Carlos Arturo Ávila García
Prueba técnica – Científico de datos II

Primera parte: preguntas abiertas

1. **Explica cómo implementarías un modelo de machine learning que debe ser explicable y auditable en un entorno regulado (ej. sector financiero o salud).**

En primer lugar, es necesario tener en cuenta que deben aplicarse modelos con un número limitado de variables, cuyo origen, transformaciones y arreglos sean totalmente verificables y documentados. El repositorio deberá estar bien estructurado de manera que los datos en crudo y transformados, se trabajen de manera separada. Así mismo, se debe tener en cuenta que se deben utilizar modelos fácilmente interpretables que permitan la socialización de resultados para toda clase de audiencias; sin embargo, en caso de requerir la aplicación de modelos más complejos de explicar, se deberá recurrir a técnicas de interpretabilidad donde se intenta linealizar a partir de técnica como valores SHAP (SHapley Additive exPlanations). En segundo lugar, en línea con la trazabilidad y mantenimiento de la memoria institucional, se deben mantener registros detallados de decisiones, datos de entrenamiento y pruebas; además de un registro de las justificaciones en cada paso del modelado. Documentar el flujo de datos y las transformaciones para asegurar trazabilidad en auditorías. Finalmente, se debe tener en cuenta preservar la anonimidad de la información asegurando que no haya fuga de datos personales y sensibles de la población objeto del análisis. De esta manera, es recomendable no guardar los datos en computadores de uso personal, haciendo uso de conexiones seguras a proveedores de servicios en la nube.

2. **¿Cómo diseñarías un experimento de A/B Testing para evaluar el impacto de un nuevo modelo de machine learning en producción?**

Procedería a dividir aleatoriamente el conjunto de datos en dos (A y B), tratando de mantener balanceadas las características de los datos entre los dos grupos, evitando de esta manera problemas asociados con sesgos en la muestra. El modelo que se encuentra en producción será entrenado con el grupo de datos A y con el subconjunto de datos B se evaluará el rendimiento del nuevo modelo. Buscando comparar qué tan buenos son los modelos, se deberán utilizar métricas clave como precisión, recall, sensibilidad, F1-score, entre otros. Posteriormente se podrían comparar directamente a partir de un análisis estadístico (ej. t-test) para comparar las métricas entre ambos grupos y asegurar que las diferencias observadas sean significativas. Así mismo, teniendo en cuenta que se debe observar el rendimiento en el mediano y largo plazo, será necesario extender el análisis a medida que se nutre la base de datos con nuevos registros para ver el comportamiento a largo del tiempo.

3. **En un modelo basado en redes neuronales profundas, ¿cómo manejarías el problema de vanishing/exploding gradients y qué soluciones aplicarías?**

Los problemas de gradientes que explotan o desaparecen se presentan sobre todo en modelos de redes neuronales con una gran cantidad de capas. Estos problemas guardan relación con las funciones de activación que se utilizan, los valores de los pesos que son utilizados en la retropropagación y son evidentes cuando el modelo converge muy lento a la solución (vanishing) o tiene cambios abruptos en la función de pérdida a lo largo del

entrenamiento (exploding). Las soluciones podrían estar orientadas principalmente a la inicialización de pesos más adecuados, la normalización por lotes y el uso de funciones de activación que mantienen un comportamiento constante en valores extremos.

4. **¿Cómo diseñarías un sistema de recomendaciones en producción que pueda actualizarse dinámicamente con feedback del usuario?**

Implementar un modelo de recomendaciones que pueda actualizarse dinámicamente implica que la información que los usuarios proporcionan en términos de clicks, valoraciones y comentarios sean ingestados en tiempo real a una base de datos que servirá tanto para la evaluación como para el re entrenamiento de los algoritmos. De esta manera, a partir de las recomendaciones que ofrece el modelo y el comportamiento del usuario con base en las mismas, podrá evaluarse si las recomendaciones para un usuario específico resultan ser relevantes y genera algún tipo de enganche. A partir de estas métricas de evaluación y de los umbrales que se determinen, se deberán implementar pipelines para realizar actualizaciones periódicas del modelo.

