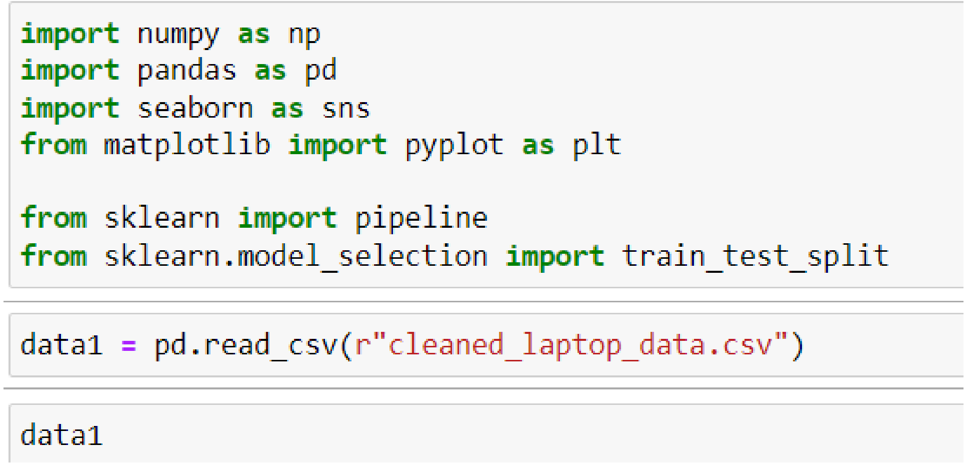
	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	71378.6832
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	47895.5232
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	30636.0000
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	135195.3360
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	96095.8080
...	...	...	...	...	...	...	...	...	...	...	...
1298	Lenovo	2 in 1 Convertible	14.0	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7	4GB	128GB SSD	Intel HD Graphics 520	Windows 10	1.8kg	33992.6400
1299	Lenovo	2 in 1 Convertible	13.3	IPS Panel Quad HD+ / Touchscreen 3200x1800	Intel Core i7	16GB	512GB SSD	Intel HD Graphics 520	Windows 10	1.3kg	79866.7200
1300	Lenovo	Notebook	14.0	1366x768	Others	2GB	64GB Flash Storage	Intel HD Graphics	Windows 10	1.5kg	12201.1200
1301	HP	Notebook	15.6	1366x768	Intel Core i7	6GB	1TB HDD	AMD Radeon R5 M330	Windows 10	2.19kg	40705.9200
1302	Asus	Notebook	15.6	1366x768	Others	4GB	500GB HDD	Intel HD Graphics	Windows 10	2.2kg	19660.3200

After Cleaning:



	Company	TypeName	Inches	Cpu	Ram	Gpu	OpSys	Weight	Price	Size_Flash Storage	Size_HDD	Size_Hybrid	Size_SSD	ppi
0	Apple	Ultrabook	13.3	Intel Core i5	8	Intel	macOS	1.37	71378.6832	0	0	0	128	226.983005
1	Apple	Ultrabook	13.3	Intel Core i5	8	Intel	macOS	1.34	47895.5232	128	0	0	0	127.677940
2	HP	Notebook	15.6	Intel Core i5	8	Intel	No OS	1.86	30636.0000	0	0	0	256	141.211998
3	Apple	Ultrabook	15.4	Intel Core i7	16	AMD	macOS	1.83	135195.3360	0	0	0	512	220.534624
4	Apple	Ultrabook	13.3	Intel Core i5	8	Intel	macOS	1.37	96095.8080	0	0	0	256	226.983005
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1298	Lenovo	2 in 1 Convertible	14.0	Intel Core i7	4	Intel	Windows 10	1.80	33992.6400	0	0	0	128	157.350512
1299	Lenovo	2 in 1 Convertible	13.3	Intel Core i7	16	Intel	Windows 10	1.30	79866.7200	0	0	0	512	276.053530
1300	Lenovo	Notebook	14.0	Others	2	Intel	Windows 10	1.50	12201.1200	64	0	0	0	111.935204
1301	HP	Notebook	15.6	Intel Core i7	6	AMD	Windows 10	2.19	40705.9200	0	1	0	0	100.454670
1302	Asus	Notebook	15.6	Others	4	Intel	Windows 10	2.20	19660.3200	0	500	0	0	100.454670

