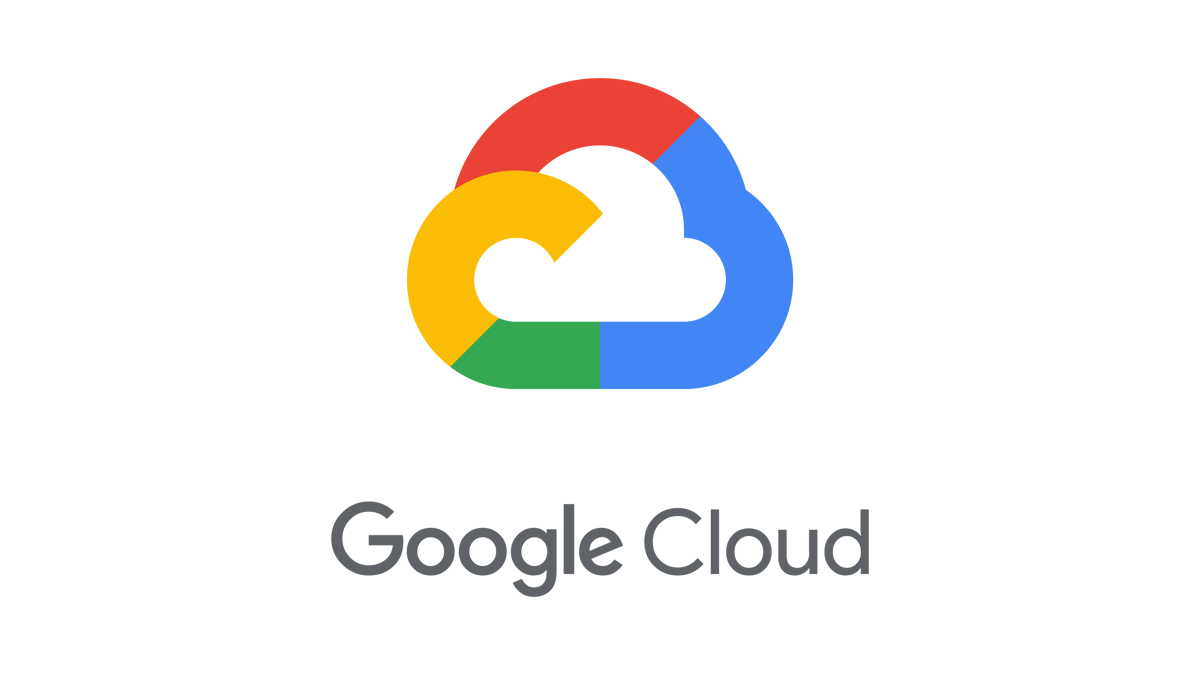
|  |
| --- |
| Lenguajes y desarrollo de soluciones IA |
| Despliegue de modelos en GCP |
|  |

|  |
| --- |
| Martín Manso de Andrés  24-7-2023 |

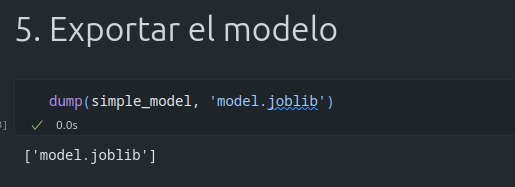


# **Exportación del Modelo**

***Nota****: En este caso y como reto personal, he decidido utilizar el modelo generado en el trabajo sobre el CSV de tips, por lo que he tenido que especificar la versión 1.2.0 de Scikit Learn para poder utilizarlo en GCP.*

Para la exportación del modelo he utilizado la librería [*joblib*](https://scikit-learn.org/stable/model_persistence.html)con la que he podido guardar el fichero en formato ***.joblib***.

Para esta exportación he reutilizado el notebook de la anterior tarea y he añadido un nuevo apartado para que una vez haya realizado la predicción, guarde el modelo.