5. **¿Cómo aplicarías modelos de machine learning en un entorno con datos altamente heterogéneos y no estructurados (ej. imágenes, texto y datos tabulares)?**

Es necesario tener en cuenta que el tipo de datos con el que se procederá a implementar modelos de machine learning determinará el tipo de tratamiento que se le pueda dar a la información. Por el ejemplo, si se hablar de imágenes se tendrá que procesar y estandarizar la información a nivel de pixeles, mientras que, si los datos son de tipo texto, el procesamiento de información deberán tener en cuenta las técnicas de limpieza de texto asociadas con NLP y la representación de la información en embeddings y vectorización de la misma. Los modelos de aprendizaje profundo son altamente recomendados para este tipo de datos; sin embargo, habrá que tener en cuenta técnicas de análisis que en el caso de texto implican, pero no se limita, al análisis de sentimiento, de discurso, entidades nombradas, etc.

6. **¿Cuáles son las ventajas y desventajas de desplegar un modelo de machine learning como un microservicio en comparación con integrarlo directamente en una aplicación monolítica?**

El diseño de aplicaciones a través de microservicios se caracteriza por estar compuesto por pequeños servicios independientes que permiten que su escalabilidad sea más fácil y su desarrollo notoriamente más rápido que los desarrollos monolíticos. Así, mismo su implementación es sencilla, permite libertad en el uso de tecnologías, el código puede ser reutilizable en otros procesos, ya que ejecuta tareas muy definidas que pueden servir para más de un proyecto y es más resistente a los errores, debido a que si ocurre un error en el sistema se va degradando la funcionalidad, pero no se cae el servicio por completo. Las desventajas asociadas a este tipo de arquitecturas se relacionan con su complejidad a la hora de orquestar procesos y la consecuente sobrecarga de comunicación entre servicios.

Por otra parte, las ventajas de aplicaciones monolíticas subyacen en su simplicidad y la menor latencia de comunicación interna; sin embargo, a diferencia de los microservicios,

su proceso de escalado es más complejo, existe menos flexibilidad a la hora de desplegar actualizaciones y son altamente más sensibles ante los fallos (es probable que solucionar fallos tome más tiempo debido a que no se sabe directamente donde están ocurriendo los errores).

7. **¿Cómo manejarías la monitorización y logging de un modelo en producción para detectar degradaciones en su rendimiento?**

Se puede hacer uso de plataformas para manejar flujos de trabajo en machine learning, como MLflow donde se puede dar seguimiento a los resultados de la implementación de modelos, donde se puede programar la ejecución de los algoritmos a través del tiempo y en términos de monitorización y logging, se pueden programar alertas que se envíen a los usuarios basadas en umbrales que se definan para las métricas de evaluación y rendimiento de los modelos.

8. **Explica cómo diseñarías una arquitectura de CI/CD para modelos de machine learning en producción.**

La arquitectura de CI/CD se caracterizaría por el uso de herramientas apropiadas para el versionamiento de código como Git y del uso de plataformas para automatizar el pipeline de desarrollo y despliegue como MLflow. La arquitectura, también debe incluir pasos de validación de modelos (tests unitarios, integración, evaluación de rendimiento), entrenamiento en un entorno controlado y despliegue automatizado en producción.

9. **¿Cómo optimizarías el tiempo de inferencia de un modelo en producción sin comprometer su precisión?**

Existen algunas intervenciones que pueden realizarse desde el back-end, como la optimización del código para la ingesta de datos (haciéndolos más ligeros), hacer uso de infraestructuras robustas que respondan a las peticiones que se esperan el servidor debe soportar, así como el uso de soluciones de proveedores en la nube que estén diseñadas para optimizar el tiempo de respuesta. Adicionalmente, se podrían utilizar versiones “ligeras” de modelos más grandes en caso de que existan y siempre existe la opción de mejorar las especificaciones de hardware con el uso de aceleradores.

10. **Explica cómo manejarías la orquestación de workflows de ML en la nube y qué herramientas utilizarías.**

Hacer uso de herramientas como Airflow o MLflow para orquestar los pipelines de los modelos de machine learning en la nube. Estas plataformas permiten integrar tareas de preprocesamiento, entrenamiento, validación y despliegue en un solo flujo de trabajo automatizado, gestionando dependencias, y realizando monitorización en tiempo real. Ahora bien, se podría hacer uso de servicios en la nube que faciliten la creación, el entrenamiento e implementación de modelos a gran escala de algunos proveedores como AWS o Google Cloud Platform, que incluyen herramientas como SageMaker o VertexAI.

