Taller 2 - (@October 17, 2024) - ML Training

- *Juan Andrés Ruiz*
- *Juan Sebástian Ortiz*

Contexto del Problema

En un Marketplace en línea, es crucial identificar si un producto listado es **nuevo** o **usado** para optimizar las recomendaciones y mejorar la experiencia del usuario. El objetivo de este proyecto es construir un modelo de machine learning que permita predecir la condición de los productos listados basándose en un conjunto de características del producto. El conjunto de datos utilizado se llama `MLA_100k.jsonlines`, y contiene diversas características del producto, vendedor, imágenes, entre otros.

---

1. Análisis Exploratorio de Datos (EDA)

Carga y Limpieza de Datos

En el análisis exploratorio (EDA), se realizaron las siguientes tareas:

- **Carga de datos**: El archivo `MLA_100k.jsonlines` fue cargado y procesado para transformar sus datos a un formato adecuado para el entrenamiento del modelo.
- **Desempaquetado de datos JSON**: Varias columnas en formato JSON o listas, fueron transformadas para extraer características adicionales.
- **Eliminación de valores nulos y columnas irrelevantes**: Se eliminaron las columnas con más del 80% de valores nulos o con un solo valor único.
- **Creación de nuevas características**: A partir de los datos originales, se generaron columnas.

Estas tareas se realizaron en el notebook **001_EDA.ipynb** y el conjunto de datos final fue guardado como `MLA_100k_cleaned.csv` para utilizarse en el proceso de modelado.

▼ Eliminación de columnas

▼ IDs

- site_id: solamente tiene un valor que es MLA, no contiene tanta información. Se elimina.
- seller_id: id del vendedor. No suma al ML
- id: id del producto. Código alfanumerico interno sin ningún patron aparente. Se elimina
- official_store_id: Código alfanumerico interno sin ningún patron aparente. Se elimina
- currency_id: Código alfanumerico interno sin ningún patron aparente. Se elimina
- video_id: Código alfanumerico interno sin ningún patron aparente. Se elimina
- catalog_product_id: Código alfanumerico interno sin ningún patron aparente. Se elimina
- site_id: Código alfanumerico interno sin ningún patron aparente. Se elimina.
- parent_item_id: Código alfanumerico interno sin ningún patron aparente. Se elimina.
- category_id: Código alfanumerico interno sin ningún patron aparente. Se elimina.

▼ Location and details

- seller_address: Solamente tiene valor AR, lo que significa que los datos son provenientes de argentina o son nulos. Por ende sabemos que son datos de argentina y se elimina la columna.
- geolocalitation. La localización no suele importar, pues ya sabemos que es un analisis de argentina en general
- seller_contact: no es tan importante los datos de contacto del vendedor

▼ Links

- permalink: es un link del producto. No suele ser tan importante en este caso, pues solo es el link de la imagen.
- thumbnail: es un link de la miniatura del producto. No suma. Se elimina

▼ Details (no importants)

- shipping: Se va a extraer de la columna shipping el local pick up y el free_shipping, y se eliminará shipping, pues la información que vamos a extraer son las anteriores
- description: son una clase de códigos sin alguna referencia en especial.
- subtitle: Código alfanumerico interno sin ningún patron aparente. Se elimina
- differential_pricing: Todos los registros son nulos. Es decir, la columna no sirve para nada.
- base_price: se tiene la columna price. Por ende se borra.
- original_price: se tiene la columna price. Por ende se borra.
- warranty: Si bien la columna puede generar un impacto y sería interesante, son más de 10 mil valores unicos que tiene la columna, por lo que detectar la garantía en estas podría requerir de un mapeo muy robusto, y si le adicionamos que solamente tiene 38 mil datos no nulos, se considera eliminar esta columna.

---

2. Proceso de Entrenamiento del Modelo

Selección de Características

Para entrenar el modelo, se seleccionaron características clave que proporcionan información importante sobre los listados. Estas características incluyen, entre otras:

- `base_price`
- `price`
- `non_mercado_pago_payment_methods`

- `seller_id`
- `initial_quantity`
- `sold_quantity`
- `available_quantity`

La **variable objetivo** fue la columna `condition`, transformada a una variable binaria, donde 1 representa productos **nuevos** y 0 productos **usados**.

División de Datos

El conjunto de datos fue dividido en un 80% para entrenamiento y un 20% para prueba, utilizando una semilla aleatoria (`random_state=42`) para asegurar la reproducibilidad. Esta partición garantiza que los datos de prueba no se utilicen en el entrenamiento y nos permitan evaluar la capacidad de generalización del modelo.

---

3. Modelos de Machine Learning Utilizados

En el notebook **002_model_training.ipynb**, se entrenaron los siguientes cinco modelos:

1. **Regresión Logística** (`LogisticRegression`)
2. **MLP (Perceptrón Multicapa)** (`MLPClassifier`)
3. **Árbol de Decisión** (`DecisionTreeClassifier`)
4. **Random Forest** (`RandomForestClassifier`)
5. **XGBoost** (`XGBClassifier`)

Evaluación y Métricas

Cada modelo fue evaluado utilizando la siguiente métrica:

- **Exactitud (Accuracy)**

---

4. Evaluación de Modelos y Resultados

A continuación, se presentan los resultados obtenidos tras la evaluación de cada modelo en el conjunto de datos de prueba:

- **Logistic Regression**: Accuracy = 71.50%

- **MLP**: Accuracy = 71.74%
- **Decision Tree**: Accuracy = 81.60%
- **Random Forest**: Accuracy = 82.27%
- **XGBoost**: Accuracy = 83.09%

Mejor Modelo: XGBoost

El modelo **XGBoost** fue el que obtuvo el mejor rendimiento general, con una **precisión del 83.09%**, lo cual indica que es el más adecuado para este problema.

---

5. Conclusiones Finales

- **XGBoost** fue seleccionado como el modelo final, ya que presentó el mejor rendimiento en términos de accuracy.
- Este modelo es adecuado para implementar en producción, ya que generaliza bien y permite predecir con alta precisión si un producto es nuevo o usado.
- El modelo fue guardado como un archivo **.pkl** para su uso posterior en despliegue.
- Este proceso ha permitido construir un modelo robusto que puede ser desplegado para mejorar la experiencia de los usuarios en un Marketplace prediciendo con precisión si un producto es nuevo o usado.

Archivos Adjuntos:

- **001_EDA.ipynb**: Contiene el análisis exploratorio de datos y preprocesamiento.
- **002_model_training.ipynb**: Incluye el entrenamiento, evaluación de `features` y selección del modelo.
- **model.pkl**: Archivo con el modelo final entrenado listo para ser implementado.