# Capstone project progress

## Análisis y comprensión (EDA)

### Comprensión:

El dataset a analizar está compuesto por información acerca de los editores de Wikipedia en español. Las columnas comprenden datos acerca de la antiguedad del editor, el tipo de material que ha editado, la cantidad de ediciones, páginas creadas, interacción de varios tipos en wikipedia y el género que se ha asignado mediante distintas investigaciones.

Hablando de estas últimas columnas que consideran el género del editor, la columna C\_man representa el género deducido por alguna persona al estudiar el pérfil del editor fijándose en el artículo con el cual el editor se refiere a sí mismo. Por otra parte, la columna C\_api obtiene el género que la API de wikipedia entrega sobre el editor.

Como se puede apreciar, el género de los editores es el tema principal de este dataset. En todas las columnas relacionadas al género tenemos una gran cantidad de registros con el género indeterminado, lo cual hace difícil obtener conclusiones acerca de qué puede hacer Wikipedia para aumentar el número de mujeres editoras. La importancia de esta pregunta puede deberse a que Wikipedia se sostiene principalmente de donaciones ya que no cobra por la lectura de ningún artículo publicado. Puede ser que el al aumentar el número de mujeres editoras seas acreedores a algunos apoyos gubernamentales entre otros. De cualquier modo, tener certeza de la población femenina de editores y sus hábitos es de vital importancia para Wikipedia.

### EDA

Con el análisis exploratorio podemos observar que:

* No existen registros nulos en el dataset
* La columna ‘gender’ resume los resultados obtenidos acerca del género sin discrepancias (no hay algún registro para el cual las columnas indiquen generos definidos distintos)
* Al agrupar columnas según el género, se tienen diferencias significativas en géneros (en especial las correspondientes a ediciones en páginas relacionadas con mujeres). Lo cual nos indica una alta probabilidad de poder predecir el género con certeza.
* Algunas columnas tienen una correlación alta, por lo que podríamos prescindir de alguna.

El dataset está bastante limpio debido al esfuerzo tomado por Wikipedia para descifrar el género como se puede leer en el artículo correspondiente.

## Pregunta a abordar

La importancia que tiene saber con certeza el género de los editores es bastante alta para Wikipedia, como se ha comentado antes. incluso en el artículo correspondiente se menciona la posibilidad de obtener una clasificación certera mediante un modelo de aprendizaje automático.

Así que la pregunta que intentaremos resolver mediante ML será: ¿Cuál es la porción del total de editores de Wikipedia correspondiente a mujeres?

Una pregunta relacionada y que debería ser fácil responder con el proceso que se llevará a cabo es: ¿Qué tan certera es la respuesta dada por el modelo? o ¿Cuánta confianza podemos tener en la respuesta proporcionada?

## ¿Por qué ML?

Como ya se ha mencionado, el esfuerzo por determinar el género de los editores ha sido basto y ha llegado a una fase de estancamiento. En estos esfuerzos se ha recolectado suficiente información para entrenar un modelo y para tener la certeza de que la información ingestada al modelo es confiable (se ha eliminado todo registro donde hubiera confusión acerca del género).

Un modelo de ML podría identificar relaciones ocultas entre las columnas para clasificar los registros de una manera mucho más confiable e informada que cualquier esfuerzo humano.

## Pipeline

Se separan los datos con género indeterminado (no se usarán en ninguna etapa de la operacionalización del modelo). Los pasos en el pipeline de datos son:

* Selección de features
* Limpieza de outliers
* Normalización de features (transformación Boxcox)
* Bootstrap para balancear ‘males’ y females’

## Creación del modelo

Debido a que la finalidad del modelo es estimar de una manera certera la población total de mujeres, la métrica que pienso optimizar es el recall, ya que compara la cantidad de TP con la cantidad entera de positivos en el dataset.

Los modelos probados fueron:

* Gradient Boosting
* Random Forest
* Logistic Regression

Al principio tuve problemas de overfitting con todos los modelos. Estos se resolvieron al hacer una limpieza de las features usadas.

El modelo seleccionado fué la regresión logística debido a que su tiempo de ejecución es óptimo y a que tuvo el mejor recall (0.91)

## Configuración/Refactorización

Se refactorizó el código segregando funciones y modularizando según su objetivo. Se eliminaron las partes “hard coded” obteniendo un archivo general de configuración llamado settings.

En esta estructura se cuenta con una clase llamada Kernel por la cual se accede a las configuraciones.

