**University Of Uyo**

**Department Of Computer Engineering**

**Mini-Project Report**

**Laptop Price Prediction**

**Using Machine Learning Models**

**Submitted By:**

**Inameti Mercy Etim**

**23/EG/CO/003**

**Group J**

**CPE 221- Data Science and Analytics**

**Submitted To:**

**Mr Bright Agbor**

**Department Of Computer Engineering**

**Faculty Of Engineering**

**September,2025**

**Abstract**

The rapid growth of the laptop market has introduced a wide range of devices differing in brand, specifications, and price. Predicting laptop prices using machine learning techniques can provide valuable insights for consumers, manufacturers, and resellers.

This project develops an end-to-end machine learning pipeline using the Laptop Price Dataset from Kaggle. The process involved **data cleaning, exploratory data analysis (EDA), feature engineering, model building, evaluation, version control using GitHub, and deployment through a Streamlit web application**.

Two models were built: **Linear Regression and Random Forest Regressor**. Linear Regression, while simple and interpretable, generated negative prices and was not suitable for deployment. After retraining with Random Forest, the model achieved strong predictive performance (R² ≈ 1.0) and consistently produced positive, realistic prices.

An interactive **Streamlit Cloud web app** was deployed to allow users to predict laptop prices by entering specifications. The results highlight the importance of model selection in machine learning and demonstrate a complete end-to-end ML pipeline.

**Introduction**

Laptops are essential tools for education, business, and personal use. However, prices vary greatly depending on brand reputation, performance, and hardware features. Understanding the factors that influence prices can guide purchasing decisions and assist retailers in competitive pricing strategies.

The objective of this project was to build a **supervised learning model that accurately predicts the selling price of laptops** based on their specifications. . Using the Kaggle dataset, we applied the following workflow:

* Data cleaning and preprocessing
* Exploratory Data Analysis (EDA)
* Feature engineering
* Model building and evaluation
* Version control with GitHub
* Deployment with Streamlit

By completing this workflow, we demonstrate competence in data analysis, regression modeling, Git-based version control, and interactive model deployment.

**Dataset Description**

The dataset used is available on Kaggle: Laptop Price Dataset by muhammetvarl. It contains 1,303 entries with features describing laptop specifications and the corresponding price in Euros.

**Key Features**

* Company: Laptop brand (e.g., Apple, Dell, HP)
* Product: Brand and model
* TypeName: Category (Notebook, Ultrabook, Gaming, Workstation)
* Inches: Screen size
* ScreenResolution: Resolution (e.g., 1920x1080)
* Cpu: Processor details
* Ram: Memory capacity (GB)
* Memory: Storage type and size
* Gpu: Graphics card
* OpSys: Operating system (Windows, MacOS, Linux, etc.)
* Weight: Weight of laptop (kg)
* Price\_euros: Target variable – selling price (numeric, in Euros)

The dataset contains both categorical and numerical variables, requiring preprocessing before model training.

**Data Cleaning & Preprocessing**

Missing Values - No missing values were found in the dataset.

Outliers - Were handled using IQR Method

Encoding Categorical Variables – Label encoding and One-hot encoding were applied. This converted text categories into numerical columns suitable for ML.

**Exploratory Data Analysis (EDA)**

EDA provided insights into how different specifications affect price.

**RAM Size** - Correlation with price: 0.74. Scatterplots showed clear trend: higher RAM → higher price.

**Company (Brand)** - Highest average prices: Razer (~2387 €), LG (~2099 €), Apple (~1562 €). Lowest average prices: Acer (~627 €), Chuwi (~314 €), Mediacom (~295 €).

**Laptop Type** - Workstations (~2158 €) and Gaming laptops (~1669 €) were premium categories. Notebooks (~779 €) and Netbooks (~636 €) were budget.

**Operating System** - (~1747 €) and Windows (~1622 €) were premium. Linux (~617 €) and Chrome OS (~554 €) were budget.

📊 **Key Influencers**: RAM, Brand, Laptop Type, and Operating System.

**Feature Engineering**

To enhance performance, new features were engineered:

**Log\_Price**: log transformation of price (to reduce skewness).

**Screen\_Area**: estimated from diagonal inches.

**Price\_per\_GB**: price normalized by RAM.

**Portability Index**: weight ÷ screen size.

**Is\_Touchscreen**: binary flag for touchscreen laptops.

**OS\_Category**: grouped OS.

**Is\_Gaming and Is\_Workstation**: binary indicators.

These features improved interpretability and representation of hidden patterns.

**Model Building**

The dataset was split into 80% training and 20% testing.

**Linear Regression (Baseline Model)** :

* Initially chosen for simplicity.
* Produced **negative predictions** in some cases.
* Evaluation - R² ≈ 0.95 but unrealistic price outputs.
* Conclusion - Not suitable for deployment.

