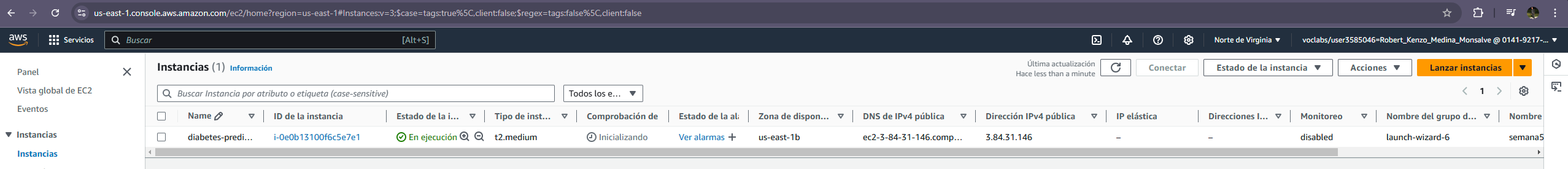
# Informe

Creación de la instancia:



Instalación de paquetes

Texto

Descripción generada automáticamente

Se instala mlflow, tox, Python, git, se crea un ambiente virtual (venv).

Texto

Descripción generada automáticamente

Ejecución servidor mlflow:

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Experimentos con los diferentes modelos

Se decidió implementar un modelo de regresión logística, random forest y el de XGboost de los tantos que se intentaron.

Captura de pantalla de computadora

Descripción generada automáticamente

A continuación, la comparación de los 3 modelos:   
Captura de pantalla de computadora

Descripción generada automáticamente

Como se observa en la comparación, el XGboost es el indicado por ser el mejor en las 3 métricas.

Entrenando el modelo

Se usa tox.ini con el modo de train para usar el archivo model/train\_pipeline.py

Texto

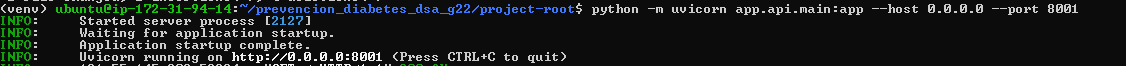
Descripción generada automáticamente

Primero, se modifica la aplicación para que tenga en cuenta la ip pública del servidor:

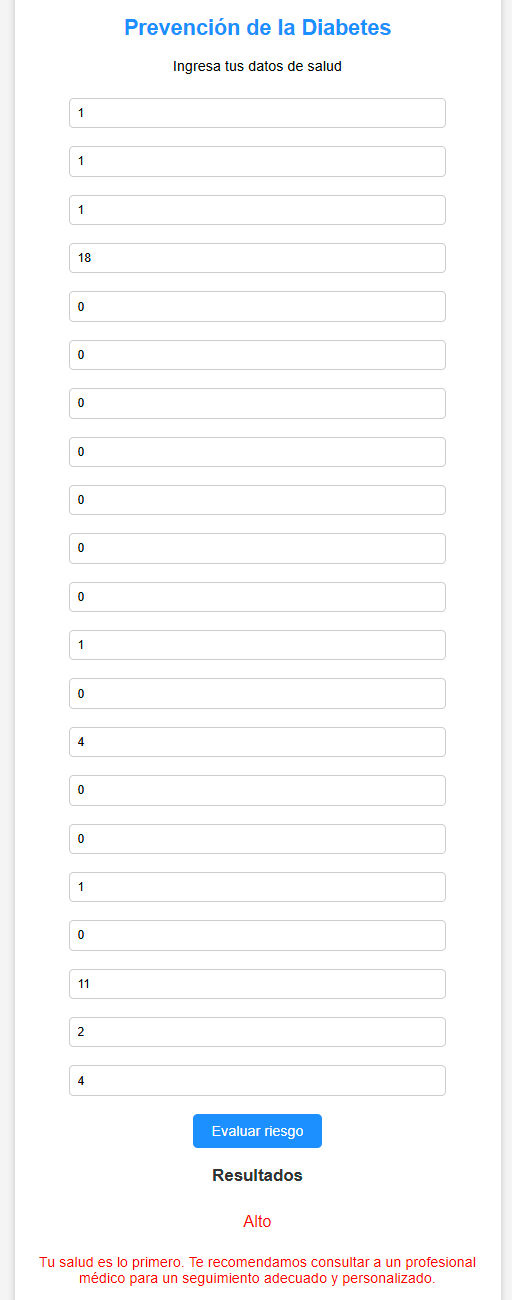
Texto

Descripción generada automáticamente

Una vez el modelo se entrenó y la modificación anterior se realizó, ponemos en marcha nuestro servidor:



Llenamos datos al azar y evaluamos:



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Aquí podemos presenciar el log del servicio:

Texto

Descripción generada automáticamente