La estructura no está completa. Hace falta el archivo setup para configurar el ambiente virtual (junto con el archivo de requerimientos), una carpeta de logs y en general hace falta implementar el logging.

El Logging pienso llevarlo mediante el Kernel, es por eso que este componente está por todas partes.

## Versionamiento del modelo

Se ha integrado DVC al proyecto para llevar el seguimiento de experimentos contando con reproducibilidad. El seguimiento es simple ya que sólo considera el pipeline entero de entrenamiento como dependencia y el score obtenido como output.

## Contenerizacón y despliegue

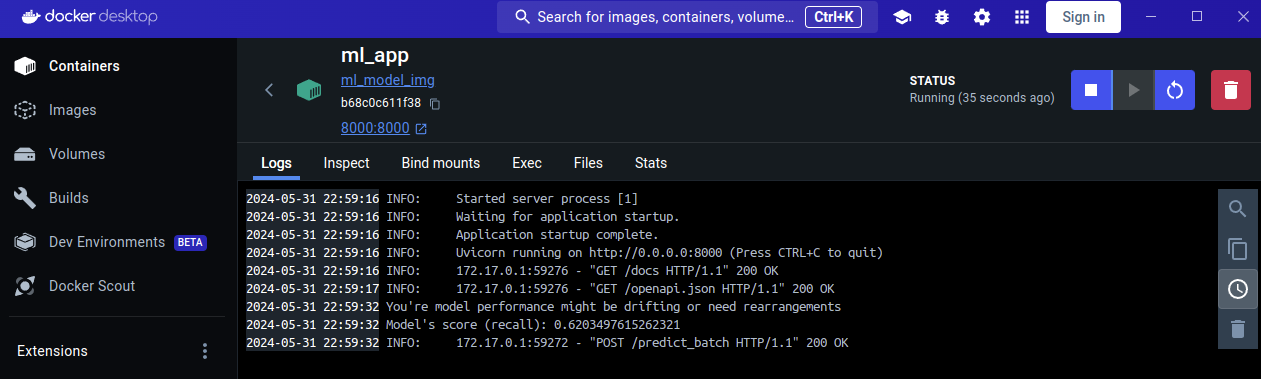
Se añadió un directorio dedicado a la creación de la imágen de docker para el despliegue. En éste mismo directorio se encuentra la implementación de la API que da el servicio de batch prediction.

