CVR Challenge

En este trabajo práctico, crearán un sistema con el propósito de **predecir la probabilidad de que un usuario** que interactúa con un anuncio específico de un destacado *retailer online* (Mercado Libre) **lleve a cabo la compra del producto anunciado**. Esto lo harán entrenando modelos de aprendizaje automático sobre la base de datos de avisos previamente visitados (algunos de los cuales fueron comprados y otros no).

El trabajo práctico se deberá resolver de a grupos de **2 integrantes**.

La **evaluación** tendrá en cuenta **3 componentes** (más sobre esto abajo): 1) el resultado del sistema en el *leaderboard* privado, 2) la calidad del informe entregado y 3) la claridad del código entregado.

Los sistemas propuestos por los diferentes grupos competirán a través de la plataforma **Kaggle**. El *link* para acceder y registrarse a la competencia se encuentra disponible en el Campus Virtual de la materia.

En el Campus también cuentan con un link para descargar un archivo comprimido que contiene los siguientes tres archivos:

- `train.csv.gz`: contiene los datos de entrenamiento. Este conjunto de datos se encuentra comprimido en formato gzip (gz). Este dataset no contiene la columna `ROW_ID`, pero sí contiene la columna `conversion`.
- `test.csv.gz`: contiene los datos de evaluación. Este conjunto de datos no se encuentra comprimido. Este dataset contiene la columna `ROW_ID`, pero no contiene la columna `conversion`.
- `sample_submission.csv.gz`: es un archivo CSV que sigue el formato del archivo que deben subir en la competencia.Este conjunto de datos se encuentra comprimido en formato gzip (gz).

En la página de la competencia, se indica qué es cada una de las **variables** que contiene el *dataset* provisto. Allí, también se indican las **reglas** de la competencia. A saber:

- La métrica de evaluación será **ROC-AUC**.
- Un 30% elegido al azar de los datos de evaluación da lugar al puntaje del leaderboard público. Este valor les servirá de guía para evaluar su desempeño, pero la **evaluación final** se realizará sobre el restante 70%. Esta evaluación final corresponde al leaderboard privado, que podrán ver una vez cerrada la competencia.
- Cada grupo podrá realizar, a lo sumo, **3 *submits* diarios** (¡no dejen todo para último momento!).

<u>Criterios de evaluación del TP</u>

1) **Performance en el leaderboard privado** (**30%** de la nota final). Las soluciones propuestas deben alcanzar una buena performance. Esto implica superar ampliamente el benchmark propuesto y no quedar excesivamente debajo (en términos de AUC) de aquellos grupos que tengan la mejor performance.

   IMPORTANTE:

   a) Se **penalizará** a aquellos grupos que hagan pocos submits.

   b) Para el **20/6** (inclusive), cada equipo debe haber realizado un **primer submit**. No importa que el mismo tenga una mala performance. El objetivo de este submit es que, para dicha fecha, ya se encuentren **familiarizados** con el funcionamiento de Kaggle. <u>Esto es condición necesaria para aprobar el TP</u>.

c) Para el **27/6** (inclusive), cada equipo debe haber realizado, al menos, un **submit** cuya performance sea un **5% mayor** al benchmark básico. Esto es condición necesaria para aprobar el TP.

2) **Código**. Deberán entregar el código que lleve adelante todo lo presentado en el informe y que genere la solución final propuesta (**25%** de la nota final). Se deberá entregar un único script o notebook de Python que lleve adelante todo lo presentado en el informe (gráficos, creación de variables, selección de modelo, etc.). El mismo debe ejecutarse de punta a punta sin errores y debe ser claro y legible para una persona ajena al grupo.

3) **Informe**. Se debe entregar un informe que presente la solución propuesta (**45%** de la nota final). El mismo debe tener como máximo 6 carillas (pudiendo ser menor, siempre y cuando cubra lo que se pide). El informe debe llevar adelante una descripción de la estrategia que utilizaron para generar sus predicciones. Idealmente el informe debe hacer referencia a secciones de dicho código que muestren cómo fue que llevaron adelante lo que reportan. A lo largo del informe pueden ir reportando los desafíos a los que se enfrentaron y cómo los superaron (por ej.: "*los datos eran muy voluminosos y no se podían cargar en su totalidad en memoria, de modo que optamos por usar una muestra elegida de … manera*", "*entrenar un modelo demora un tiempo considerable, de modo que no realizamos una búsqueda exhaustiva de hiperparámetros, entonces elegimos los hiperparámetros de acuerdo al criterio …*").

Secciones a incorporar en el informe:

A continuación se listan la estructura que **debe** seguir el informe:

1. Análisis exploratorio de datos. No debe ser hiper exhaustivo, pero se pide que sí se incorporen los siguientes puntos:
    a. Debe contener al menos dos figuras que muestren algún/algunos insights adquiridos y el informe los debe detallar.
    b. Debe mencionar cualquier característica de los datos que les haya llamado particularmente la atención (a modo de ejemplo, podría mencionar algunos de los siguientes puntos: valores missings en los predictores, predictores con poca varianza).
2. Selección de variables / Ingeniería de atributos probada. Detalles:
    a. Además de mencionar qué decisiones tomaron (por ej: no considerar la variable X), mencionen brevemente los motivos por el cual tomaron dicha decisión (por ej: "*optamos por no considerar la variable X por no poseer prácticamente variabilidad*").
    b. Esa sección puede presentar pruebas no hayan sido usadas en el modelo final, en cuyo caso se debe explicar por qué no se las usó y el motivo por el que creen que no funcionó. A modo de ejemplo, algunas opciones que podrían probar son:
        i. Crear atributos de fecha (ej.: hora, minuto, día de la semana, etc.).
        ii. Discretizar atributos continuos.
        iii. Hacer transformaciones de atributos numéricos (por ej. transformaciones logarítmicas).
        iv. Hacer *bin-counting* de variables categóricas que crean que pueden tener poder predictivo, pero no puedan manipular por la alta cantidad de valores distintos que poseen.
3. Proponer, justificar y utilizar un sistema de validación adecuado.
    a. Justifiquen brevemente la decisión tomada.
    b. Analicen si los valores de performance obtenidos en validación se condicen con los obtenidos en la plataforma de Kaggle sobre el conjunto de evaluación. Si no dieran

los mismos valores, propongan brevemente una hipótesis del motivo por el cual se da esto.

4. Detallar el/los algoritmo/s de aprendizaje usado/s y los hiperparámetros seleccionados.
    a. Mencionen los distintos algoritmos probados.
    b. Detallen brevemente la estrategia que utilizaron para encontrar buenos hiperparámetros para el algoritmo final seleccionado.
    c. En caso de ensamblar distintos modelos, mencionen cómo fue hecho esto y si mejoró o no el resultado.
5. Detallar del total de tiempo que asignaron al trabajo práctico, cuánto dedicaron a cada uno de los ítems anteriores.

Fechas y modalidad de entrega

- Se podrán realizar **submits** en Kaggle hasta el **16/7** (inclusive). Pasado ese momento, se cerrará la competencia y se harán públicos los *scores* del leaderboard privado.
- El **informe** y el **código** podrán entregarlos hasta el **23/7** (inclusive). Para ello deben enviar por mail el código e informe (en formato pdf) a datamining.mim@gmail.com. En el correo deben estar copiados todos los integrantes del grupo y se debe identificar qué nombre le pusieron a su equipo en Kaggle.
- **Sólo 1** integrante del grupo debe realizar la **entrega**.