

Clasificador de LEDs

Parte 1: Clasificador

1. Entrene un modelo para clasificar combinaciones binarias de LEDs. Cada entrada del dataset consta de 8 columnas. Las primeras 7 columnas describen un valor booleano para cada LED. La octava columna describe el valor entero para la combinación de las 7 columnas anteriores. El dataset está disponible en el siguiente [enlace](#).
2. Escriba el modelo en disco de manera que pueda cargarse a memoria para consultas posteriores.

Parte 2: Servicio gRPC

1. Especifique un servicio gRPC para el clasificador. El servicio debe constar de dos llamadas:
 - a. **Llamada 1:** crea un modelo y le asocia un nombre.
 - Entradas:
 - El nombre del modelo
 - Un valor entre 0 y 1 que indica la proporción del dataset utilizada para entrenamiento
 - Un URL con la ubicación del dataset de entrenamiento.
 - Salidas:
 - Una representación básica de la entidad asociada al modelo.
 - Error en caso de que la entrada sea inválida.
 - b. **Llamada 2:** consulta un modelo.
 - Entradas:
 - El nombre del modelo.
 - Los valores booleanos para cada uno de los 7 LEDs.
 - Salidas:
 - El entero producido por el clasificador.
 - Error en caso de que la entrada sea inválida.

Parte 3: Opcionales

- Genere la especificación para lanzar el servicio gRPC a un cluster de Kubernetes.
- Decore cada llamada del servicio gRPC para que permita llamadas HTTP/JSON.
- Entrene el modelo de clasificación de manera asíncrona utilizando una cola de trabajo.
- Desarrolle utilizando Git como control de versiones (en caso de hacerlo, incluir el directorio **.git** en el archivo comprimido).
- Escriba un cliente para el servicio gRPC.

Entregable

- Debe enviar un correo electrónico a **careers@singularities.com** con su nombre en el cuerpo y un archivo ZIP adjunto que contenga al menos todo el código fuente y un **README.md** que describa su implementación.

Notas

- Se espera que exista una demostración que justifique la correctitud de su implementación.
- Puede utilizar cualquier framework para crear, entrenar y consultar el clasificador.
- La implementación debe estar escrita en C/C++, Go, Java, Node, Python o Scala.
- Se recomienda seguir [API Design Guide](#).