Carlos Arturo Ávila García
Prueba técnica – Científico de datos II

Segunda parte: Desarrollo de un Motor de Recomendaciones

En este reporte se presenta el desarrollo y evaluación de un motor de recomendaciones. El objetivo principal es generar recomendaciones personalizadas para los usuarios, combinando la información de interacciones previas con productos y características intrínsecas de los mismos.

Conjunto de Datos

Para la construcción del modelo, se utilizaron tres bases de datos principales:

- **users.csv**: Contiene información sobre los usuarios de la plataforma.
- **products.csv**: Incluye características de los productos disponibles para recomendación.
- **interactions.csv**: Registra interacciones entre los usuarios y los productos, lo que permite inferir preferencias y patrones de comportamiento.

Análisis Exploratorio de Datos (EDA)

Previo al entrenamiento del modelo, se realizó un análisis exploratorio de los datos (EDA), el cual incluyó:

- Identificación de valores faltantes y su tratamiento.
- Análisis de la distribución de interacciones entre usuarios y productos.
- Inspección de estadísticas generales para comprender la densidad de la matriz de interacciones.
- Verificación de sesgos en los datos, como productos con muchas o muy pocas interacciones.

La distribución de las variables categóricas en cada una de las bases de datos presenta una distribución balanceada, como se muestra en la imagen a continuación. Las variables categóricas en la base de datos de usuarios son: genero, nivel_ingresos, nivel_educativo, tipo_suscripcion, categoria_cliente, ubicacion, dispositivo, frecuencia_login. Por otra parte, las variables categóricas de la

Carlos Arturo Ávila García
Prueba técnica – Científico de datos II

Gráfico 1. Base de datos de usuarios

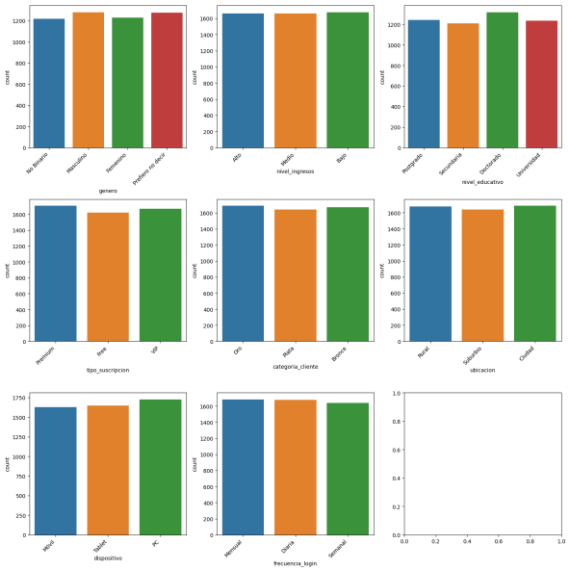
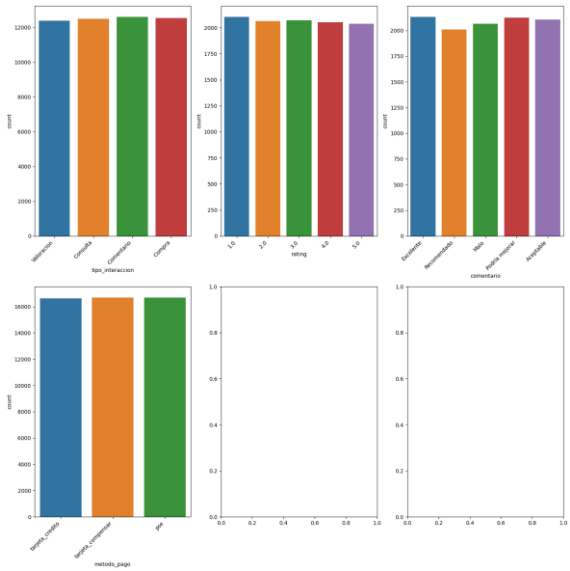


Gráfico 2. Base de datos de interacciones



Este análisis permitió detectar que la matriz usuario-producto es dispersa, es decir, la mayoría de los usuarios han interactuado solo con una fracción pequeña del catálogo disponible. Esto motivó el uso de un modelo de descomposición matricial para mejorar la calidad de las predicciones.

Metodología: Enfoque Híbrido

Se optó por un enfoque híbrido combinando dos técnicas fundamentales:

Filtrado Colaborativo

El modelo de filtrado colaborativo se basa en la descomposición matricial utilizando **Singular Value Decomposition (SVD)**. Este modelo aprovecha patrones de interacción entre usuarios y productos para predecir futuras preferencias. Su implementación sigue los siguientes pasos:

1. Construcción de la matriz usuario-producto basada en interacciones previas.
2. Aplicación de SVD para reducir la dimensionalidad y extraer representaciones latentes de usuarios y productos.
3. Predicción de calificaciones futuras mediante la combinación de estas representaciones.