# **Subir el modelo a Google Cloud Storage**

## 2.1 Crear el proyecto

Para ello iré a la interfaz de la consola de Google Cloud, el proyecto se llamará **IA Edix Tips (ia-edix-tips)**

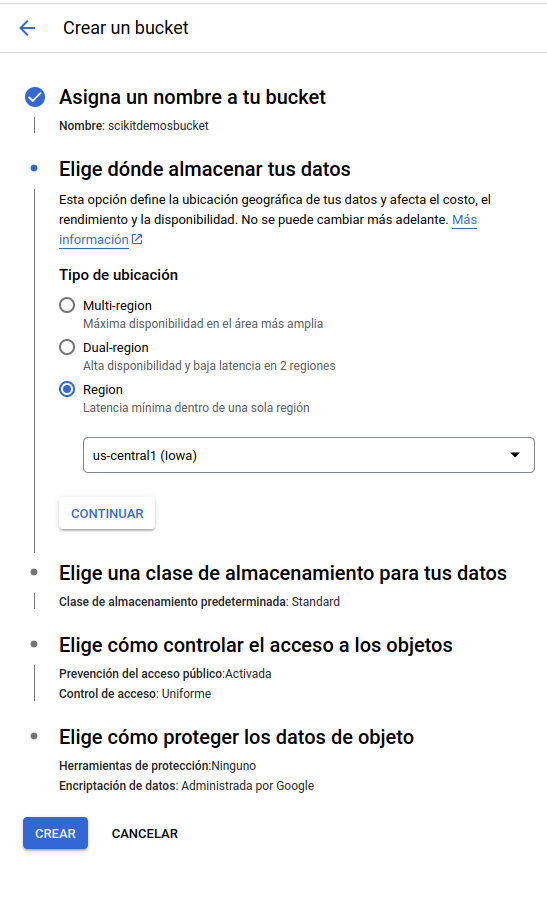


Aquí se puede ver la creación:

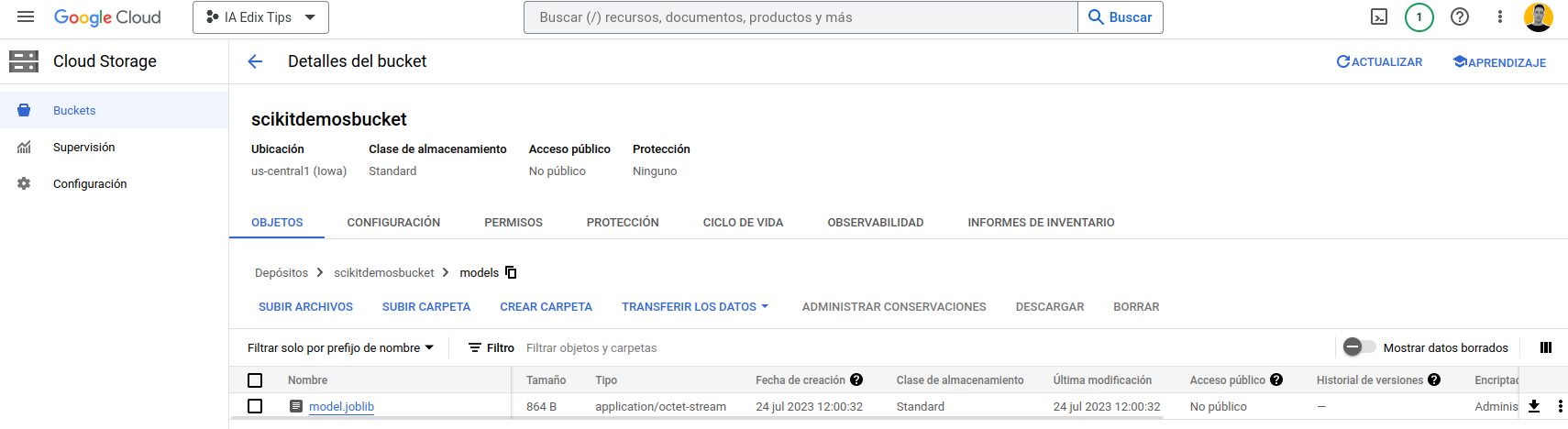
## 

## 2.2 Crear el bucket

Primero iniciamos la configuración del bucket



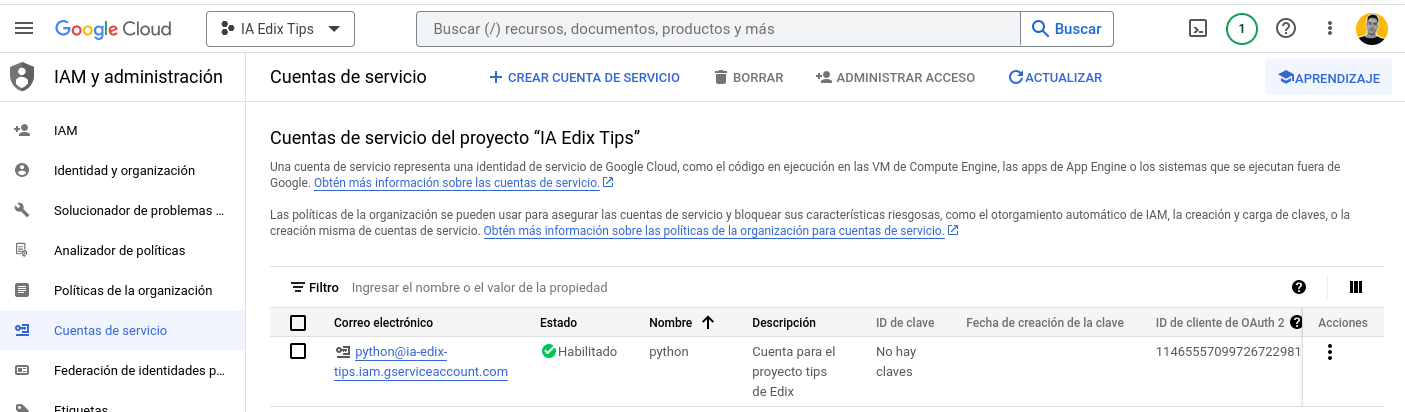
Una vez tenemos el bucket creado, procedemos a generar la carpeta models y a subir el modelo model.joblib



# **Crear cuenta Service Account**

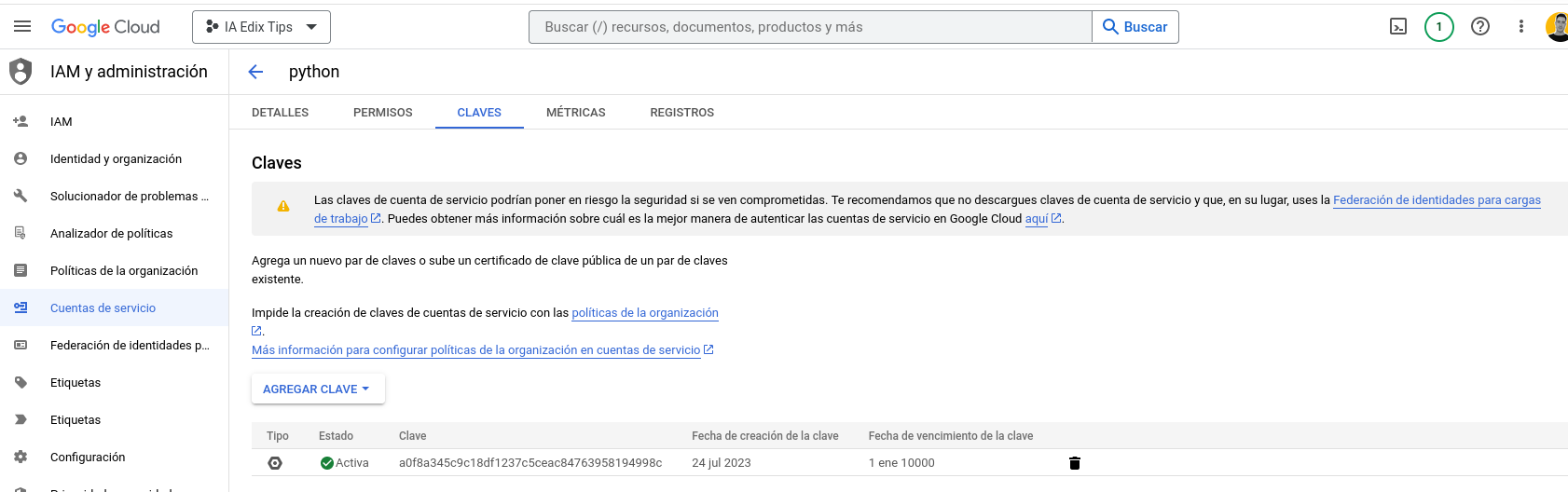
Para poder acceder a nuestro modelo en GCP tenemos que crear una cuenta de servicio, esta cuenta de servicio controlará los permisos con los que accederemos a nuestros recursos en la nube.

En este caso crearemos una cuanta con nombre **python** y rol **Vertex AI Administrator** y una vez terminado el formulario, nos aparecerá en el listado de cuentas de nuestro proyecto.

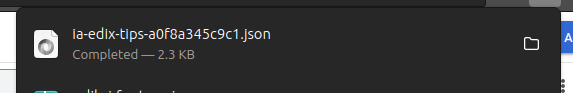


## 3.1 Configurar credenciales

Para poder utilizar la cuenta, tenemos que ir al panel de claves y una vez dentro agregaremos una nueva clave.

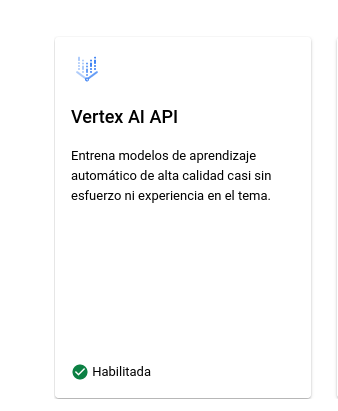


Y automáticamente se nos descargará el archivo JSON

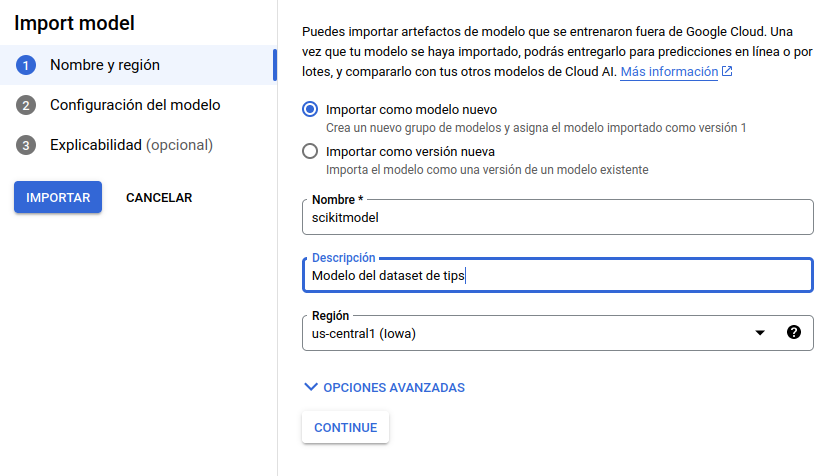


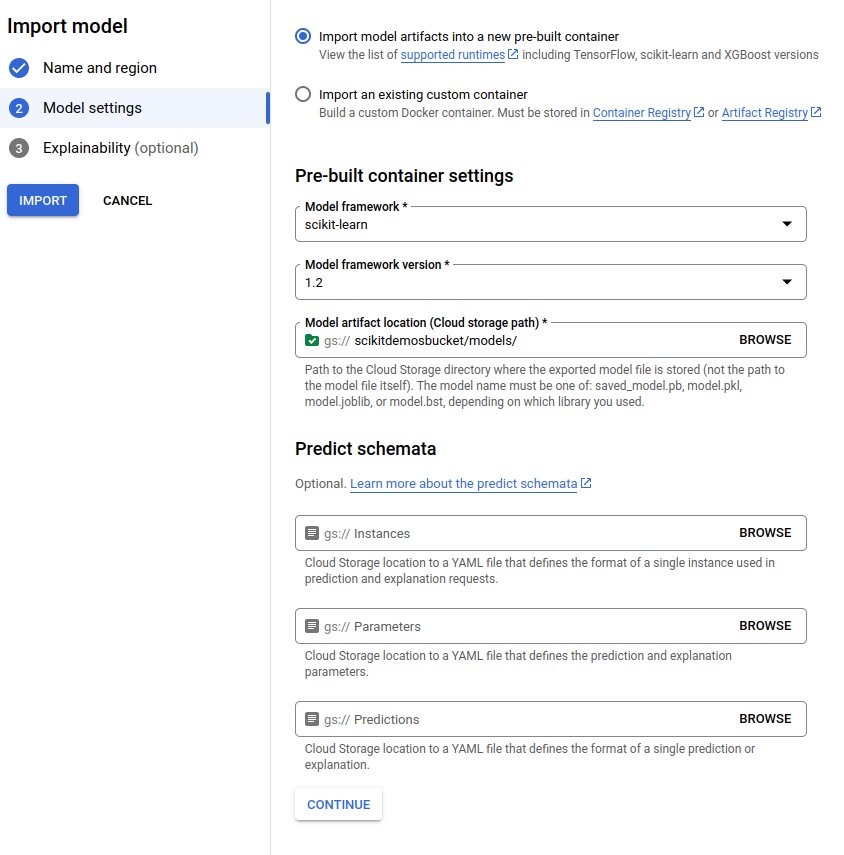
# **Crear el modelo en Vertex AI**

Primero tenemos que habilitar Vertex AI API

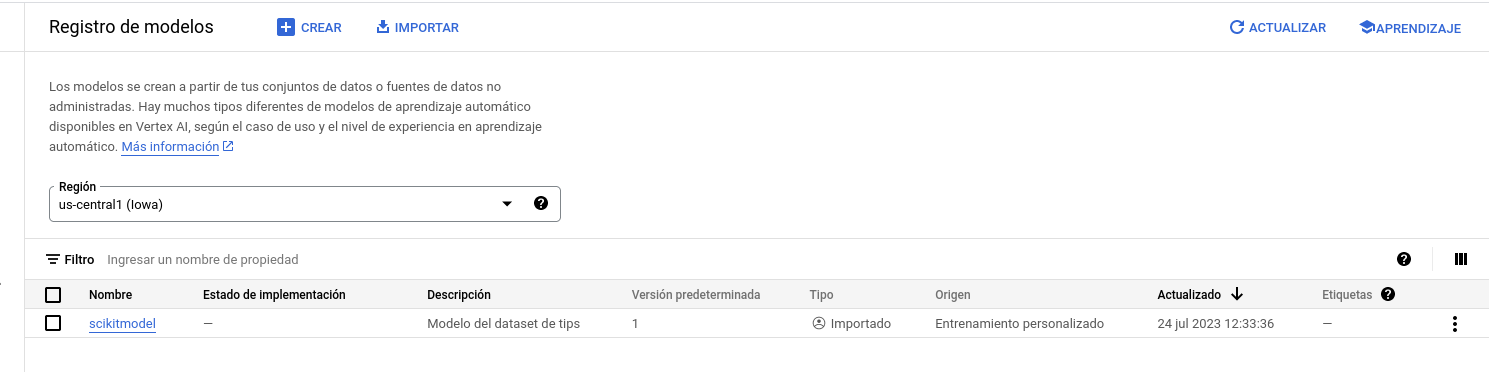


Y una vez tengamos habilitada la API vamos al Registro de modelos, configuramos la región a **us-central1 (lowa**) e importamos el modelo.