Data Import:



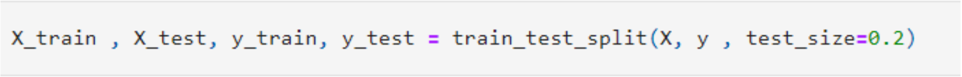
```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt

from sklearn import pipeline
from sklearn.model_selection import train_test_split
```

```
data1 = pd.read_csv(r"cleaned_laptop_data.csv")
```

```
data1
```

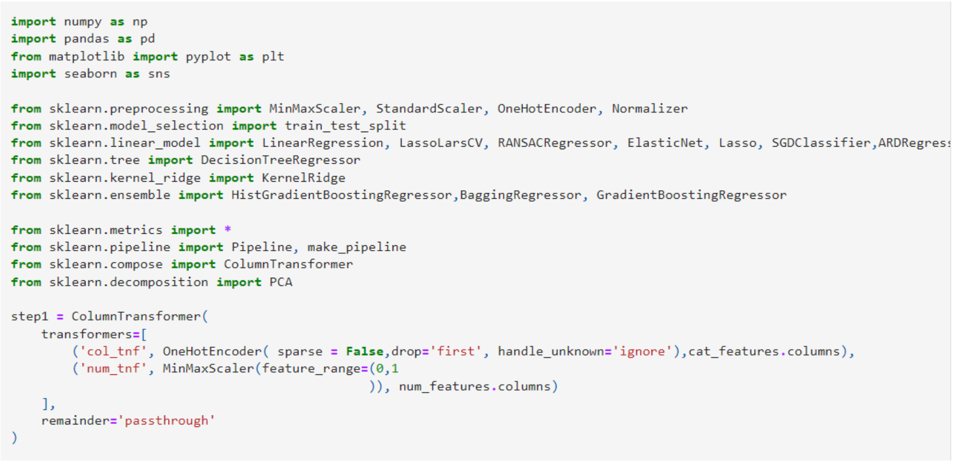
Target Variable: Price

Data Split:

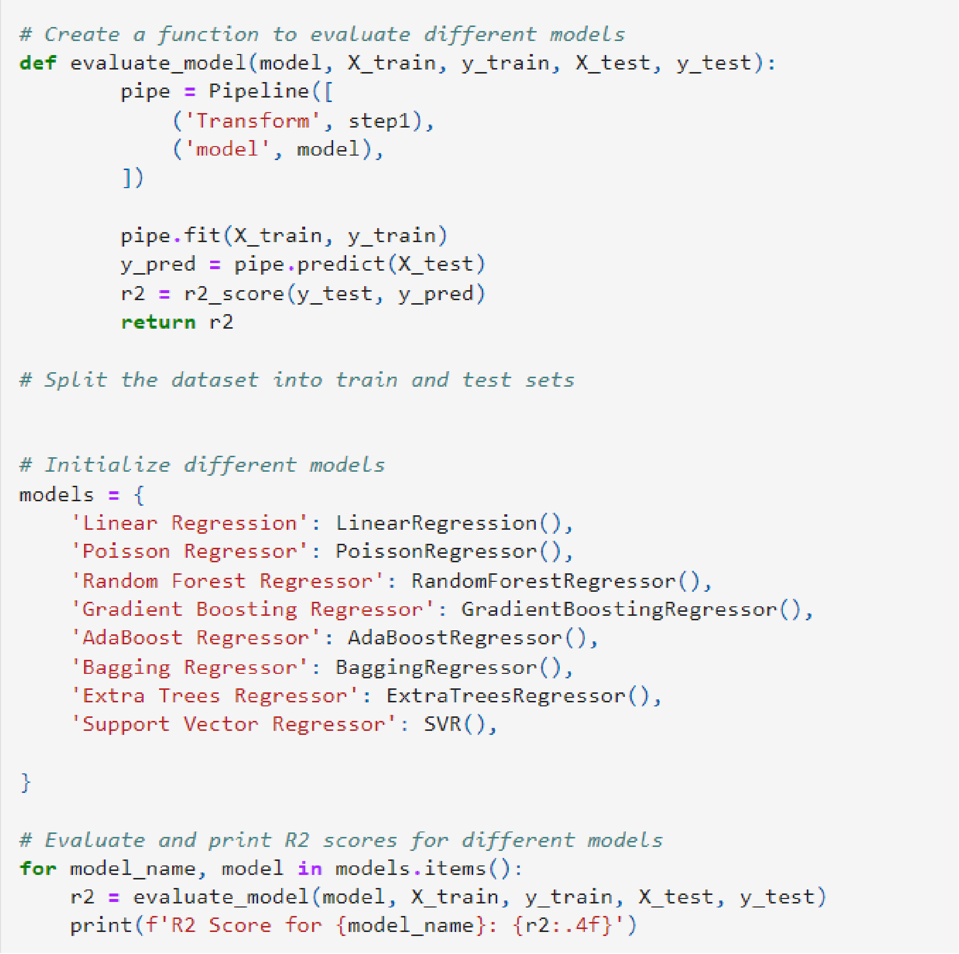


```
X_train , X_test, y_train, y_test = train_test_split(X, y , test_size=0.2)
```

