En este repositorio, compartimos el código en python de un modelo de Machine Learning de predicción de precios de ordenadores portátiles. Este trabajo parte desde un dataset sin organizar, con la mayor parte de la información en una sola columna. Por lo tanto, se puede apreciar el trabajo de limpieza, EDA y feature engineering (o ingeniería de variables), el entrenamiento de diferentes modelos con pipeline, la búsqueda de los mejores hiperparámetros con Cross Validation (o validación cruzada) y toda la lógica utilizada para el desarrollo de este modelo.

# 📌 Predicción de precios de ordenadores portátiles 💻

Este proyecto desarrolla un modelo de Machine Learning para predecir el precio de un portátil a partir de sus características técnicas (RAM, CPU, GPU, almacenamiento, peso, etc.).

El trabajo se ha realizado como proyecto final del bootcamp de Data, aplicando todo el pipeline completo:

* Adquisición de datos.
* Limpieza y análisis exploratorio (EDA).
* Feature engineering.
* Entrenamiento y evaluación de modelos.
* Interpretación de resultados.
* Despliegue en una aplicación web con Streamlit.

## 🔎 Dataset

* Fuente: el dataset utilizado es un csv fruto de la extracción, a través de web scrapping, de más de 1.700 ejemplos reales de ordenadores en Amazon.
* Target: Price (precio del portátil).
* Variables principales:
* Brand: marca del portátil.
* CPU / GPU: procesador y tarjeta gráfica.
* RAM, Storage, Weight: características técnicas.
* Rating: puntuación de usuarios.
* OS: Sistema operativo

## ⚙️ Metodología

1. EDA y limpieza  
   * Tratamiento de nulos.
   * Normalización de texto (CPU, GPU).
   * Codificación de variables categóricas.
   * Visualizaciones: distribución de precios, boxplots por marca, correlaciones.
2. Modelado  
   * Modelos probados:  
     + Regresión Lineal
     + Ridge / Lasso
     + Random Forest
     + Gradient Boosting (incluyendo variante con pérdida Huber y target log-transformado)
     + XGBoost
     + KMeans (no supervisado, clustering de portátiles)
   * Selección de hiperparámetros con GridSearchCV.
3. Evaluación  
   * Métricas: MAE, RMSE, R².
   * Gráficas: real vs predicho, distribución de errores, curva de aprendizaje.
   * Importancia de variables.
4. Modelo final  
   * Guardado en models/final\_model.pkl. El modelo final elegido ha sido RandomForest

## 🚀 App en Streamlit

Se ha creado una aplicación en Streamlit para probar el modelo fácilmente:

### Ejecución local

1. Instalar dependencias:  
     
    pip install -r app\_streamlit/requirements.txt
2. Ejecutar la app:  
     
    streamlit run app\_streamlit/app.py
3. Subir un CSV con las mismas columnas que el dataset de entrenamiento.
4. La app muestra:  
   * Tabla con predicciones.
   * KPIs (precio medio, min, max).
   * Gráficas (distribución de predicciones, comparación real vs predicho).
   * Botón para descargar resultados.

## 📊 Resultados principales

* El modelo final obtuvo un MAE ≈ 264 € y un R2 de 0.746
* Variables más importantes en la predicción: Brand, CPU y RAM, CPU
* Buen equilibrio entre interpretabilidad y rendimiento.

## 📌 Conclusiones y mejoras

* La predicción de precios de portátiles es viable y útil para e-commerce o comparadores.
* El despliegue con Streamlit permite usar el modelo de manera sencilla.
* Mejoras posibles:  
  + Ampliar dataset con más productos y más actualizados. También sería interesante utilizar el modelo con datasets de diferentes momentos del año para hacer una comparativa de cada época (verano, rebajas, navidad…)
  + Explorar modelos más avanzados (LightGBM, CatBoost).