Filtrado Basado en Contenido

Este método se apoya en la similitud de los productos utilizando **similitud coseno** sobre las características extraídas del conjunto de datos products.csv. El proceso implica:

1. Transformación de los atributos del producto en un espacio vectorial.
2. Cálculo de la matriz de similitud coseno entre todos los productos.

3. Generación de recomendaciones basadas en productos similares a aquellos que el usuario ha preferido previamente.

Combinación de Métodos

El enfoque híbrido combina los resultados de ambos métodos mediante una estrategia ponderada, en la que se promedian las puntuaciones de recomendación obtenidas por cada técnica. De esta manera, se busca complementar las fortalezas del filtrado colaborativo (capacidad de identificar relaciones implícitas) con las del filtrado basado en contenido (capacidad de hacer recomendaciones incluso para productos poco interactuados).

Evaluación del Modelo

Para evaluar la efectividad del sistema de recomendación, se emplearon diferentes métricas:

- **Error Cuadrático Medio (RMSE)**: Se utilizó para medir la precisión de las predicciones de calificación en el modelo de filtrado colaborativo.

Análisis de Rendimiento y Posibles Mejoras

Si bien el modelo presentó un buen desempeño, existen varias oportunidades de optimización:

- **Optimización del tiempo de inferencia**: Precomputar matrices de similitud y almacenar representaciones latentes para reducir la carga computacional durante la generación de recomendaciones.
- **Reducción de dimensionalidad**: Experimentar con técnicas avanzadas como **Autoencoders** o **Factorización de Matrices Regularizada** para mejorar la precisión sin comprometer el rendimiento.
- **Hiperparámetros del modelo SVD**: Ajustar valores como el número de factores latentes y la regularización para optimizar el desempeño.
- **Incorporación de aprendizaje profundo**: Explorar modelos de redes neuronales para mejorar el filtrado basado en contenido, especialmente en casos de productos con descripciones más complejas.

Conclusiones

Este estudio demuestra que un enfoque híbrido mejora significativamente la calidad de las recomendaciones al aprovechar la complementariedad de métodos colaborativos y basados en contenido. La combinación de SVD y similitud coseno permitió construir un sistema robusto, capaz de generar recomendaciones relevantes y precisas. No obstante, futuras mejoras en optimización computacional y técnicas avanzadas podrían incrementar aún más la efectividad del sistema.

Repositorio del proyecto

Todo el proyecto se encuentra alojado en un repositorio público de github en la siguiente url https://github.com/caravilaga/Recommender_app.git

Carlos Arturo Ávila García
Prueba técnica – Científico de datos II

Tecnología y arquitectura del sistema

El repositorio está constituido por la siguiente estructura de archivos:

```
Recommender_app
├— Dockerfile
├— App.py
├— Datos
│   ├— products.csv
├— Resultados
│   ├— sim.npy
│   ├— model.pkl
├— templates
│   ├— static
│   ├— index.html
```

Dockerfile es el archivo que me permite generar la imagen de Docker para el despliegue en local de la aplicación. El archivo **App.py**, contiene el código donde se ejecuta la aplicación basada en el framework de Flask. A partir de este script se realiza la predicción de los algoritmos previamente entrenados y almacenados en la carpeta **Resultados** (**model.pkl** y **sim.npy**). Como se evidencia en los gráficos Gráfico 3 y Gráfico 4, los resultados de la predicción arrojan información sobre el nombre del producto o servicio que se recomienda, su descripción y categoría, que provienen del archivo **products.csv** dentro de la carpeta **Datos.** Adicionalmente, el desarrollo del front-end para la visualización de los resultados se estructuró a partir de una aplicación web en Vanilla JS, cuyo archivo principal es **index.html** y en **static** se ubican los demás archivos de configuración.

Carlos Arturo Ávila García
Prueba técnica – Científico de datos II

Resultados del despliegue

Gráfico 3. Despliegue de la aplicación en local a partir de una imagen contenerizada en Docker. Recomendaciones para el usuario 4915.

Gráfico 4. Resultados del motor de recomendaciones de productos y servicios para el usuario 3584.