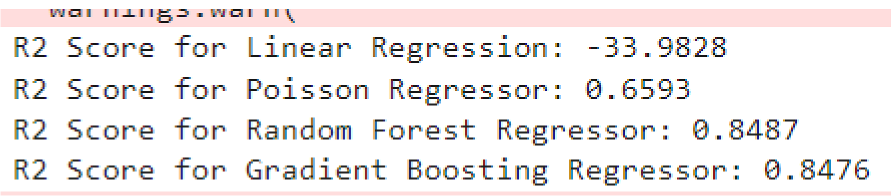
Model Building:



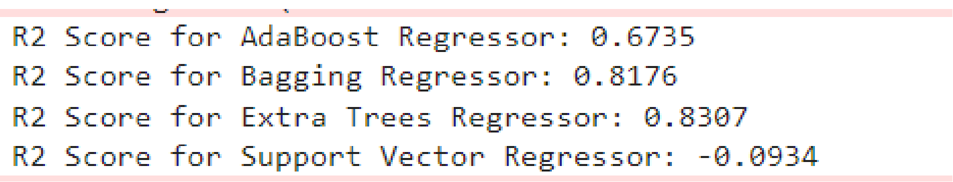
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns

from sklearn.preprocessing import MinMaxScaler, StandardScaler, OneHotEncoder, Normalizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LassoLarsCV, RANSACRegressor, ElasticNet, Lasso, SGDClassifier,ARDRegres
from sklearn.tree import DecisionTreeRegressor
from sklearn.kernel_ridge import KernelRidge
from sklearn.ensemble import HistGradientBoostingRegressor,BaggingRegressor, GradientBoostingRegressor

from sklearn.metrics import *
from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.compose import ColumnTransformer
from sklearn.decomposition import PCA

step1 = ColumnTransformer(
    transformers=[
        ('col_tnf', OneHotEncoder( sparse = False,drop='first', handle_unknown='ignore'),cat_features.columns),
        ('num_tnf', MinMaxScaler(feature_range=(0,1
                                                )), num_features.columns)
    ],
    remainder='passthrough'
)
```



```
# Create a function to evaluate different models
def evaluate_model(model, X_train, y_train, X_test, y_test):
        pipe = Pipeline([
            ('Transform', step1),
            ('model', model),
        ])

        pipe.fit(X_train, y_train)
        y_pred = pipe.predict(X_test)
        r2 = r2_score(y_test, y_pred)
        return r2

# Split the dataset into train and test sets


# Initialize different models
models = {
    'Linear Regression': LinearRegression(),
    'Poisson Regressor': PoissonRegressor(),
    'Random Forest Regressor': RandomForestRegressor(),
    'Gradient Boosting Regressor': GradientBoostingRegressor(),
    'AdaBoost Regressor': AdaBoostRegressor(),
    'Bagging Regressor': BaggingRegressor(),
    'Extra Trees Regressor': ExtraTreesRegressor(),
    'Support Vector Regressor': SVR(),

