Laptop Price Prediction Using Machine Learning

# Project Overview

This project aims to predict laptop prices based on various features such as brand, type, RAM, weight, screen size, CPU, GPU, and more. The goal is to help SmartTech Co. understand pricing trends and assist customers in making better decisions when purchasing laptops.  
  
The model uses machine learning techniques to build a predictive model that outputs the predicted price of a laptop based on user inputs. The project leverages a user-friendly web application built with Streamlit to allow real-time predictions.

# Technologies Used

- Python: The core programming language used for model training and data processing.  
- Pandas: Data manipulation and analysis library.  
- NumPy: For numerical computations.  
- scikit-learn: For machine learning model building, preprocessing, and pipeline management.  
- Streamlit: A Python framework for building interactive web applications.  
- Pickle: For saving and loading trained models.  
- pandas and NumPy libraries: For managing and processing data efficiently.

# Key Steps in the Project

## 1. Data Collection

The dataset used for this project contains information about various laptop features. It includes columns such as:  
- Brand  
- Type  
- RAM (in GB)  
- Weight  
- Touchscreen availability  
- IPS screen  
- Screen size  
- Screen resolution  
- CPU brand  
- HDD (in GB)  
- SSD (in GB)  
- GPU brand  
- Operating System

## 2. Data Preprocessing

The preprocessing steps include:  
- Handling missing values  
- Encoding categorical variables using OneHotEncoder within a ColumnTransformer  
- Feature scaling, if required (though not explicitly mentioned in the code)

## 3. Model Training

We use a Random Forest Regressor or another regression model (e.g., ExtraTreesRegressor) to predict laptop prices based on the input features.  
The model is trained using historical data and is evaluated on its performance using metrics such as Mean Squared Error (MSE) or R² score.

## 4. Model Serialization

After training the model, it is saved using Pickle to allow easy loading in the Streamlit application for real-time predictions.

## 5. Deployment (Streamlit App)

A Streamlit application allows users to input their laptop configuration, which is then processed by the trained model to predict the price. The application includes:  
- Dropdown menus for selecting the brand, type, RAM, CPU, GPU, etc.  
- A button to trigger the price prediction  
- The predicted price is displayed in an interactive and user-friendly manner.

## 6. Price Prediction

Once the user selects the options and clicks the 'Predict Price' button, the selected features are processed (e.g., encoding categorical variables, calculating PPI from screen resolution, etc.) and the model predicts the price based on the user's laptop configuration.

## 7. Post-Prediction

The predicted price is displayed on the app interface. The model uses an exponential function (np.exp) to ensure that the price prediction is positive.

# Code Explanation

## 1. Imports

The required libraries for this project include sklearn for model building, pandas for data manipulation, and Streamlit for building the interactive app.

## 2. Loading the Model and Data

The trained machine learning pipeline and dataset are loaded using Pickle and pandas, respectively.

## 3. User Input Interface

The user is prompted to select various options related to the laptop's features, such as:  
- Brand  
- Type of laptop  
- RAM  
- Weight  
- Touchscreen availability  
and others.

## 4. Feature Processing and Price Prediction

The input features are converted into a query array and reshaped for the model. The predicted price is then displayed after processing.

## 5. Displaying the Prediction

The predicted price is displayed on the Streamlit interface, showing the estimated cost based on the laptop configuration.

# How to Use the Application

1. Select Features: Choose your laptop's specifications from the dropdown menus (e.g., Brand, Type, RAM, etc.).  
2. Click 'Predict Price': After selecting the desired configurations, click the 'Predict Price' button.  
3. View the Predicted Price: The app will display the predicted price based on the selected specifications.

# Future Improvements

- Model Optimization: Improve the model's accuracy by experimenting with different machine learning algorithms and hyperparameters.  
- User Authentication: Allow users to save and track their predicted prices.  
- Additional Features: Include more features (e.g., battery life, camera quality) to further improve the prediction accuracy.  
- Visualizations: Add visualizations like bar graphs for feature importance or a comparison of predicted vs. actual prices.

# Conclusion

This project demonstrates how machine learning can be used to predict laptop prices based on various features. It showcases the application of regression techniques in a practical scenario and is easily accessible through a user-friendly Streamlit web interface.

import pickle  
import numpy as np  
import streamlit as st  
import pandas as pd  
  
# Assuming the model and dataframe are already pickled  
pipe = pickle.load(open('pipe.pkl','rb'))  
df = pd.read\_csv('df.csv') # Load the CSV version of the dataframe  
  
st.title("Laptop Predictor")  
  
# brand  
company = st.selectbox('Brand',df['Company'].unique())  
  
# type of laptop  
type = st.selectbox('Type',df['TypeName'].unique())  
  
# RAM  
ram = st.selectbox('RAM(in GB)',[2,4,6,8,12,16,24,32,64])  
  
# weight  
weight = st.number\_input('Weight of the Laptop')  
  
# Touchscreen  
touchscreen = st.selectbox('Touchscreen',['No','Yes'])  
  
# IPS  
ips = st.selectbox('IPS',['No','Yes'])  
  
# screen size  
screen\_size = st.slider('Screensize in inches', 10.0, 18.0, 13.0)  
  
# resolution  
resolution = st.selectbox('Screen Resolution',['1920x1080','1366x768','1600x900','3840x2160','3200x1800','2880x1800','2560x1600','2560x1440','2304x1440'])  
  
# cpu  
cpu = st.selectbox('CPU',df['Cpu brand'].unique())  
  
hdd = st.selectbox('HDD(in GB)',[0,128,256,512,1024,2048])  
  
ssd = st.selectbox('SSD(in GB)',[0,8,128,256,512,1024])  
  
gpu = st.selectbox('GPU',df['Gpu brand'].unique())  
  
os = st.selectbox('OS',df['os'].unique())  
  
# Process the data and make predictions  
if st.button('Predict Price'):  
 # Convert categorical inputs to numerical equivalents  
 touchscreen = 1 if touchscreen == 'Yes' else 0  
 ips = 1 if ips == 'Yes' else 0  
  
 X\_res = int(resolution.split('x')[0])  
 Y\_res = int(resolution.split('x')[1])  
 ppi = ((X\_res\*\*2) + (Y\_res\*\*2))\*\*0.5 / screen\_size  
  
 query = np.array([company, type, ram, weight, touchscreen, ips, ppi, cpu, hdd, ssd, gpu, os])  
 query = query.reshape(1, -1)  
  
 # Use the model to predict the price  
 predicted\_price = np.exp(pipe.predict(query)[0])  
  
 st.title(f"The predicted price of this configuration is ₹{int(predicted\_price)}")