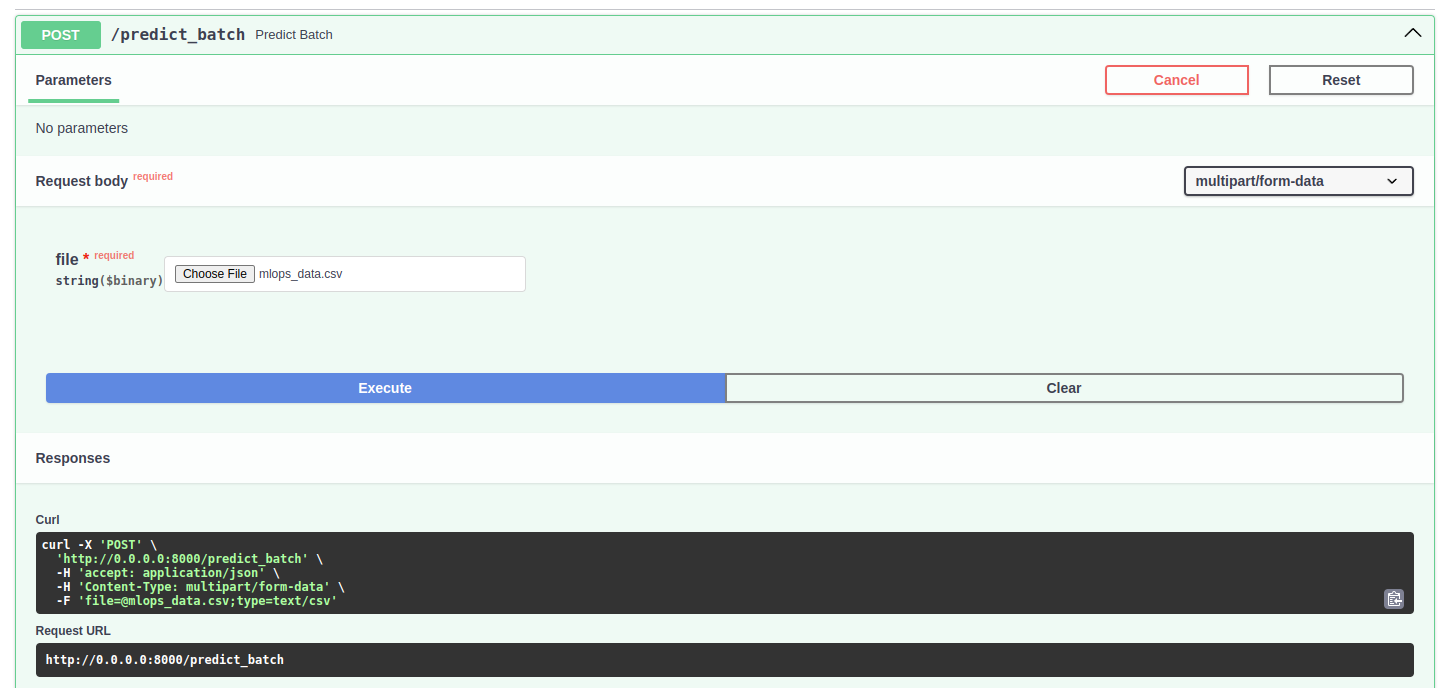
## Monitoreo

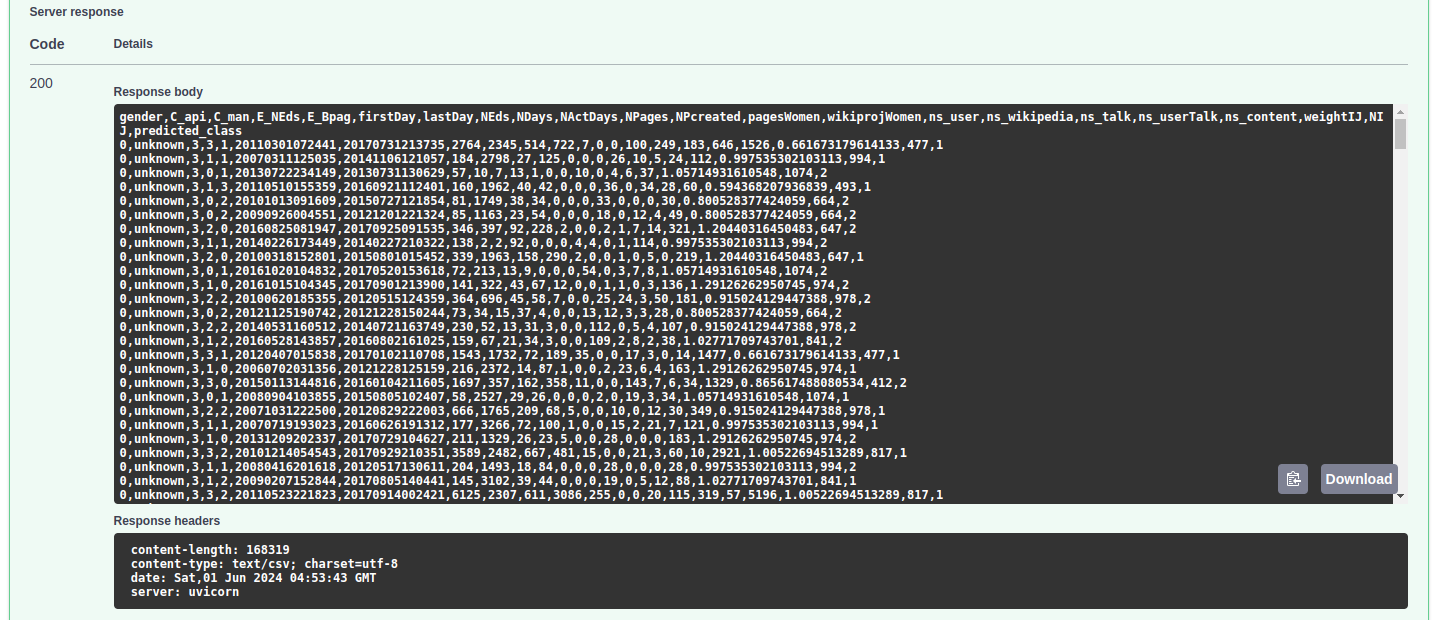
Se modificó el endpoint de la API para agregar el scoring de la predicción con el fin de monitorear el desempeño del modelo

## Evidencias

### 







## Documentación

Los diagramas y demás documentación están en el README.md del repositorio: