Tópicos de Ingeniería de Software 2

2024

Enunciado

Este trabajo integrador consiste en el desarrollo de un servicio Web (API) que expone un modelo de red neuronal para detectar similaridades en grafos de conocimiento.

El modelo se especializa en identificar si subgrafos hablan de la misma entidad. Esto resulta sumamente útil para eliminar duplicados en grafos de conocimiento.

El servicio solo deberá poder ser invocado por clientes de nuestra plataforma y por lo tanto deberán pasar una API Key. Las invocaciones (request) HTTP deberán contar con el header HTTP 'Authorization' indicando la API key. Si la API key no se encuentra en la invocación, ésta deberá ser rechazada.

GET /service
Authorization: <API Key generada>

{
inputs:[“propE1A,propE1B,..”,“propE2A,propE2B,..”]
}

Resultado:
{
“probabilidad”:0.7
}

Existen dos tipos de cuentas que restringen la cantidad de solicitudes HTTP por minuto que el sistema está autorizado a resolver por minuto:

- FREEMIUM; 5 solicitudes por minuto (RPM).
- PREMIUM: 50 solicitudes por minuto (RPM).

El servicio deberá satisfacer los siguientes requerimientos:

- Deberá correr el modelo entrenado previamente.
- Todos las invocaciones que reciba el servicio deberán ser controladas verificando dos aspectos:
    - Autorización. A partir de la API key, se verifica si existe registrada la API key en la base de datos del sistema.
    - Limitación. De acuerdo a la suscripción del cliente tiene una limitación de invocaciones por segundo: FREEMIUM y PREMIUM.

- Cada solicitud recibida deberá ser registrada en la bitácora (log). Capturando el tiempo que tomo para procesar el requerimiento HTTP de diagnóstico: iniciar el timer cuando se recibe la solicitud HTTP, procesar la autenticación de la key, correr la red neuronal, registrar el resultado en la bitácora, y retornar la respuesta.
- Para datos de solo lectura y de poca volatilidad, se espera que se implemente cache.
- La solución deberá ser implementada en base a microservicios.

Entregables

La solución deberá contar con los siguientes entregables:

- Instructivo para correr el proceso de entrenamiento del modelo.
- Informe de diseño donde se presentan diagramas, aclaraciones sobre los requerimientos y toma de decisiones.
- Test HTTP para probar el funcionamiento de los servicios. Pueden utilizar .HTTP , Postman, JMeter, cURL.

Fechas de entrega

15 de Enero del 2025

Preguntas

- ¿ Si se pueden utilizar librerías para solucionar la capturas de tiempo de procesamiento de los requerimientos HTTP?
  Si, se pueden utilizar.
- ¿Se tiene utilizar cualquier tecnología para resolver el trabajo?
  No, el trabajo esta restricto a Python.
- ¿Se puede realizar el trabajo en grupo?
  Si.
-