Proyecto de Datos 1

Entrega 2: Preparación de datos

El objetivo de esta entrega es preparar los datos para la fase de modelado. Para ello seguiremos un proceso ETL (*Extract-Transform-Load*).

El trabajo requiere ya programar y gestionar un proyecto de AA colaborativamente y se usarán para ello herramientas software. En particular, debéis usar un repositorio Git en GitHub. Con respecto a los lenguajes de programación, podéis usar además cualquiera de los aprendidos en la carrera. Respecto a los entornos de desarrollo podéis usar el que os resulte más cómodo.

Además, debéis entregar una memoria donde se resuma brevemente el trabajo llevado a cabo que consistirá en cada uno de los siguientes puntos.

1. Extracción de los datos

En este punto debes documentar el proceso de captura de datos que iniciasteis de forma tentativa en la anterior entrega. Este proceso puede implicar el uso de APIs de redes sociales, webs públicas, aplicaciones diversas o la realización de *web scraping* y *web crawling* de sitios web. En concreto debéis identificar brevemente las siguientes cuestiones para cada una de las fuentes de datos utilizadas:

- Origen de Datos: Indica la URL desde la que has accedido o el *endpoint* de la API
- Herramientas o librerías utilizadas para la extracción de datos
- Aspectos relevantes de la captura: Explica brevemente la lógica que has seguido para acceder a los datos, objetivo de cada una de las operaciones de captura o consultas hechas, indicando se relacionan entre sí. Puedes usar pseudocódigo, pero no pegues el código original.
- Problemas encontrados: Documenta si habéis encontrado alguna restricción para acceder a los datos y la estrategia o estrategias seguidas para sortearla.

2. Transformación de los datos

Para cada una de las fuentes de datos que planeas utilizar en tu proyecto realiza los siguientes pasos y documentarlos en la memoria. Además, deberás desarrollar código para automatizar las tareas y poder reutilizarlas en el futuro.

1.1 Evaluación inicial

Identifica cuántas observaciones tiene tu conjunto de datos y haz un listado de todas las variables disponibles indicando para cada una de ellas:

- Fuente: Fuente donde se encuentra si hubiera más de una.
- Formato: Formato en el que se encuentra (texto, entero, decimal y precisión, etc)
- Utilidad: Si es útil o no para el objetivo a predecir. Si no lo es, no la incorpores a los siguientes pasos.
    - La utilidad puedes indicarla de forma binaria (SI/NO) o ponerla de forma más matizada (p.ej. SI/NO/QUIZA/SECUNDARIA)
- Transformaciones a realizar: Indica si vas a transformarla o si la vas a usar para construir alguna variable nueva (indicando cuál y qué medirá).
    - Algunas transformaciones se suelen diferir hasta el momento de trabajar con el modelo. Por ejemplo, *one-hot encoding* o tomar logaritmos. En esos casos, la variable se suele almacenar sin transformar.

1.2 Limpieza de los datos

Para cada variable indica si has tenido que realizar algún proceso de limpieza.

En caso de existir valores problemáticos debes indicar qué decisiones has tomado (imputar los valores, ignorarlo, eliminar la variable, eliminar la fila…).

Para llevar a cabo el proceso deberás programar una función o serie de funciones que evalúen si hay valores perdidos, comprueben el rango de valores usado, etc.

1.3 Integración de datos

Describe si has tenido que llevar a cabo un proceso de integración de datos al integrar datos de distintas fuentes (añadiendo atributos) o de distintas "tandas" (añadiendo filas de, p.ej. ejecuciones distintas de un script).

Documenta cualquier operación reseñable, como si has tenido que eliminar elementos o atributos duplicados.

Para llevar a cabo el proceso deberás programar una función o serie de funciones que faciliten estas tareas y la automatización del proceso.

1.4 Creación de nuevas variables

Indica qué variables has tenido que crear y describe de forma sucinta cómo se crean esas variables y qué miden. Si lo consideras adecuado, usa fórmulas matemáticas o pseudocódigo.

Para crear cada variable deberás programar una función o serie de funciones que faciliten estas tareas y la automatización del proceso.

3. Carga de los datos

Para el destino final de los datos usaremos ficheros Parquet, los cuales nos proporcionan muchas ventajas como que ocupan menos espacio en disco y su carga es más rápida.

El formato de archivo Parquet de manera estándar está diseñado para almacenar una única tabla bidimensional (o DataFrame) por archivo, donde cada archivo representa una matriz bidimensional de datos.

