Hito 2 - Implementación

- Para realizar este proyecto debes haber cursado previamente todos los módulos de la carrera Data Science.
- Luego de resolver el hito, comprime la carpeta con lo solicitado y sube el `.zip` a la plataforma.
- El proyecto debe ser desarrollado en equipo, el cual debe estar compuesto por un mínimo de 4 y un máximo de 6 integrantes. *No se podrá trabajar de forma individual.*
- En cada hito, los y las estudiantes tendrán una nota individual y una grupal.
- Puntaje Total del Hito: 10 puntos.

Objetivo

Presentar el estado de avance del proyecto y los principales hallazgos.

Descripción

En este hito los grupos deben presentar el avance de su trabajo. El análisis debe incluir:

- Análisis descriptivo y gráficos relevantes para la investigación
- Limpieza, estructuración de los datos y Feature engineering. Preparación de vector objetivo, binarización, definición de atributos y depuración de valores perdidos, outliers, tokenización de texto y calibración de clases.
- Entrenamiento de modelos candidatos. Argumentar razones para la selección.
- Según las características específicas de cada proyecto, el o la docente podrán solicitar avances particulares para este hito.

Entrega

Los puntos anteriormente descritos que componen el hito 2 serán evaluados con un máximo de 10 puntos divididos en un documento y una exposición.

Documento Técnico

Consiste en **1** archivo principal de tipo ***Jupyter notebook***, para ser ejecutado con kernel de Python 3, junto con todos los archivos necesarios para permitir su correcta ejecución desde un entorno de Anaconda con Python 3.7 o superior. Adicionalmente, en caso de que la solución lo amerite, se puede incluir 1 Jupyter notebook adicional con kernel de PySpark.

El entregable debe incluir:

- Jupyter Notebook principal con kernel Python 3
- Jupyter Notebook adicional con kernel PySpark (opcional)
- Archivo(s) de funciones auxiliares
- Imágenes (opcional, puede incluir logo, diagramas de flujo, etc)
- Set(s) de datos
- Script SQL para generar set de datos inicial (opcional)

Antes de entregar, recuerde reiniciar el kernel del Jupyter notebook, y realizar una ejecución en orden de cada celda.

Los puntajes se distribuirán de la siguiente manera:

- Preparación del ambiente de trabajo **(1 Punto)**
  - Definir brevemente el problema y la solución propuesta, indicando la naturaleza del conjunto de datos, e indicando el vector objetivo.
  - Incluir un chunk con los módulos adicionales necesarios de instalar, que no vienen incluidos dentro de un ambiente básico de Anaconda (opcional).
  - Indicar con código markdown si es necesaria la instalación de un driver / motor de base de datos (opcional).
  - Importar todos los módulos y funciones propias (en archivo(s) auxiliar(es)) necesarias para la correcta ejecución del notebook.
  - Ingesta de datos al ambiente de trabajo:
    - **Desde archivo CSV:** Importar el o los archivos necesarios para generar el o los DataFrame de trabajo iniciales.
    - **Desde SQL:** En caso de ser una base de datos alojada en algún servidor, se debe adjuntar las credenciales de acceso en un archivo .py auxiliar. En caso de que se haga la conexión a una base de datos local, incluir script .sql para generarla. En caso de que no sea posible

dar acceso a los datos (ni mediante credenciales ni mediante script), explicar brevemente el proceso realizado (puede adjuntar un diagrama de flujo), y consultas realizadas (en formato markdown-sql) para generar el set final en formato CSV, el cual se debe incorporar al ambiente de trabajo.
        - **Desde API:** Incluir solamente una muestra de cómo descargar un conjunto pequeño de datos, adjuntando explicación del proceso. La descarga total de datos la deben realizar los alumnos, los cuales deben venir incluidos en formato CSV, y luego deben agregarse desde este formato al ambiente de trabajo en Jupyter notebook.
        - En caso de contar con un volumen de datos que amerite ingesta y pre procesamiento inicial (ver siguiente punto) en un clúster EMR, esto debe ir desarrollado en un Jupyter notebook con kernel PySpark. El set de datos resultante debe llevarse a formato CSV y adjuntarse dentro de los archivos entregables, para de esta forma poder importarlo en el Jupyter notebook principal.
        - Consultar con el docente / ayudante por algún caso no cubierto en esta pauta.
    - Pre procesamiento y limpieza ***inicial***; Luego de mostrar una muestra del set de datos:
        - Remover columnas de índice como Unnamed: 0 y/o index, en caso de que se presenten.
        - Asignar nombres de columnas, en caso de que no vengan definidas.
        - Generación de nulos, en caso de que los considerados "datos perdidos" vengan en otra representación (Ej: string).
        - Transformación de tipo de dato, en caso de que corresponda (privilegiar operaciones vectorizadas).
        - Cualquier otra operación que se considere necesaria para el caso específico en esta etapa.
    - Mostrar columnas originales de los datos, cantidad de datos de cada una, y su tipo de dato correspondiente.
    - Identificar la porción de datos correspondiente a validación. Se recomienda (siempre que el caso específico lo permita) utilizar 1 DataFrame para modelamiento y validación, identificando mediante una columna adicional si el registro corresponde a modelamiento o validación. De esta forma, se asegura que la selección de atributos para modelamiento es la misma para ambas muestras.

