Laptop Price Prediction Project

1. Overview

The **Laptop Price Prediction** project aims to build a machine learning model that predicts the price of a laptop based on its specifications. By analyzing features such as RAM, storage, screen size, weight, and additional attributes, the model provides an estimated price. The system uses supervised learning techniques and is integrated with a Streamlit-based user interface for easy interaction.

2. Dataset Description

The dataset consists of various laptop specifications and their corresponding prices in Euros. The key attributes include:

- **Brand**: Manufacturer of the laptop.
- **Model**: Specific model name.
- **Processor**: CPU details.
- **RAM**: Size of RAM in GB.
- **Storage**: Type and capacity of storage (SSD/HDD).
- **Screen Size**: Display size in inches.
- **Weight**: Weight of the laptop in kg.
- **Touchscreen**: Indicator if the laptop has a touchscreen.
- **IPS Display**: Indicator for IPS panel.
- **Price (Euros)**: The target variable.

The dataset undergoes preprocessing to handle missing values, encode categorical variables, and prepare it for training.

3. Technologies & Libraries Used

- **Python** – Programming language for model development.
- **Pandas** – Data manipulation and cleaning.
- **NumPy** – Numerical computations.

- **Scikit-learn** – Machine learning model implementation.
- **Streamlit** – Web application framework for UI.
- **Joblib** – Model serialization.

4. Data Preprocessing

The dataset is pre-processed by handling missing values, encoding categorical variables, and saving a cleaned dataset.

```
import pandas as pd

import numpy as np

import os


# Load dataset

df = pd.read_csv("cleaned_laptop_prices.csv")


# Drop duplicate rows

df.drop_duplicates(inplace=True)


# Handle missing values

df.dropna(inplace=True)


# Convert categorical variables to numerical

df = pd.get_dummies(df, drop_first=True)


# Save cleaned dataset

df.to_csv("cleaned_laptop_prices.csv", index=False)

print("Preprocessing complete. Cleaned dataset saved.")
```

5. Model Training

A Random Forest Regressor is used for training the price prediction model.

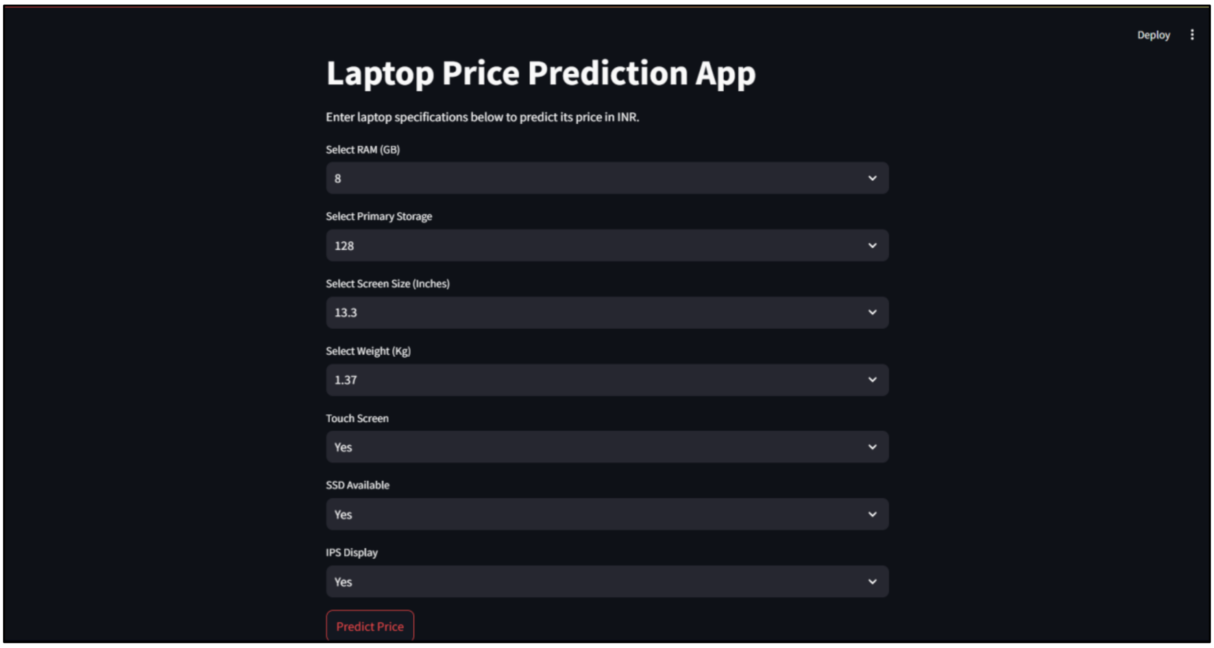
```
from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_absolute_error, mean_squared_error

import joblib


# Load dataset

df = pd.read_csv("cleaned_laptop_prices.csv")


# Define features and target variable

X = df.drop(columns=['Price_euros'])

y = df['Price_euros']


# Split dataset

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Train model

model = RandomForestRegressor(n_estimators=100, random_state=42)

model.fit(X_train, y_train)


# Evaluate model

y_pred = model.predict(X_test)

print("Mean Absolute Error:", mean_absolute_error(y_test, y_pred))

print("Mean Squared Error:", mean_squared_error(y_test, y_pred))

# Save model

joblib.dump(model, "laptop_price_model.pkl")

print("Model saved successfully.")
```