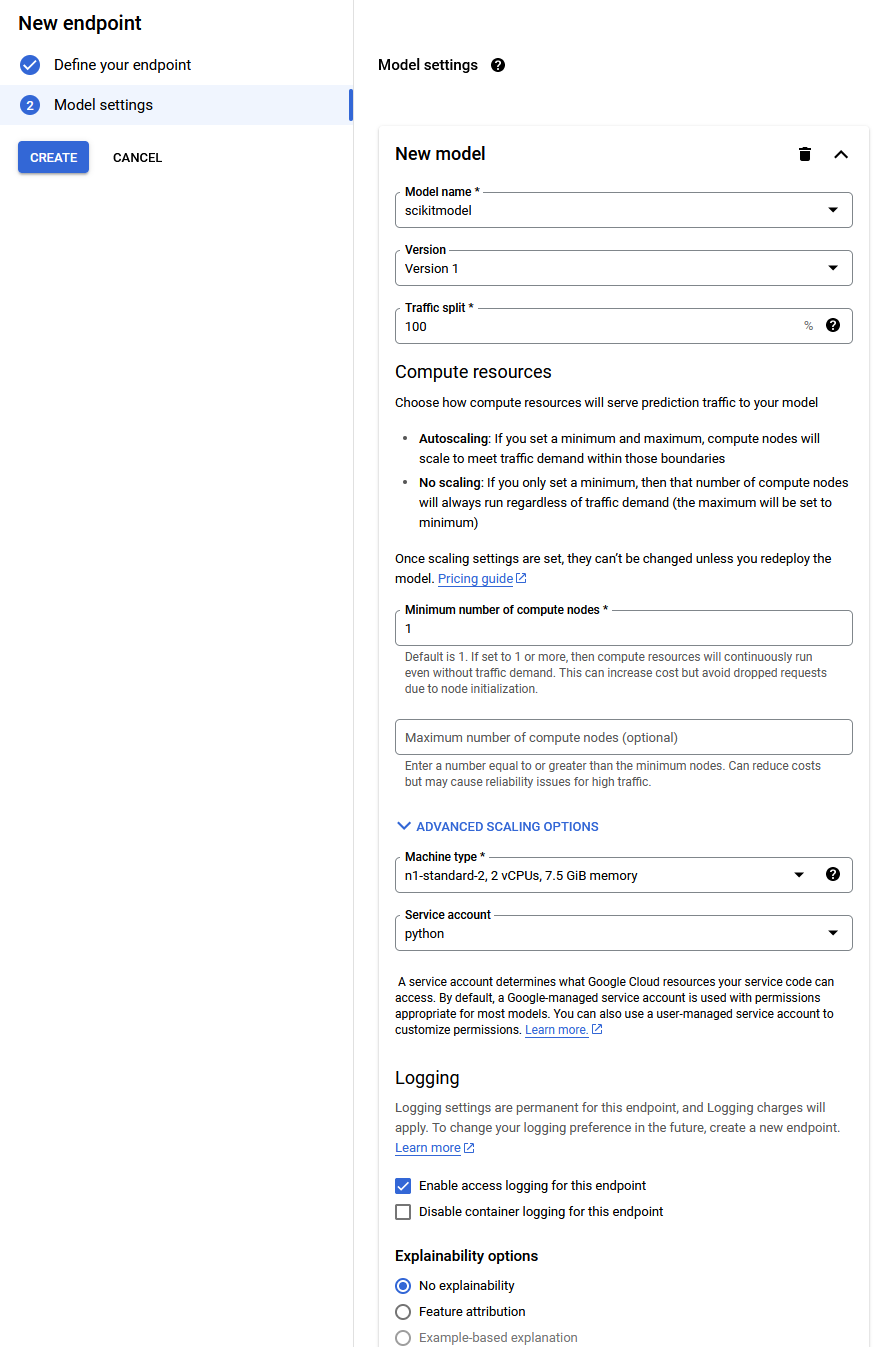


Y obtenemos el modelo creado:

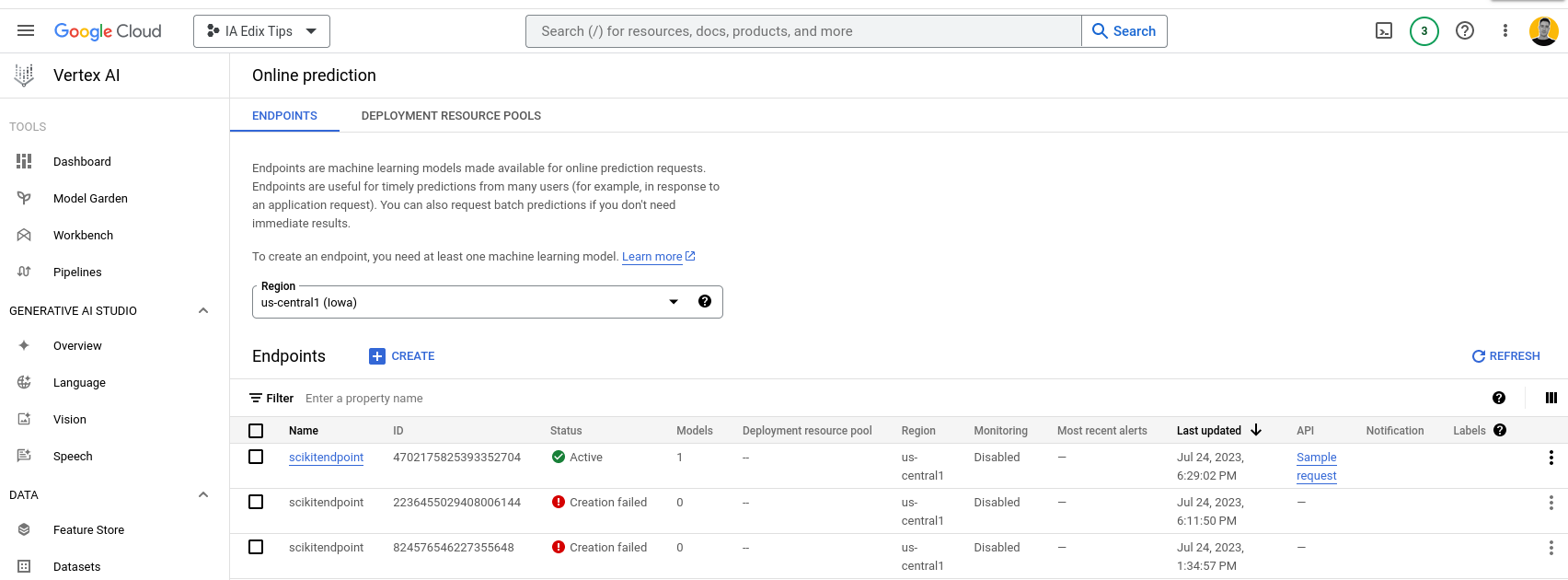


# **Despliegue del modelo**

Para ello crearemos un endpoint el cual será utilizado de forma remota. En este caso he tenido que utilizar la versión 1.2.0 de scikit-learn para que a la hora del despliegue no apareciesen errores.

