**Parte 1: desarrollo de modelos**

1.1 Abra el archivo tox.ini y familiarícese con su estructura y contenido. En este archivo se definen dos ambientes clave que usaremos hacia adelante: testapp y run. En cada uno de estos ambientes se define un directorio para el ambiente, un conjunto de dependencias y los comandos a ejecutar. Incluya en su reporte una descripción de qué dependencias requiere cada uno de estos ambientes y qué comandos ejecuta.

En el ambiente [testenv:test\_app] se instalan las dependencias definidas en el archivo test\_requirements.txt y se ejecutan los comandos pytest \ -vv \ {posargs:app/tests/}. En este ambiente de pruebas que asegura que el código funcione como se espera mediante pruebas automatizadas.

En el ambiente [testenv:run] se instalan las dependencias definidas en el archivo test\_requirements.txt (al igual que en el ambiente test\_app) y se el comando python app/main.py.

1.2 En la carpeta app/schemas encontrará dos esquemas para la API: health y predict. Incluya en su reporte una descripción de los dos esquemas definidos allí. Explique qué contenidos incluyen estos esquemas.

El esquema Health está basado en el modelo BaseModel de Pydantic y sirve para definir la estructura de la respuesta que proporciona información de estado o salud de la API.

En el archivo predict.py tenemos dos clases Esquemas PredictionResults y MultipleDataInputs. El esquemas PredictionResults define la estructura de la respuesta de predicción, indicando los posibles resultados y errores que podrían ocurrir al realizar predicciones con el modelo. El esquema MultipleDataInputs define la estructura de los datos de entrada para realizar predicciones, permitiendo el envío de múltiples datos a la vez. Está pensado para validar las entradas en un endpoint que reciba múltiples datos de clientes

1.3 En la carpeta app encontrará el archivo principal de la api, main.py, donde se ejecuta el servidor uvicorn. Incluya en su reporte una descripción de los 4 argumentos de la función run empleados allí.

app: es la instancia de la aplicación FastAPI que define los endpoints, rutas, y toda la lógica de la API. Este argumento le indica a Uvicorn qué aplicación ejecutar.

host: Establece la dirección IP en la que el servidor Uvicorn escuchará las solicitudes entrantes.

port:Define el puerto en el que Uvicorn escucha. Aquí está configurado en 8001.

log\_level:Establece el nivel de detalle de los mensajes de registro que se generarán en la consola mientras el servidor está activo.

1.4 Finalmente, el corazón de la ejecución de las rutas de la API se encuentra en api.py. Allí encontrará las dos rutas definidas: health y predict. Incluya en su reporte una descripción de qué se ejecuta cuando se hace un llamado a cada una de estas rutas.

Para la ruta health se usa el método get. Es una ruta de diagnóstico o "ruta de salud" que confirma si la API está activa y funcional.

En la ruta predictions se usa el motodo post. Ejecuta una predicción en base a datos enviados por el cliente. Como datos de entrada recibe un json. Usa la función make\_prediction para realizar la predicción, y reemplaza valores NaN en los datos para asegurar que no haya problemas con datos faltantes.

**Parte 2: probando y ejecutando la API**

Lance una instancia en AWS EC2. Se recomienda una máquina t2.small, con sistema operativo Ubuntu y 20GB de disco. Incluya un pantallazo de la consola de AWS EC2 con la máquina en ejecución en su reporte. Su usuario de AWS debe estar visible en el pantallazo.

**A screenshot of a computer

Description automatically generated**

Note que usamos en este caso ubuntu, pues éste es el usuario creado por defecto como administrador con sistema operativo Ubuntu server. Incluya en su reporte un screenshot de la conexión a la máquina virtual

**A computer screen with white text

Description automatically generated**

**A black screen with white text

Description automatically generated**

Ingrese a la carpeta bankchurn-api cd bankchurn -api y ejecute el ambiente de prueba de la api de tox tox run -e test\_app. Incluya en su reporte un screenshot de la salida de este comando y una breve explicación.

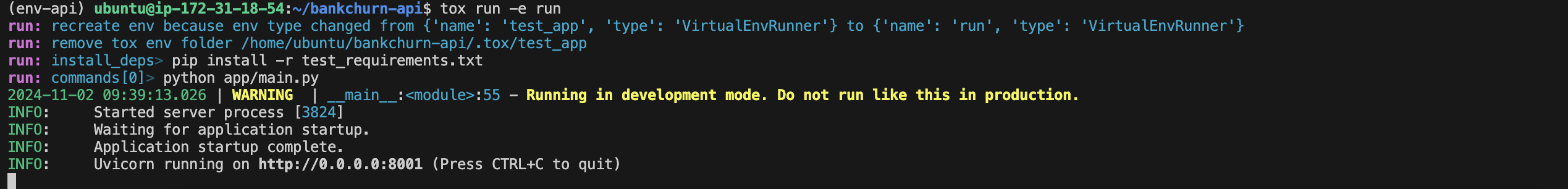
A black screen with white text

Description automatically generated

Se realiza la instalación de las dependencias requeridas para la ejecución de los scripts. Adicionalmente se ejecutan las pruebas definidas en app/tests/test\_api.py, específicamente test\_make\_prediction. Como se observa en la imagen las pruebas fueron exitosas.

