**1. Caso de negocio**

Un gran problema en Mercado Libre es que muchos productos idénticos o muy similares son vendidos por distintos sellers, lo que hace que los compradores tengan dificultades para comparar opciones y tomar decisiones de compra. **La variedad de sellers en la plataforma genera duplicaciones que complican la comparación de productos, afectando la experiencia del usuario. Además, para Mercado Libre, esto representa un desafío en la gestión del inventario y la optimización de la oferta.**

Para solucionar esto, implementé un enfoque basado en **Procesamiento de Lenguaje Natural (NLP)**. Primero, utilicé **embeddings de texto** para convertir los títulos de los productos en representaciones numéricas (vectores), permitiendo medir similitudes entre ellos. Luego, entrené un modelo de clasificación **(Random Forest)** para predecir la categoría de cada producto y, dentro de esa misma categoría, apliqué **FAISS un metódo similar a KNN** para encontrar productos similares. Esto permite agrupar artículos comparables, facilitando la navegación y mejorando la experiencia del usuario al mostrarle opciones relevantes en el marketplace.

**2. Casos Previos Considerados**

**Caso 1: Optimización de Inventario**

En un primer intento, se buscó desarrollar un modelo para sugerir la cantidad óptima de inventario a almacenar para cada producto. Para ello, se necesitaban datos históricos de:

* Cantidad de inventario inicial.
* Ventas por product (histórico).
* Inventario actual (no en rangos).

**Problema encontrado:** La API de Mercado Libre no proporcionaba acceso (permisos) a datos históricos de inventario y ventas, lo que imposibilitó el desarrollo de la solución.

**Caso 2: Clasificación de Sellers**

El equipo comercial quería segmentar a los sellers según su relevancia para el negocio con el fin de implementar estrategias focalizadas.

**Problema encontrado:** La falta de acceso ( permisos) a datos clave sobre el comportamiento de ventas y engagement de los sellers impidió la implementación del modelo.

**3. Solución Seleccionada: Agrupación de Productos Similares**

Ante las limitaciones en los casos previos, se decidió abordar el problema de la similitud de productos dentro del marketplace.

**Objetivo:** Desarrollar un modelo de clasificación que identifique productos similares o idénticos a partir de los títulos y embeddings de texto.

**4. Solución Seleccionada: Agrupación de Productos Similares**

Ante las limitaciones en los casos previos, se decidió abordar el problema de la similitud de productos dentro del marketplace.

Objetivo: Desarrollar un modelo de clasificación que identifique productos similares o idénticos a partir de los títulos y embeddings de texto y muestre su comparación

Se asumió que los títulos de los productos contienen información suficiente para identificar similitudes entre ellos. También se supuso que los embeddings de texto permitirían capturar relaciones semánticas más profundas que solo la clasificación por categoría.

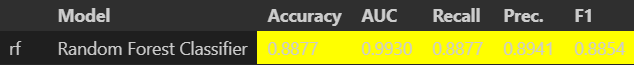
**4.1. Metodología Aplicada**

1. Extracción de Datos: Se extrajeron los datos de productos a través de la API de Mercado Libre. Se usó el código **Extracción de datos.** 
   * Conexión con la API.
   * Obtención de Productos por Categoría.
   * Filtrado y Organización de Datos.
   * Exportacion de los Datos.
2. Preprocesamiento: Se limpiaron y normalizaron los títulos de los productos. Se usó el código **EDA**.
   * Selección de Columnas Relevantes.
   * Manejo de Valores Nulos.
   * Normalización de Texto.
   * Filtrado de Registros.
   * Almacenamiento del DataFrame Procesado.

Insights:

* + Número total de categorías: 32 categorías principales en la base de datos, con un total de 2093 subcategorías, lo que permitió trabajar a un nivel más detallado.
  + El promedio de productos por subcategoría es 15.
  + Hay varias subcategorías que aconglomeran un gran porcentaje de arículos, mientras que otras subcategorías no tienen muchos de ellos.
  + El 93.44% de los productos son nuevos, mientras que solo el 6.56 de ellos son usados.

