



Ejercicio: AB Testing en MercadoLibre

En MercadoLibre(MELI) navegan unos 80 millones de usuarios mensualmente, los cuáles generan un volumen de actividad y eventos enorme. Hoy vas a poder acceder a un subset del mismo.

El dataset que adjuntamos en el siguiente ejercicio, contiene porciones de la navegación de nuestros usuarios en nuestra plataforma. Cuenta con las siguiente columnas:

Nombre Columna	Descripción	Tipo
event_name	Sección del sitio navegada. Puede tomar los valores: <ul style="list-style-type: none">- SEARCH (búsqueda realizada)- PRODUCT (producto visto)- CHECKOUT_1 (visito el 1er paso del checkout)- CHECKOUT_2 (visito el 2do paso del checkout)- CHECKOUT_3 (visito el 3er paso del checkout)- BUY (compra realizada)	string
item_id	item_id que se vio	numérico
timestamp	Tiempo en el cual ocurrió dicho evento	timestamp
site	Sitio/País desde el cual el usuario navegó mercadolibre	string
experiments	Listado de experimentos a los cuales fue sometido dicho usuario en esa sección del sitio	Map<exp,variant>
user_id	Usuario que navegó el sitio en cuestión	numérico

El data set se puede encontrar en

<https://drive.google.com/file/d/1q-kVDe62HY-6SbLsetsi1s1vvYgzPUOi/view?usp=sharing>

El interés en dicho dataset radica en que describe las interacciones de nuestros usuarios previo a realizar una compra (o no) en la plataforma. En el contexto de [MELI](#), corremos cientos de [AB testings](#) para entender qué features/ideas/variantes llevan a los usuarios a comprar dentro de mercadolibre y que cosas no.

La columna experiments, tiene por objeto guardar un listado de todos los experimentos a los que fue sometido un usuario a lo largo de su navegación. Nótese que la forma en la cual se guarda implica entries/tuplas del formato *experiment_name => variant_id*. Siendo estas interpretables como “al ver esta página el usuario $\{user_id\}$ participó del siguiente experimento $\{experiment_name\}$, viendo la variante $\{variant_id\}$ ”.

Nivel 1

Implementar una query, notebook o algoritmo que pueda leer este dataset y compute para cada experimento del dataset un número de compras asociadas de forma de poder comparar cual de las n variantes de cada experimento fue mejor.



Nivel 2

Implementación de algunos checks correspondientes a la correctitud de los datos y a la confianza que se puede tener en los resultados, la naturaleza y selección de estos tests queda a discreción.

Nivel 3 (opcional)

Implementar un API en el lenguaje de preferencia que permita consultar los resultados de un experimento para un día dado. La firma HTTP por la cual se consultan los resultados de un experimento en particular debe respetar la siguiente cabecera:

GET https://host/experiment/<:id>/result?day=YYYY-MM-DD HH

Debe retornar un JSON con la siguiente estructura:

```
results: {
  exp_name : {
    number_of_participants : NNN
    winner: variant_id
    variants : [
      {
        id: 1
        number_of_purchases: 100
      },
      {
        id: 2
        number_of_purchases: 101
      }
    ]
  }
}
```

Si el experimento no existiese, una respuesta de “experimento no encontrado” (con su HTTP status correspondiente) debería ser retornada.

Nivel 4 (opcional)

Hostear dicha API en un proveedor cloud a elección y compartir los detalles de acceso a la misma para poder hacer consultas a dicha API hosteada.



ENTREGABLES:

- Código fuente de la implementación del algoritmo (En repositorio *github*).
- Readme (En repositorio *github*) que incluya:
 - Documentación de hipótesis y asunciones tomadas en el desarrollo del ejercicio.
 - Documentación importante para la interpretación de la resolución.
 - Instrucciones para utilizar y arrancar el repositorio y/o API resultante.
 - Consideraciones y tradeoffs vislumbrados a futuro para que su solución pueda ser llevada a un nivel de escala superior que el del dataset implementado.
 - URL de la API y ejemplo de consulta si se desarrolla nivel 4.