}

# Evaluate and print R2 scores for different models
for model_name, model in models.items():
    r2 = evaluate_model(model, X_train, y_train, X_test, y_test)
    print(f'R2 Score for {model_name}: {r2:.4f}')
```



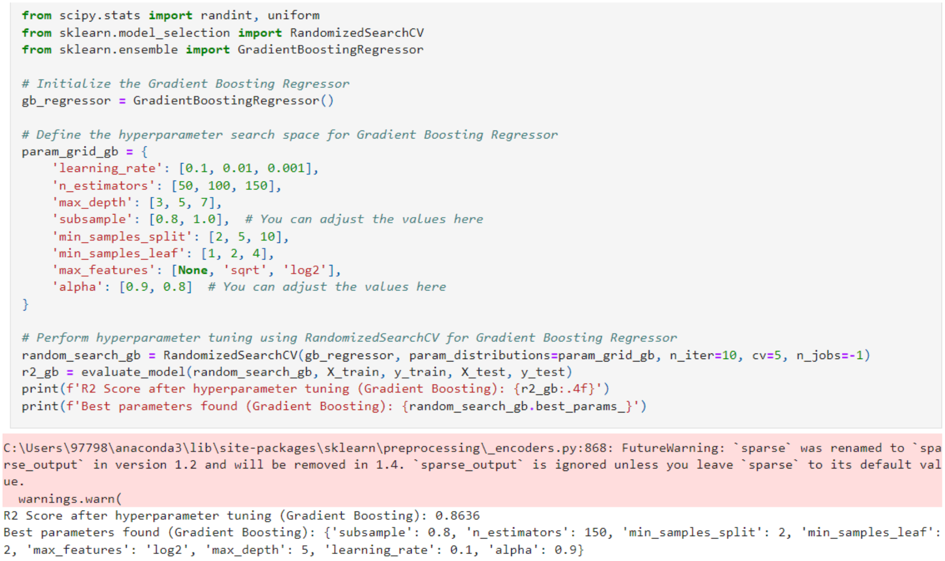
```
    warnings.warn(
R2 Score for Linear Regression: -33.9828
R2 Score for Poisson Regressor: 0.6593
R2 Score for Random Forest Regressor: 0.8487
R2 Score for Gradient Boosting Regressor: 0.8476
```



```
R2 Score for AdaBoost Regressor: 0.6735
R2 Score for Bagging Regressor: 0.8176
R2 Score for Extra Trees Regressor: 0.8307
R2 Score for Support Vector Regressor: -0.0934
```

Selection of Gradient Boosting Regressor

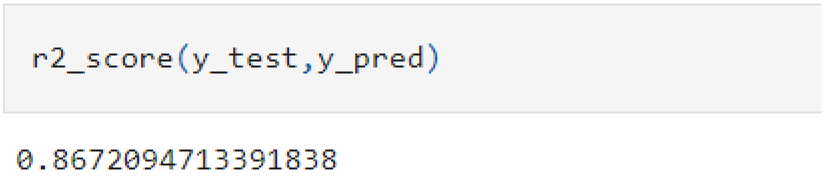
Hyperparameter Tuning the model



```
from scipy.stats import randint, uniform
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import GradientBoostingRegressor

# Initialize the Gradient Boosting Regressor
gb_regressor = GradientBoostingRegressor()

# Define the hyperparameter search space for Gradient Boosting Regressor
param_grid_gb = {
    'learning_rate': [0.1, 0.01, 0.001],
    'n_estimators': [50, 100, 150],
    'max_depth': [3, 5, 7],
    'subsample': [0.8, 1.0],  # You can adjust the values here
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': [None, 'sqrt', 'log2'],
    'alpha': [0.9, 0.8]  # You can adjust the values here
}

# Perform hyperparameter tuning using RandomizedSearchCV for Gradient Boosting Regressor
random_search_gb = RandomizedSearchCV(gb_regressor, param_distributions=param_grid_gb, n_iter=10, cv=5, n_jobs=-1)
r2_gb = evaluate_model(random_search_gb, X_train, y_train, X_test, y_test)
print(f'R2 Score after hyperparameter tuning (Gradient Boosting): {r2_gb:.4f}')
print(f'Best parameters found (Gradient Boosting): {random_search_gb.best_params_}')
```

```
C:\Users\97798\anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:868: FutureWarning: `sparse` was renamed to `spa
rse_output` in version 1.2 and will be removed in 1.4. `sparse_output` is ignored unless you leave `sparse` to its default val
ue.
  warnings.warn(
R2 Score after hyperparameter tuning (Gradient Boosting): 0.8636
Best parameters found (Gradient Boosting): {'subsample': 0.8, 'n_estimators': 150, 'min_samples_split': 2, 'min_samples_leaf':
2, 'max_features': 'log2', 'max_depth': 5, 'learning_rate': 0.1, 'alpha': 0.9}
```

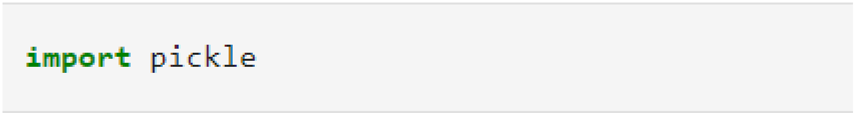
Evaluating model:



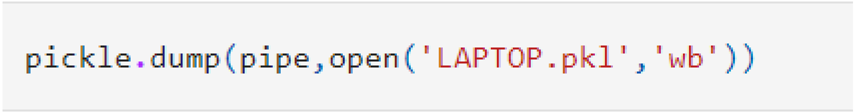
```