1. Generación de Embeddings: Se utilizaron embeddings de texto para convertir los títulos en representaciones numéricas. Se usó el código **Creación de embeddings.**
   * Selección de Columnas Relevantes.
   * Generación de Embeddings con BERT.
   * Normalización de Variables Numéricas.
   * Creación de un Índice de Búsqueda Rápida con FAISS.
   * Almacenamiento de Datos Procesados.
2. Entrenamiento y evaluación del Modelo: Se entrenó un modelo de clasificación para detectar similitudes entre productos. Se usó el código **Modelo predictivo de categoría.**
   * Carga y Preprocesamiento de Datos
   * Se filtraron solo las categorías con al menos 50 productos para asegurar representatividad
   * Se seleccionaron las columnas **title** y **category\_id**, eliminando caracteres especiales del título
   * Vectorización del Texto con TF-IDF
   * Entrenamiento del Modelo con PyCaret (80% entrenamiento, 20% Prueba)
   * Se compararon múltiples algoritmos de clasificación, excluyendo algunos modelos menos eficientes. Y se seleccionó el mejor modelo basado en su rendimiento en validación cruzada, siendo el Random Forest el mejor evaluado.



* + Evaluación y Guardado del Modelo y del vectorizador.

El modelo entrenado permite predecir la categoría de un producto basándose en su título con un enfoque basado en aprendizaje automático y procesamiento de lenguaje natural

1. Búsqueda de productos similares y comparación de atributos clave: Esto con el objetivo de mejorar la experiencia de usuario luego de encontrar productos muy similares. **Se uso el código Búsqueda de productos similares**
   * **Se recibe el producto por el usuario, sea palabra o frase,** se convierte en su representación TF-IDF, y se predice su categoría.
   * Se extrae los artículos de la categoría predicha, para asegurar que la comparación sea relevante.
   * Vectorización y búsqueda en FAISS para sacar un top 50 de productos similares.
   * Teniendo el conjunto de productos similares , se aplican criterios adicionales para que la comparación sea mas específica y refinada.
   * Tomamos en cuenta variables que hasta acá no se habían tocado como condition, listing\_type\_id.
   * Se tomó estas otras variable como original\_price, available\_quantity, free\_shipping, seller, installments, thumbnail y permalink, para que el usuario tenga la posibilidad de visualizar las diferencias.

**Este paso final permite no solo encontrar productos parecidos, sino también facilitar la compra con información clave como precio, envío, descuentos y vendedor.**

1. **Conclusiones de la Solución:**
   * Mejora en la Experiencia del Usuario:  
     Al proporcionar comparaciones detalladas entre productos similares, los compradores pueden tomar decisiones informadas de manera más rápida y sencilla. Esto reduce la fricción en el proceso de compra, ya que los usuarios no necesitan explorar múltiples listados sin referencia clara de sus diferencias. La agrupación de productos por similitud y características adicionales como precio, estado y seller hace que la experiencia sea más fluida y eficiente.
   * Optimización del Marketplace para Mercado Libre  
     Desde la perspectiva del negocio, la solución permite una mejor organización del catálogo de productos, reduciendo la redundancia de artículos similares y optimizando la oferta. Esto no solo mejora la conversión al facilitar la búsqueda, sino que también permite una gestión más eficiente del inventario y la publicidad en la plataforma. Al ofrecer recomendaciones basadas en similitud, se puede incentivar la compra de productos con mejores precios o condiciones más atractivas.
   * Para las subcategorías más pequeñas, los modelos tienden a cometer más errores debido a la falta de datos representativos. Esto podría evidenciar la necesidad de desarrollar un modelo específico para estas categorías, aplicando técnicas de balanceo o enfoques más flexibles que mejoren la precisión.
2. Próximos pasos
   * **Optimización del espacio vectorial**: Explorar modelos de NLP más avanzados de BERT o GPT entre otros (especializados en markets) que puedan capturar mejor el significado de los títulos y mejorar la agrupación de productos similares. También se pueden probar técnicas de fine-tuning con datos específicos del marketplace.
   * **Desarrollo de una API y una interfaz amigable**: Implementar una API que permita a los usuarios realizar búsquedas y recibir recomendaciones de productos similares en tiempo real. Además, diseñar una interfaz intuitiva y atractiva que facilite la visualización de los resultados sin depender de tablas en código. Algo como una herramienta que se sobre exponga en la página de Mercado Libre, y sirva para hacer las comparaciones de productos).
   * **Mejorar la clasificación en subcategorías pequeñas**: Implementar estrategias para mejorar la predicción en categorías con pocos datos, como técnicas de data augmentation, o mirar si se pueden agrupar varias de esas pequeñas categorías en una sola, y aumentar la precisión de los modelos.
   * **Incluir más características en la recomendación**: Debido al poco acceso de información que se tenía, pero existe la posibilidad de incorporar información adicional, como historial de compras, comportamiento de los usuarios o descripciones de productos, para enriquecer las recomendaciones y hacerlas más precisas.
   * **Evaluación en un entorno de producción**: Probar la solución en un entorno real para medir su impacto en la experiencia del usuario y optimizarla en función del feedback y métricas de uso.