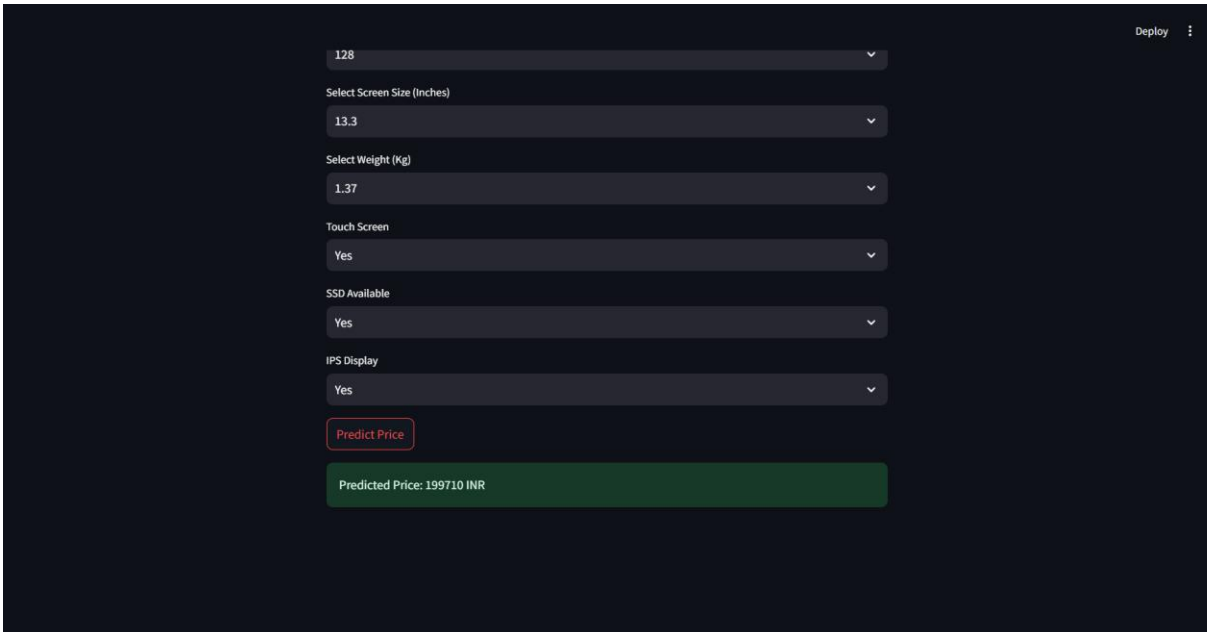
6. Model Testing & Streamlit Application

A Streamlit web app allows users to input laptop specifications and get price predictions.

```
import streamlit as st

import numpy as np

import pandas as pd

import joblib


# Load trained model

model = joblib.load("laptop_price_model.pkl")


# Define input fields

st.title("Laptop Price Prediction")

ram = st.selectbox("Select RAM (GB)", [4, 8, 16, 32])

storage = st.selectbox("Select Storage (GB)", [128, 256, 512, 1024])

screen_size = st.slider("Select Screen Size", 11, 17, 15)

weight = st.slider("Select Weight (Kg)", 1.0, 3.0, 1.5)

touchscreen = st.selectbox("Touchscreen", ['Yes', 'No'])

ssd = st.selectbox("SSD Available", ['Yes', 'No'])


# Predict price

if st.button("Predict Price"):

    features = np.array([[ram, storage, screen_size, weight, touchscreen == 'Yes', ssd == 'Yes']])

    price_prediction = model.predict(features)

    st.success(f"Predicted Price: €{price_prediction[0]:.2f}")
```

7. OUTPUT





8. Applications and Implications

The **Laptop Price Prediction System** has a wide range of applications across different industries and user groups. For consumers, it serves as a **cost estimation tool**, helping them make informed purchasing decisions by predicting the price of a laptop based on specifications. This is especially useful for students, professionals, and businesses looking for budget-friendly options without compromising on necessary features. **Retailers and e-commerce platforms** can integrate this model into their websites to provide real-time price estimates, enhancing customer experience and increasing transparency in pricing. **Manufacturers** can use it for market analysis, ensuring competitive pricing based on consumer demand and technological trends.

The implications of this project extend beyond just price prediction. By continuously updating the dataset, the model can **analyze market trends**, identify popular configurations, and help businesses strategize their product offerings. This predictive tool can also be adapted for use in other product categories, such as smartphones and tablets. Additionally, the system highlights the potential of **machine learning in price estimation**, showcasing how AI-driven models can simplify complex decision-making processes. As technology advances, integrating real-time exchange rates and price fluctuations will further enhance the model's accuracy and reliability.

9. Deployment & Future Enhancements

The model is deployed using Streamlit, and future enhancements may include:

- Adding more features like GPU type and build quality.
- Incorporating deep learning techniques.
- Expanding the dataset for better accuracy.

10. Conclusion

This project successfully implements a machine learning model to predict laptop prices, demonstrating data preprocessing, model training, and UI integration using Streamlit. This project not only benefits consumers but also aids **retailers, manufacturers, and market analysts** in understanding pricing trends and consumer behavior. The ability to integrate additional features, such as **real-time currency conversion and dynamic price updates**, makes this system highly scalable and adaptable for future enhancements. The combination of **machine learning and practical implementation** makes it a valuable tool in the evolving digital marketplace. Moving forward, improvements such as **deep learning models, larger datasets, and enhanced UX/UI features** can further refine the prediction accuracy and expand its usability. This project serves as a stepping stone for future innovations in automated price prediction and AI-driven decision-making systems.