- Análisis descriptivo y pre selección de atributos **(1 Punto)**
    - Análisis univariado de atributos y vector(es) objetivo(s). Usar métricas y/o gráficos adecuados según se trate de datos continuos o categóricos.
    - Análisis bivariado entre atributos y con respecto a vector(es) objetivo(s) (cuando corresponda).
    - Correlaciones.

  - Exploración de outliers.
  - Exploración de datos nulos.
  - Frecuencia de palabras (en caso de trabajar con textos).
  - Cualquier otro análisis que considere necesario para su caso específico en esta etapa.
  - Pre selección de atributos: Si mediante los resultados obtenidos en los análisis anteriores determina que en esta etapa se cuenta con columnas innecesarias, puede eliminarlas dando la correspondiente justificación.

- Preprocesamiento para modelación y feature engineering. De los siguientes puntos, aplicar los que correspondan a su caso de trabajo específico **(1 Punto)**
  - Imputación de nulos (mostrar distribuciones / frecuencias posteriores).
  - Tratamiento de outliers (mostrar distribuciones / frecuencias posteriores).
  - Recodificación y/o binarización de atributos y/o vector objetivo.
  - Estandarización de datos.
  - Reducción de dimensiones (puede mostrar correlaciones posteriores).
  - Calibración de clases (mostrar frecuencias posteriores).
  - Selección de atributos para modelamiento. *Debe incluir justificación de la selección hecha*.
  - Generación de subsets para distintos modelos según clases específicas.
  - Selección de método de vectorización y parámetros correspondientes, incluyendo función de pre procesamiento (textos).
  - Cualquier otro procesamiento que considere necesario para su caso específico en esta etapa.

- Modelos candidatos **(1 Punto)**
  - Generar muestra(s) de validación (en caso de estar dentro del DataFrame original).
  - Generación de subsets de entrenamiento y prueba a partir del/(de los) set(s) destinado(s) a modelamiento.
  - Entrenamiento de al menos 2 modelos (para cada vector objetivo y/o set de modelamiento), con o sin modificación de hiperparámetros. En caso de modificar hiperparámetros, el uso de una grilla de búsqueda no es necesaria en esta etapa.
  - Realizar predicciones en set de prueba y de validación, y reportar métricas de desempeño en cada caso (puede también mostrar las del set de entrenamiento).
  - Según resultados obtenidos, *indicar* cuál corresponde al mejor modelo parcial, y/o posible(s) candidatos al modelo final.
  - Serializar *todos* los modelos entrenados. En caso de haber utilizado búsquedas de grilla, serializar solo el mejor modelo de la grilla.
  - En caso de usar MLlib, realizar la ingesta de los set de datos listos para modelamiento y validación (desde un bucket de s3) en un Jupyter notebook con kernel de PySpark, junto con el resto de pasos de este punto.

- Calidad y claridad del código y documentación. **(1 Punto)**
    - Respetar los estándares de buenas prácticas de trabajo para Python para nombrar variables, funciones, alias de módulos, etc.
    - Favorecer al máximo el uso de funciones, las cuales deben ir en un archivo auxiliar que se importe al ambiente de trabajo (no definir funciones del Jupyter notebook, a excepción de la función de pre procesamiento de texto). Las funciones propias deben incluir su correspondiente docstring.
    - Todos los import de módulos, librerías y archivos deben estar en 1 chunk al comienzo del Jupyter notebook.
    - Hacer uso de Markdown para separar secciones de trabajo mediante títulos y subtítulos, así como para los análisis y comentarios de los resultados obtenidos.
    - Hacer uso de comentarios de Python dentro de los chunks de código cada vez que lo considere necesario, solo para dar explicaciones. No entregar código ejecutable comentado, excepto en descargas de módulos.
    - Todo "output" (print, dataframes, gráficos, salidas de funciones, etc) *debe* ir acompañado de un comentario y/o análisis; No es necesario comentar cada gráfico de una grilla, o cada columna de un set de datos, se debe comentar principales hallazgos y/o conclusiones.
    - Evitar redundancia de información; Escoger, entre prints, gráficos o tablas, la mejor forma que consideren para el caso específico que se desea informar.
    - Los gráficos deben incluir título, etiquetas en los ejes, y leyenda de colores en caso que lo requiera. Los textos deben ser de un tamaño de fuente adecuado. Favorecer mostrar gráficos en grillas siempre que sea posible, y que se cumplan los alcances anteriores.
    - Favorecer el uso de operaciones vectorizadas y evitar el uso de bloques de ciclos.

Exposición

Presentar el avance del proyecto con al menos la participación del 75% de los integrantes del grupo de trabajo presentes en la clase.

La presentación será evaluada según los siguiente criterios:

- Evaluación grupal
  - Exposición del avance y justificación de las decisiones tomadas en el análisis, manipulación de los datos y modelos candidatos. **(2 Puntos)**
  - Calidad de la presentación e información expuesta. **(1 Punto)**
- Evaluación Individual
  - Dominio del tema y capacidad de exponer claramente los conceptos. **(2 Puntos)**

Consideraciones

- Si el o la estudiante no participa de la presentación su puntaje será de **0 Puntos**.
- Se dispondrá de 15 minutos para exponer el estado de avance del proyecto

Importante: Cada estudiante deberá subir a la plataforma el documento, los archivos auxiliares y la presentación en un archivo comprimido antes que comience la clase donde será presentado este hito.