r2_score(y_test,y_pred)
```

```
0.8672094713391838
```

Saving model:



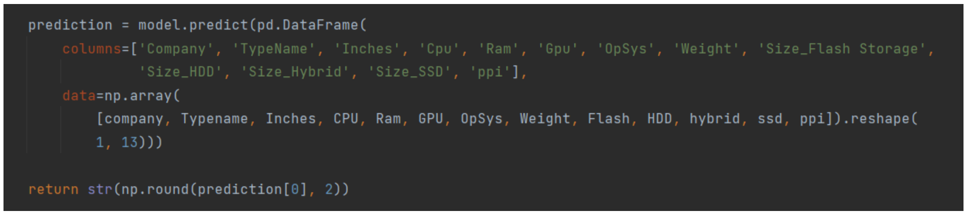
```
import pickle
```



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

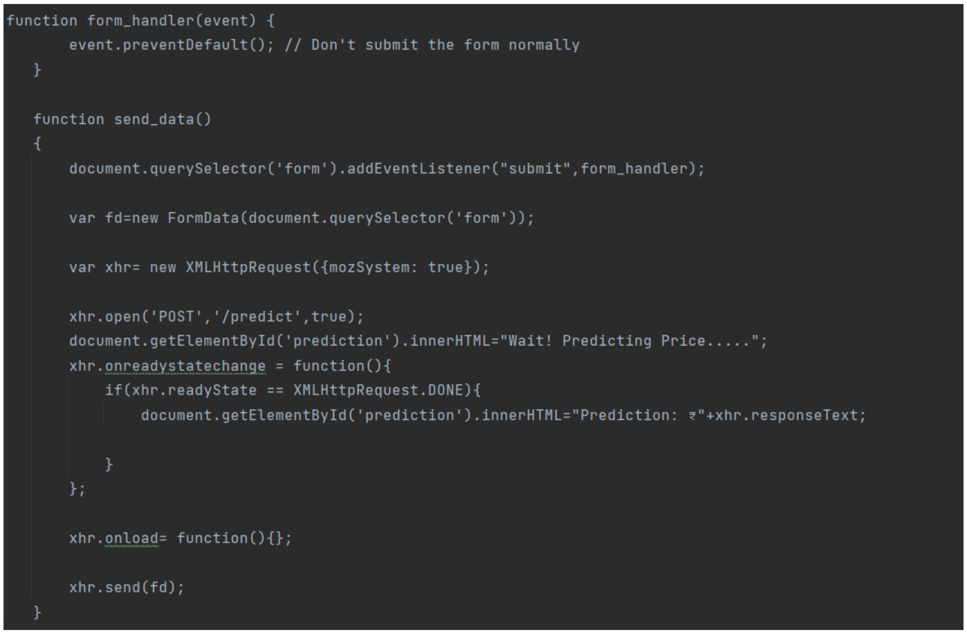
Deployment:

Prediction from Website



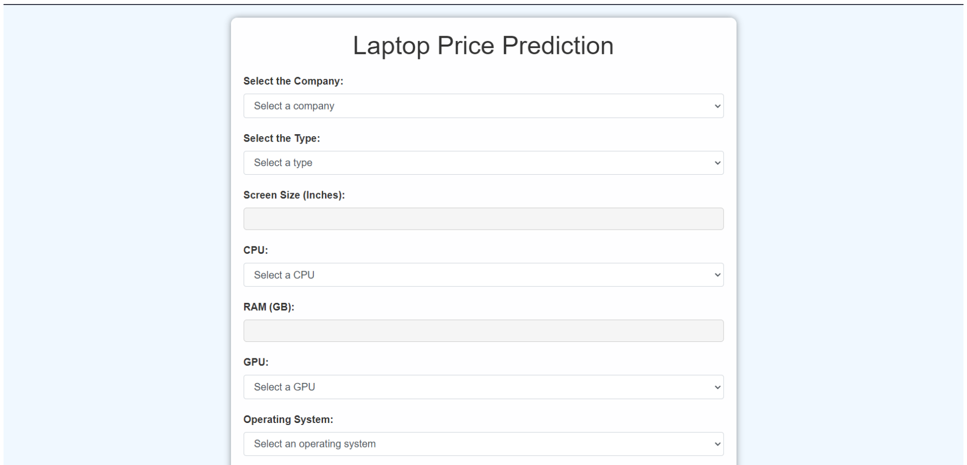
```
prediction = model.predict(pd.DataFrame(
    columns=['Company', 'TypeName', 'Inches', 'Cpu', 'Ram', 'Gpu', 'OpSys', 'Weight', 'Size_Flash Storage',
             'Size_HDD', 'Size_Hybrid', 'Size_SSD', 'ppi'],
    data=np.array(
        [company, Typename, Inches, CPU, Ram, GPU, OpSys, Weight, Flash, HDD, hybrid, ssd, ppi]).reshape(
        1, 13)))