**Random Forest Regressor (Final Model):**

* Parameters - 100 estimators, random\_state=42.
* Performance –
* RMSE = 3.52
* MAE = 1.84
* R² = 1.00.
* Training R² = 0.9999 (very high).
* Predictions were **positive and realistic**.
* Selected as **best-performing model** for deployment.

**Model Evaluation**

The final Random Forest model outperformed Linear Regression in all metrics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **RMSE** | **MAE** | **R^2** | **Notes** |
| Linear Regression | 133.17 | 95.83 | 0.95 | Produced negative prices |
| Random Forest | 3.52 | 1.84 | 1.00 | Robust |

**Version Control (GitHub)**

Version control was managed using GitHub:

A project repository was created to store the dataset, notebook, model files, requirements, and app.py. Files were uploaded and managed within the repository, ensuring accessibility and centralization of the project resources. Collaborators were added to allow team-based work and visibility.

Uploading of files was only made from the shared Github group account(Group-J-CPE221). The account served as central repository managed by all group members, ensuring every member had access to the project files and updates. Although we did not commit every step individually, the repository still contains a clear commit history reflecting multiple updates and modifications. In addition, the final Jupyter Notebook of both models – developed collaboratively by the group – was uploaded as a comprehensive version of our work. This ensured complete, polished analysis and model building process was preserved and version-controlled, even if not every intermediate step was committed.

**Deployment (Streamlit App)**

A Streamlit Cloud web app was developed to allow real-time predictions.

**Features**:

* Users input specifications (Brand, RAM, OS, etc.)
* Model predicts price instantly,
* Displayed result includes the Euro (€) symbol for clarity.

**Deploymen**t:

* Files uploaded - app.py, random\_forest\_regression\_model.pkl, rf\_model\_columns.pkl, requirements.txt
* App deployed on Streamlit Cloud
* Hosted link provided for evaluation - https://laptoppriceprediction-wtjq6qfyt8x8wcozgrmcsm.streamlit.app

**How to Use the Deployed App:**

The Streamlit application was deployed directly using Streamlit Cloud, which hosts the app online. This means the app does not run locally from the Jupyter Notebook or terminal; instead, it is accessible via a browser link. This approach ensures easy access and avoids the need for users to install Python or project dependencies on their own systems.

To run the deployed app please visit the streamlit cloud link - <https://laptoppriceprediction-wtjq6qfyt8x8wcozgrmcsm.streamlit.app>

The Streamlit application provides a simple interface for predicting laptop prices:

* Open the app in a browser via the following link: [Insert Streamlit App URL].
* Enter the laptop specifications (brand, RAM, screen size, operating system, etc.) into the provided fields.
* Click the “Predict” button.
* The estimated price will be displayed in Euros (€).

The interface is designed to be intuitive, requiring no prior technical knowledge.

**System Requirements and Recommended Settings:**

The deployed app is hosted on Streamlit Cloud, meaning users do not need to install Python or any additional libraries. To ensure smooth running:

* Device: Laptop, desktop, or mobile device with internet access.
* Browser: Latest version of Google Chrome, Mozilla Firefox, or Microsoft Edge.
* Internet: Stable connection (minimum 1 Mbps recommended).

No local setup is required; users simply access the app via the hosted link.

**Discussion & Interpretation**

Findings confirmed that **hardware specifications (RAM, CPU, GPU), brand reputation, and intended use (gaming/workstation)** drive laptop prices.

* RAM was the single strongest predictor.
* Premium brands (Apple, Razer) priced higher consistently.
* Linear Regression was interpretable but unsuitable (negative outputs).
* Random Forest gave robust, realistic predictions → best model for deployment.

**Limitations:**

* Dataset restricted to European pricing.
* Random Forest can overfit with small datasets.

**Conclusion**

This project demonstrated the **end-to-end ML pipeline**:

* Data cleaning & preprocessing ensured model compatibility.
* EDA revealed major price drivers.
* Feature engineering added interpretability.
* Two models were built: **Linear Regression (baseline) and Random Forest (final).**
* Random Forest achieved near-perfect predictive performance and was selected for deployment.
* Streamlit Cloud enabled real-time, user-friendly predictions.

**Future improvements:**

* Explore other tree-based models (e.g., XGBoost).
* Use larger and more recent datasets.
* Enhance Streamlit UI with charts and comparisons.

**References**

**Kaggle Dataset**: muhammetvarl – Laptop Price Dataset

**Libraries:** Pandas, Numpy, Matplotlib, Seaborn, Scikit-learn, Streamlit

**Documentation**: scikit-learn.org, streamlit.io