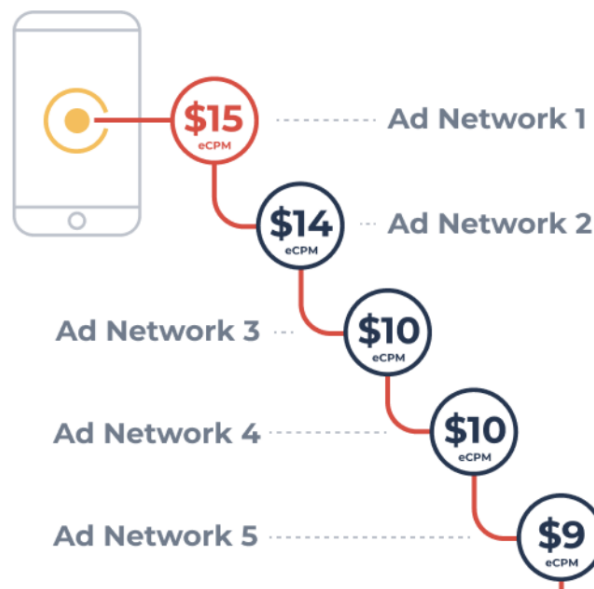


Desafío Técnico Para Machine Learning

Contexto

En el campo de adtech, una "waterfall" se refiere a un modelo de monetización de anuncios publicitarios que se utiliza para maximizar los ingresos generados por los mismos, en una plataforma digital.

En este modelo, los anuncios se intentan vender a **networks (demanda)** en un orden específico, comenzando por el ofrecimiento a mayor precio y pasando a ofrecimientos de menor precio si el anterior no es vendido. En el siguiente esquema se observa que la oportunidad comienza siendo ofrecida a 15USD a la network 1, y si esta decide no comprar la oportunidad, se pasa a la subsiguiente. Esto se repite hasta que se encuentre una network que decida comprar a un precio ofrecido, o se hayan recorrida todas las oportunidades presentes en la waterfall.



Para pasar a un lenguaje más común en la industria, a cada uno de estos ofrecimientos los pasamos a llamar **"instancias"**. Es decir, una instancia está compuesta por la tupla **(Network,**

Precio), y una waterfall es un conjunto ordenado, en forma descendente por precio, de instancias.

Al momento de que un usuario, en una plataforma digital, vea un anuncio, es de vital importancia que el mismo haya sido vendido previamente. De otro modo puede ocurrir que no se le muestre el anuncio (no se monetiza la oportunidad), o que se haga esperar al usuario in situ hasta encontrar uno (se genere una mala experiencia para este).

Para lograr lo anterior intervienen varios procesos, dentro de los cuales el obtener una estimación del tiempo que va a demorar la ejecución de la waterfall es clave. Para hacer esta estimación, previamente necesitamos estimar cual es el tiempo que nos lleva ejecutar cada instancia (ofrecer la oportunidad a una network). Con esta información luego se alimentan procesos downstream que buscan generar la mejor combinación de instancias, sujeto a una cantidad de tiempo que queremos destinar en encontrar un anuncio.

Consigna

El desafío consiste en desarrollar y disponibilizar un modelo predictivo que permita inferir el tiempo que va a demorar la ejecución de una instancia. Para esto se provee un dataset...

Los pasos a seguir son:

1. (Opcional) Realizar un EDA sobre el dataset provisto
2. **Desarrollar un modelo que permita estimar la latencia (tiempo) que va a demorar ejecutar una instancia**
3. **Disponibilizar el modelo con una API.**
4. (Opcional) Indique conceptualmente como haría un proceso de Data Quality para controlar la calidad de los datos con que se entrena al modelo
5. (Opcional) Indique conceptualmente como haría un proceso para detectar Data Shifts en los datos.
6. (Opcional) Dockerizar la API.
7. (Opcional) Incluir tests que permitan validar su funcionamiento en la API.

Consideraciones

General

- En caso de incertidumbre en alguno de los puntos, la definición queda a criterio del desarrollador.

Modelo predictivo

- Utilizar preferentemente Python. Las librerías a emplear quedan a criterio del desarrollador.
- El método de evaluación de performance del modelo queda a criterio del desarrollador.

Microservicio

- En caso de no tener preferencias, utilizar Flask.

Documentación

- La documentación a entregar, en cantidad y formato, queda a criterio del desarrollador. Mínimamente se espera encontrar la documentación necesaria para entender el proyecto y realizar un deploy en forma fluida de la API.

Datos

El dataset en cuestión es un sample de ejecución de waterfalls, compuesto por las siguiente variables:

event_id: id del evento de ejecución de waterfall.

event_time: timestamp en que se generó la waterfall.

app_id: hash de la aplicación sobre la cual se ejecutó la waterfall para vender un anuncio.

platform: plataforma del usuario.

country: país del usuario.

user_id: identificador del usuario.

adtype: formato publicitario.

waterfall_result: Lista de instancias ejecutadas hasta vender el anuncio

- **id:** identificador de la instancia.

- **partner:** Network (comprador)
- **ecpm:** precio asociado a la instancia.
- **latency:** tiempo que demoró la ejecución de esa instancia.
- **error:** código de respuesta de la waterfall.

connection_type: tipo de conexión del usuario al momento de ejecutar la waterfall.

device: información del device donde está disponible el espacio para poner el anuncio.

Notas:

- El último elemento de la lista **waterfall_result** corresponde a la instancia donde se vendió el espacio publicitario.
- La waterfall solo se recorre hasta el momento en donde se vende el espacio. De esto es que la longitud del campo varia **waterfall_result** registro a registro.
- La variable target para el modelo corresponde a **latency**. Es decir, para una waterfall, se harían N llamadas al modelo, siendo N la cantidad de instancias presentes en la misma previo a su ejecución.

¿Qué se va a evaluar?

Mediante este desafío buscamos evaluar los siguientes aspectos:

Machine learning

Buscamos ver conocimientos sólidos en desarrollo de modelos predictivos.

Desarrollo

1. Estructuración de un proyecto.
2. Buenas prácticas de desarrollo. Modularidad y Flexibilidad. Evitar código *spaghetti*.
3. Uso eficiente de recursos.
4. Criterio para selección de herramientas y bibliotecas.
5. Documentación.
 - a. Ser claro al escribir y comunicar.
 - b. Poder utilizarla para levantar el proyecto sin sufrir.
 - c. Capacidad de ayudar a entender y dar mantenimiento al sistema.

Entrega

Se espera que el proyecto sea trabajado en un repositorio git (gitlab o github). Al momento de entregar el ejercicio:

- Se envíe la URL del proyecto para poder clonarlo y levantarlo en la computadora de la persona que lo revise;
- o en caso de que el candidato no desee tenerlo en un repo público, aceptamos que nos lo envíe un zip del proyecto.