**Experimentos en MLFlow en una máquina virtual - para el sistema de recomendación**

El modelo utiliza técnicas de procesamiento de texto y similitud de coseno para recomendar noticias similares a una noticia dada. La similitud de coseno permite medir la similitud entre el contenido de las noticias, y las recomendaciones se generan en función de esa similitud.

**1. Lanzamiento de una instancia**

Se crea la instancia en AWS EC2, con una máquina t2.medium, con sistema operativo

Ubuntu y 20GB de disco. Con la ip privada ip-172-31-95-21.ec2.internal y publica 18.208.149.226.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**2. Se abren los puertos de**

**2. Conexión a la máquina:**

a) en la terminal del equipo se ejecutan los siguientes comandos

cd Downloads

ssh -i entrena.pem ubuntu@18.208.149.226

sudo apt update

sudo apt install python3-pip

pip install scikit-learn

pip install mlflow

pip install unidecode

pip3 install spacy

!python -m spacy download es\_core\_news\_sm

PATH=$PATH:/home/ubuntu/.local/bin

**3. Subir los archivos necesarios para la ejecución a la maquina virtual.**

En una terminal aparte, se sube el archivo mlflow-diab.py y los archivos de fuente (Adjuntos):

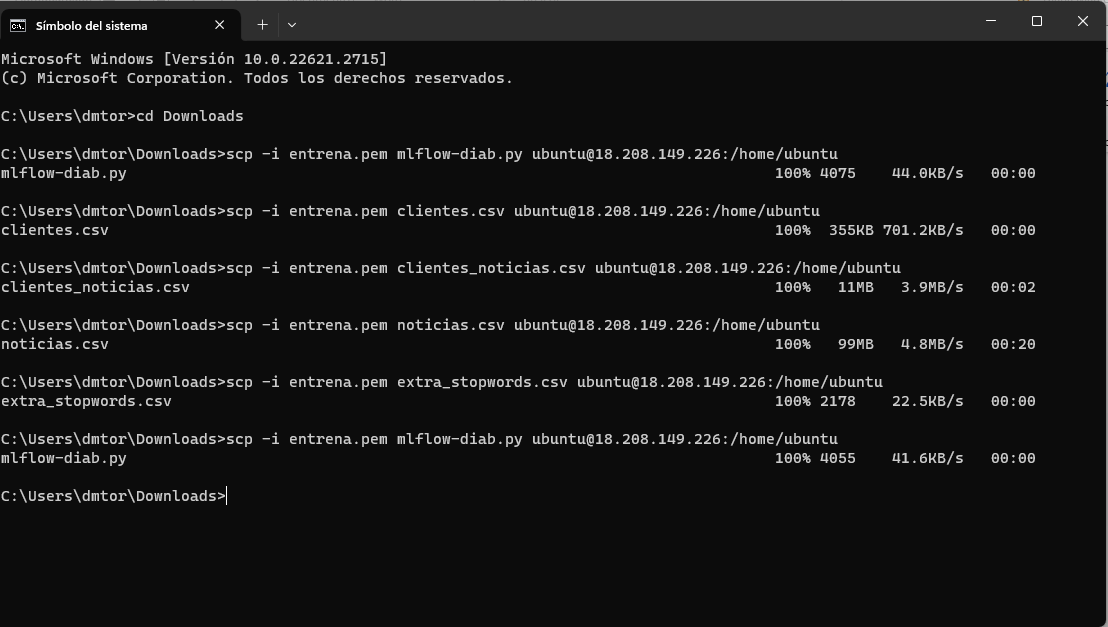
scp -i entrena.pem mlflow-diab.py ubuntu@18.208.149.226:/home/ubuntu

scp -i entrena.pem clientes.csv ubuntu@18.208.149.226:/home/ubuntu

scp -i entrena.pem clientes\_noticias.csv ubuntu@18.208.149.226:/home/ubuntu

scp -i entrena.pem noticias.csv ubuntu@18.208.149.226:/home/ubuntu

scp -i entrena.pem extra\_stopwords.csv ubuntu@18.208.149.226:/home/ubuntu



**4. Descripción de código mlflow-diab.py (Adjunto)**

Este script utiliza MLflow para realizar un seguimiento de experimentos y registra la información relacionada con un sistema de recomendación basado en contenido. Aquí está el flujo de trabajo del código:

* Carga de Datos: Se cargan los conjuntos de datos clientes, clientes\_noticias, noticias, y extra\_stopwords desde archivos CSV.
* Procesamiento de Datos: Se realiza la fusión de datos y se realiza la manipulación de los datos para obtener un DataFrame consolidado llamado df\_news.
* Procesamiento de Texto: Se define la función text\_cleaning para limpiar el texto de las noticias.
* Se aplica la función de limpieza al texto de las noticias y se crea una nueva columna llamada news\_text\_content\_clean con el texto limpio.
* CountVectorizer: Se utiliza CountVectorizer de scikit-learn para convertir el texto limpio de las noticias en una matriz de términos (conteo de palabras).
* Se calcula la matriz de similitud de coseno entre las noticias basándose en la matriz de términos generada.
* Función de Recomendación (recomendador): Se define la función recomendador que toma el título de una noticia como entrada y devuelve una lista de recomendaciones de noticias similares.
* Registra Experimento con MLflow: Se registra el experimento en MLflow utilizando mlflow.set\_experiment.
* Se inicia un nuevo run con mlflow.start\_run.
* Se registra información relevante como parámetros, recomendaciones, y subsecciones en MLflow con mlflow.log\_param.
* Artefactos de MLflow: La matriz de similitud de coseno se guarda como un artefacto (archivo CSV) en el experimento de MLflow con mlflow.log\_artifact.
* Impresión de Resultados: Se imprime la lista de recomendaciones, subsecciones y puntajes de similitud para verificar.
* mlflow.log\_param : Se utiliza para registrar parámetros específicos en el experimento de MLflow. Estos parámetros son valores que pueden ser de interés para el seguimiento y comparación de diferentes ejecuciones del experimento. En este caso, los parámetros que se registran son
  + *target\_news\_title:* El título de la noticia que se utiliza como objetivo para el sistema de recomendación.
  + *recommendations:* La lista de noticias recomendadas generada por la función recomendador.
  + *subsec:* La lista de subsecciones correspondientes a las noticias recomendadas.

Estos parámetros se registran en el contexto de un experimento específico para mantener un historial y facilitar la comparación entre diferentes ejecuciones del sistema de recomendación.

Se puede visualizar estos parámetros y sus valores en la interfaz de usuario de MLflow para analizar y comparar los resultados de diferentes experimentos.

**5. Exponer el servicio**

En la terminal de la máquina se ejecuta mlflow como servidor y se expone el servicio por el puerto 8050.

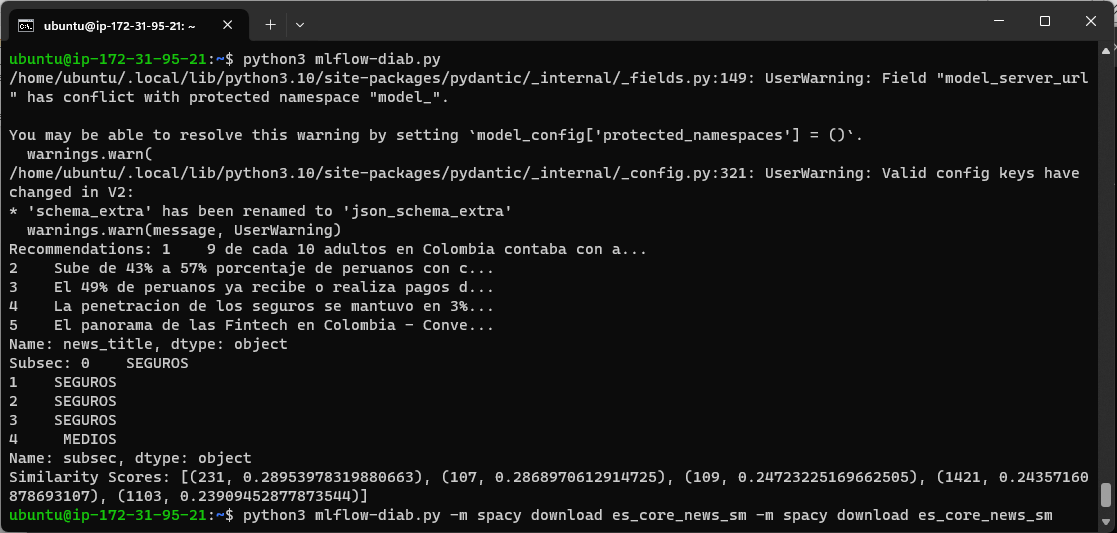
mlflow server -h 0.0.0.0 -p 8050

**6. Se ejecuta el archivo de python para entrenar y evaluar el modelo (en una nueva terminal conectada a la máquina virtual)**

cd Downloads

ssh -i entrena.pem ubuntu@18.208.149.226

python3 mlflow-diab.py



Se visualizan los resultados en mlflow:

http://18.208.149.226:8050

Interfaz de usuario gráfica, Aplicación, Correo electrónico

Descripción generada automáticamente

Se observa la noticia inicial, las recomendaciones realizadas y el sector al que pertenece:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

13. Se modifica alguno de los parámetros del modelo y vuelva a correr el script. Para ello puede usar

nano mlflow-diab.py

Texto

Descripción generada automáticamente