return str(np.round(prediction[0], 2))
```

Prediction Handling



```
function form_handler(event) {
       event.preventDefault(); // Don't submit the form normally
   }

   function send_data()
   {
       document.querySelector('form').addEventListener("submit",form_handler);

       var fd=new FormData(document.querySelector('form'));

       var xhr= new XMLHttpRequest({mozSystem: true});

       xhr.open('POST','/predict',true);
       document.getElementById('prediction').innerHTML="Wait! Predicting Price.....";
       xhr.onreadystatechange = function(){
           if(xhr.readyState == XMLHttpRequest.DONE){
               document.getElementById('prediction').innerHTML="Prediction: ₹"+xhr.responseText;

           }
       };

       xhr.onload= function(){};

       xhr.send(fd);
   }
```

Website Snapshots:



Laptop Price Prediction

Select the Company: Select a company

Select the Type: Select a type

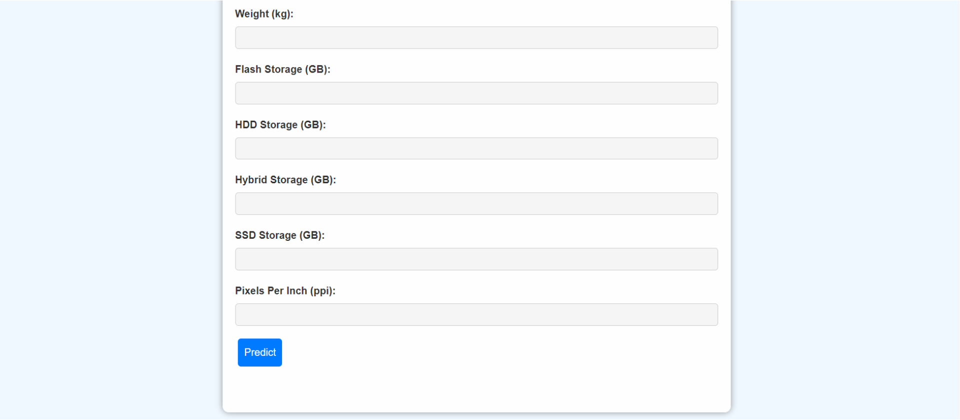
Screen Size (Inches):

CPU: Select a CPU

RAM (GB):

GPU: Select a GPU

Operating System: Select an operating system



Weight (kg):

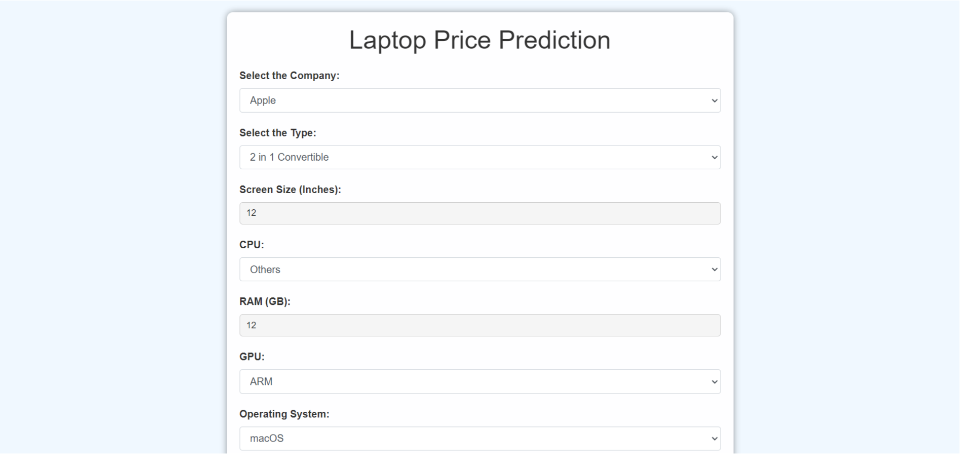
Flash Storage (GB):

HDD Storage (GB):

Hybrid Storage (GB):

SSD Storage (GB):

Pixels Per Inch (ppi):



Laptop Price Prediction

Select the Company: Apple

Select the Type: 2 in 1 Convertible

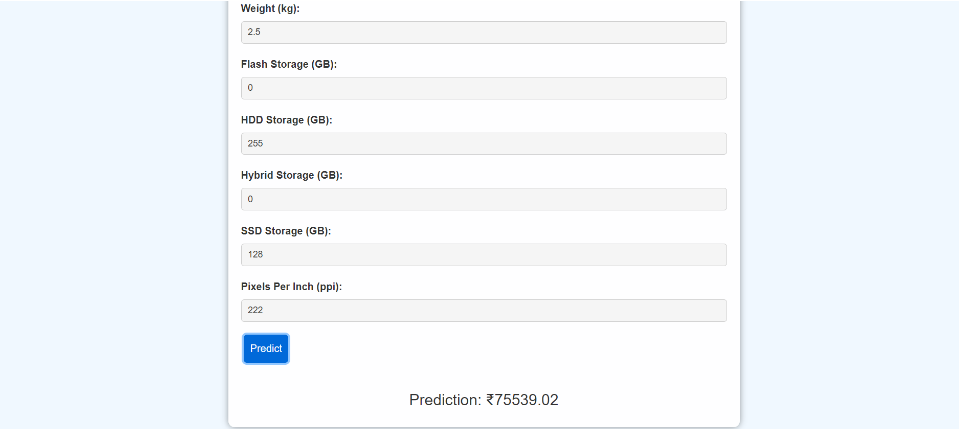
Screen Size (Inches): 12

CPU: Others

RAM (GB): 12

GPU: ARM

Operating System: macOS



Weight (kg):

2.5

Flash Storage (GB):

0

HDD Storage (GB):

255

Hybrid Storage (GB):

0

SSD Storage (GB):

128

Pixels Per Inch (ppi):

222

Prediction: ₹75539.02