Ejecute ahora el ambiente de ejecución de tox tox run -e run Incluya en su reporte un screenshot de la salida de este comando y una breve explicación.

Se ejecuta el ambiente “run”. Se realiza la instalación de dependencias definidas en el archivo test\_requirements.txt y se ejecuta el script main.py el cual incializa nuestra API la cual queda habilitada en la dirección http://0.0.0.0:8001



Interactúe con la API y sus dos rutas, la de health y la de predicción. Incluya en su reporte un screenshot donde se evidencie la ejecución de la API en el navegador y la respuesta satisfactoria de una solicitud a la ruta de predicción.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Prueba health

A black and white stripes

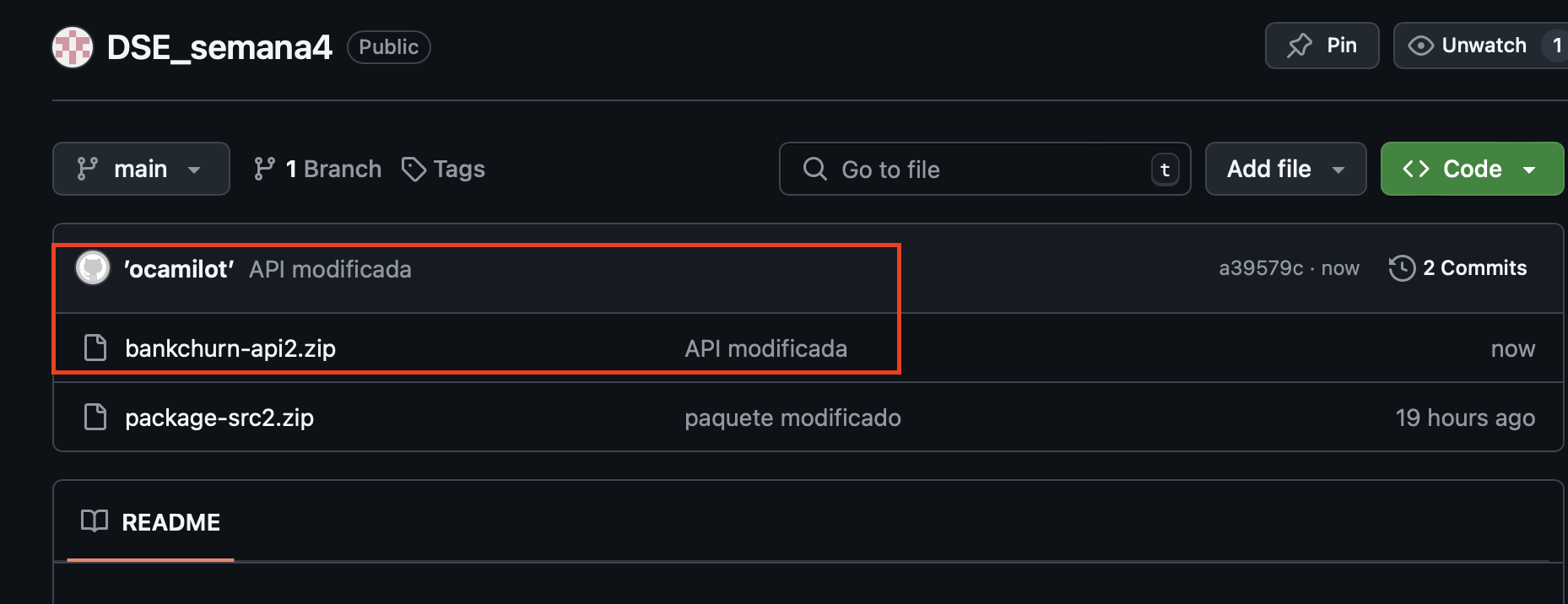
Description automatically generated

Prueba predictions

A screenshot of a computer

Description automatically generated

Incluya en su reporte un pantallazo del archivo bankchurn-api2.zip en GitHub.



Con la API en ejecución, edite el archivo de inputs para la predicción eliminando una de las variables que SÍ eliminó en la versión 0.0.2 del paquete. Incluya en su reporte un pantallazo del resultado de ejecutar la predicción una vez se hace este cambio.

La variable que elimine el archivo de inputs para la predicción es Customer\_Age. Esta variable también fue la que se había eliminado en la versión 0.0.2.

A screenshot of a computer program

Description automatically generated

Con la API en ejecución, edite el archivo de inputs para la predicción eliminando una de las variables que NO eliminó en la versión 0.0.2 del paquete. Incluya en su reporte un pantallazo del resultado de ejecutar la predicción una vez se hace este cambio.

La variable que elimine el archivo de inputs para la predicción es Total\_Amt\_Chng\_Q4\_Q1. Esta variable no fue eliminada en mi versión 0.0.2.

A screenshot of a computer

Description automatically generated