Uno de los principales puntos fuertes de los ficheros Parquet es que nos permiten hacer "consultas" a la tabla almacenada sin necesidad de cargarla previamente. De esta forma podemos seleccionar únicamente las columnas y las filas que nos interesen de manera similar a como haríamos una consulta SQL a una tabla de una base de datos.

Los ficheros Parquet en otros entornos permiten almacenar y recuperar estructuras anidadas (p.ej. una tabla de clientes donde hay un campo transacciones que almacena todas las transacciones del cliente). En nuestro caso, los usaremos de manera más sencilla y almacenar cada tabla (cada esquema) en un fichero Parquet. Si se considera adecuado, pueden crearse varios ficheros Parquet para un mismo esquema (por ejemplo, transacciones de diferentes años o productos de diferentes categorías) porque tenga más sentido para su uso.

En esta etapa tendréis que decidir razonadamente por una forma de organizar los datos en ficheros que tenga sentido para el análisis posterior que vais a realizar.

4. Exploración de los datos

En esta etapa se debe realizar un análisis descriptivo y exploratorio de las variables. Este análisis permitirá comprender mejor los datos y tomar decisiones informadas durante la fase de modelado y entrenamiento del modelo.

Debéis realizar un análisis descriptivo que resuma las características principales de las variables que entran en juego. Así como un análisis exploratorio para descubrir posibles relaciones entre las variables en juego, en especial, con la variable dependiente.

En la memoria debéis documentar únicamente los hallazgos y observaciones principales encontradas en este análisis en la memoria del proyecto con especial atención a los que os permitan comprender mejor los datos y el problema de aprendizaje automático que queréis resolver. Además, el repositorio debe incluir el código necesario para realizar estos análisis y poderlos reproducir.

Repositorio del proyecto

El código del proyecto debe ser centralizado en un repositorio Git en GitHub. Este repositorio debe estar correctamente estructurado y documentado. El profesor podrá revisar la actividad en el repositorio a lo largo del desarrollo del proyecto, valorando tanto la gestión adecuada del repositorio como la calidad de los *commits* y el trabajo de cada integrante del equipo. El repositorio debe ser privado, aunque me debéis dar acceso a él.

Cada cambio significativo en el proyecto debe ser registrado a través de un *commit* en el repositorio. En cada *commit* debe estar claramente identificado con el autor. Cada *commit* que

reflejen un único cambio o mejora significativa en el código. Esto no sólo hace que el historial de cambios sea más legible, sino que también facilita la localización y corrección de errores, y las atribuciones (quién ha realizado qué tarea).

El objetivo del repositorio es que cada uno de los pasos que se llevan a cabo en el proyecto sean reproducibles en el futuro por el equipo de desarrollo o por terceras personas. Por tanto, debéis estructurarlo, documentarlo y separar adecuadamente en ficheros de configuración cosas como claves de API, credenciales de acceso a servicios, etc en ficheros de configuración.

Una posible estructura del repositorio hasta el momento es ésta.

```
proyecto-aa/
│
├── src/             # Código fuente del proyecto.
│   ├── adquisicion/   # Módulo de adquisición de datos
│   ├── limpieza/      # Módulo de limpieza de datos
│   └── exploracion/   # Scripts o noteboks de exploración de datos
│
├── .gitignore # Archivos no rastreados
├── README.md # Descripción del proyecto y configuración
└── requirements.txt # Dependencias del proyecto
```

Para facilitar la reusabilidad del código y las buenas prácticas de programación, implementaremos un módulo para cada una de las subcarpetas del directorio `src` proyecto. En cada módulo, deberéis implementar las distintas tareas a realizar como funciones debidamente documentadas. La función `main` de cada módulo debe orquestar la llamada a otras funciones del módulo realizando la secuencia de datos esperada. También sirve para probar el módulo de forma aislada. Se recomienda usar argumentos de línea de comandos (utilizando módulos como `argparse`) para hacer el módulo más flexible y permitir que los parámetros (como URL de origen de datos, rutas de archivos, etc.) se especifiquen en tiempo de ejecución.

En el módulo de exploración, sí podéis usar scripts o notebooks de jupyter.

Entrega

Prepara un **informe de 5 páginas máximo** (fuente a tamaño 11 e interlineado sencillo) en la que respondas a lo que se indica en las secciones anteriores. El repositorio de Git será evaluado con lo que haya en la fecha de la entrega. También debes preparar unas **diapositivas** para presentar la entrega en clase (**9 minutos por grupo**). Las entregas se realizan a través del campus virtual y entregará únicamente un miembro.

Fecha límite de entrega del INFORME: 4 de marzo 23:55

Fecha límite de entrega de la presentación: 6 de marzo 13:55