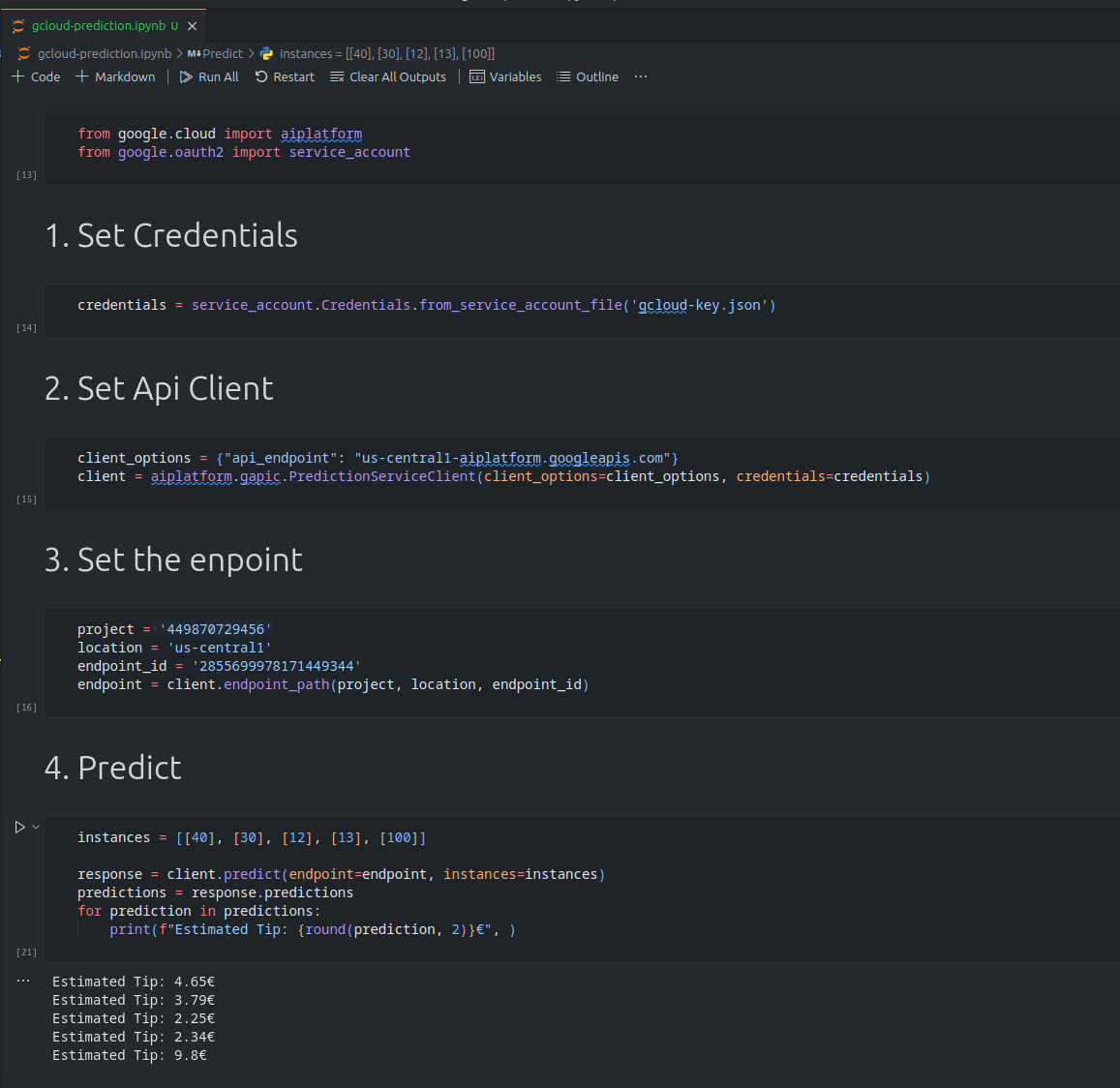


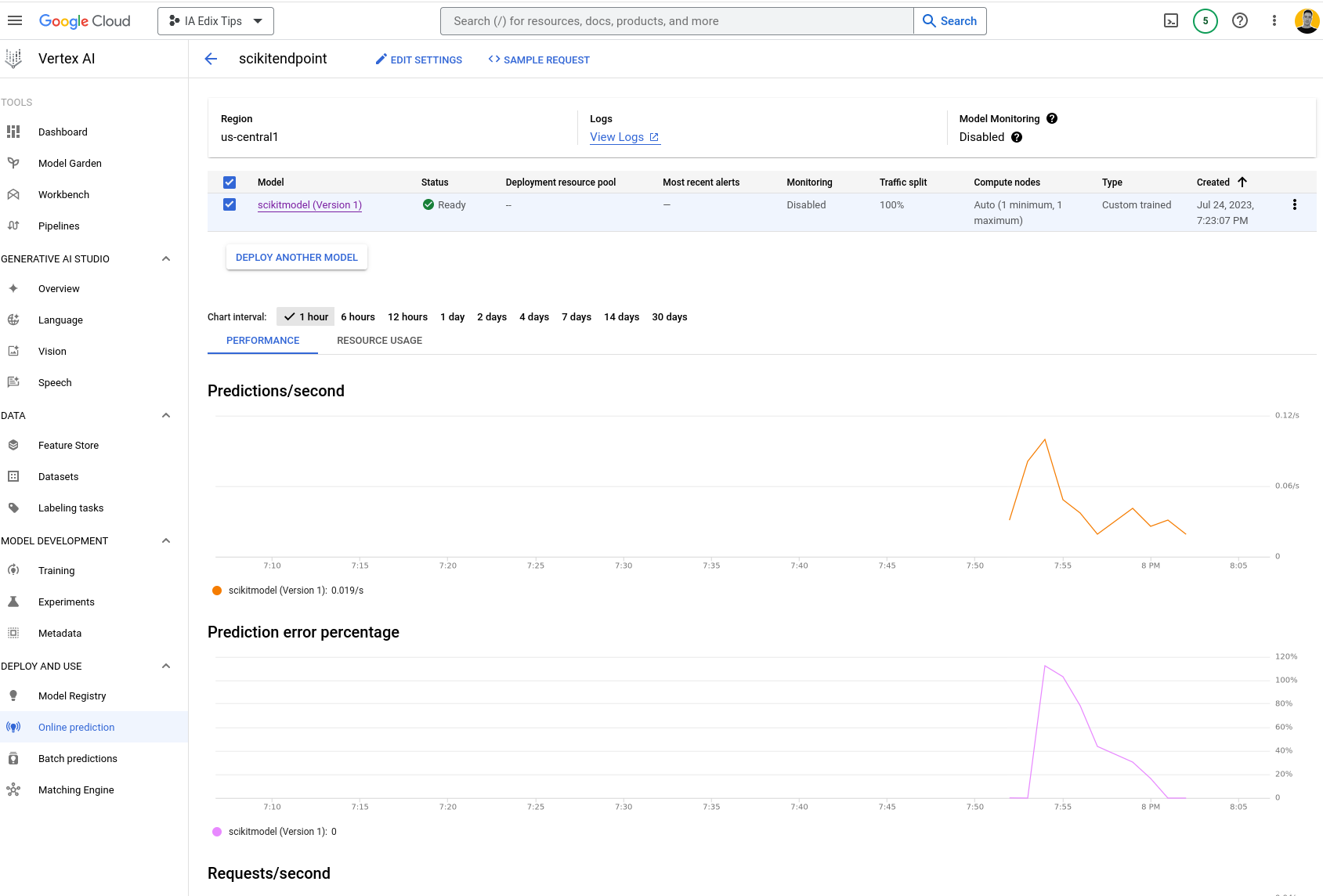
En la siguiente tabla aparecen varios errores, esto se debe a que estaba tratando de desplegar un modelo generado con la versión 1.3.0 de scikit-learn, por lo que al cambiar de versión ya pude desplegar el enpoint correctamente.



# **Utilización del modelo**

En este caso he optado por utilizar un fichero de Jupyter Notebook, por lo que es más fácil de ver la ejecución paso a paso del script.



Y aquí un ejemplo de las métricas que se han generado por el uso del endpoint: