**Laptop Price Prediction**

**Dataset Overview**

The dataset contains laptop specifications and their corresponding prices in Euros. Each row represents a laptop, and the columns describe various features that may influence the price.

**Dataset Description**

**Dataset link:-** [**Laptop Dataset**](https://drive.google.com/file/d/1jAdEtPqXaFsJIPfiCdolkLvF8H7JfUMS/view?usp=sharing)

| **Column Name** | **Description** | **Type** |
| --- | --- | --- |
| **Company** | Laptop manufacturer. | Categorical |
| **Product** | Brand and model of the laptop. | Categorical |
| **TypeName** | Type of laptop (e.g., Notebook, Ultrabook). | Categorical |
| **Inches** | Screen size of the laptop. | Numerical (float) |
| **ScreenResolution** | Screen resolution (e.g., 1920x1080). | Categorical |
| **CPU\_Company** | CPU manufacturer (e.g., Intel, AMD). | Categorical |
| **CPU\_Type** | Type of CPU. | Categorical |
| **CPU\_Frequency** | CPU speed in GHz. | Numerical (float) |
| **RAM (GB)** | RAM size in GB. | Numerical (int) |
| **Memory** | Storage capacity (e.g., HDD, SSD). | Categorical |
| **GPU\_Company** | GPU manufacturer (e.g., Nvidia, AMD). | Categorical |
| **GPU\_Type** | Type of GPU. | Categorical |
| **OpSys** | Operating system of the laptop. | Categorical |
| **Weight (kg)** | Weight of the laptop in kilograms. | Numerical (float) |
| **Price (Euro)** | Laptop price in Euros (target variable). | Numerical (float) |

**Project Overview**

The objective of this project is to create a machine learning model to predict laptop prices based on their specifications. The project follows a structured pipeline that includes data cleaning, preprocessing, exploration, modeling, and evaluation.

**Part 1: Data Ingestion**

Load the dataset and verify its integrity for analysis.

**Steps**

1. Load the dataset into a DataFrame.
2. Inspect the number of rows and columns, column names, and data types.
3. Check for missing values and duplicates.

**Part 2: Data Understanding and Exploration (EDA)**

Gain insights into the dataset, identify relationships, and detect patterns or anomalies.

**Steps**

1. **Univariate Analysis**
   * Explore the distribution of numerical features like Price (Euro), Inches, RAM (GB), Weight (kg), and CPU\_Frequency.
   * Examine categorical features like Company, TypeName, CPU\_Company for frequency distributions.
2. **Bivariate Analysis**
   * Analyze relationships between features and the target variable (Price (Euro)).
   * Use scatter plots for numerical features and box plots for categorical features.
3. **Multivariate Analysis**
   * Explore interactions between multiple features using heatmaps, pair plots, or other visualization techniques.

**Part 3: Data Cleaning**

Ensure the dataset is clean, consistent, and ready for analysis.

**Steps**

1. Handle missing values in numerical and categorical columns using appropriate methods like mean imputation or mode replacement.
2. Remove duplicate rows.
3. Standardize categorical values to ensure consistency.

**Part 4: Data Preprocessing**

Prepare the data for machine learning models by encoding, scaling, and transforming features.

**Steps**

1. Encode categorical variables using one-hot encoding for nominal features and label encoding for ordinal features.
2. Scale numerical features (Inches, CPU\_Frequency, RAM (GB), Weight (kg)) for uniformity.
3. Identify and handle outliers in numerical columns.
4. Transform or engineer features to improve data representation (e.g., create separate columns for HDD and SSD storage from Memory).

**Part 5: Model Building and Evaluation**

Train, evaluate, and fine-tune machine learning models for accurate price prediction.

**Steps**

1. Split the dataset into training and testing subsets (e.g., 80% training, 20% testing).
2. Train regression models such as Linear Regression,Decision Tree, Random Forest, and Gradient Boosting.
3. Optimize models using hyperparameter tuning techniques like GridSearchCV.
4. Evaluate models using metrics like R², MSE, and RMSE.
5. Select the best-performing model based on evaluation results and interpretability.