TAREA 3 - SPARK MACHINE LEARNING Y POSTGRES - BIG DATA

Estudiante: Yoksan Varela Cambronero

10 de JuLio, 2024

1. Contenido del archivo comprimido

Además de este archivo, los documentos contenidos en el archivo comprimido son:

1.1. Folder datasets

Contiene el dataset usado en el Jupyter Notebook, llamado *pet_adoption_data.csv*. Cabe la pena mencionar que el header original de este archivo fue eliminado para poder asignarle uno personalizado.

1.2. Archivo Tarea3.pdf

Enunciado de la tarea provisto por el profesor.

1.3. Archivo build_run_docker.bat

Este documento fue creado para facilitar la creación y la ejecución del contenedor de Docker en mi computadora personal, la cual cuenta con el OS Windows 11. Este .bat es distinto al usado anteriormente

1. *docker build –tag tarea3_yoksanvarela .*: Comando para construir el contenedor base.
2. *docker run –name tarea3-db -e POSTGRES_PASSWORD=testPassword -p 5433:5432 -d postgres*: Comando para correr un contenedor de POSTGRES, el cual mapea el puerto 5432 del contenedor al puerto 5433 de mi computadora personal.
3. *docker run -p 8888:8888 -i -t tarea3_yoksanvarela /bin/bash*: Comando para correr el contenedor base instalado anteriormente, y asignar el puerto 8888 del contenedor al puerto 8888 de mi computadora personal. Esto es necesario para poder levantar Jupyter dentro del contenedor pero ser manipulado desde el browser local.

1.4. Archivo command_list.txt

Es un archivo plano de texto que contiene las líneas necesarias para poder correr las diferentes partes de esta tarea. Cabe la pena rescatar que estos comandos hay que correrlos de forma manual en la consola del contenedor de Docker o bien dentro de una consola del contenedor de POSTGRES (en mi caso, estoy accediendo esa consola directamente en Docker Desktop). Además de los comandos del .bat explicado anteriormente, también contiene las siguientes lineas para ejecutar:

1. *jupyter notebook –ip=0.0.0.0 –port=8888 –allow-root*: Levanta una sesión de Jupyter en el contenedor, el cual se accede a través del puerto 8888 (ya mapeado anteriormente). Esta línea se ejecuta en el contenedor base.
2. *psql -h host.docker.internal -p 5433 -U postgres*: Comando para poder manipular la base de datos postgres dentro del contenedor de creado para ese propósito. Este comando debe correrse dentro de la consola del contenedor de POSTGRES, y a la hora de solicitar la contraseña (o password) se usa *testPassword*.

1.5. Archivo Dockerfile

Es la imagen usada en Docker. Es la versión renombrada de Dockerfile_full, la cual fue provista por el profesor a través de GITHUB.

1.6. Archivo postgresql-42.2.14.jar

Es el archivo comprimido con toda la parte utilitaria de POSTGRES, también provisto por el profesor a través de GITHUB.

1.7. tarea3_YoksanVarelaCambronero.ipynb

Es el cuaderno de Jupyter con solución a esta tarea. El mismo es auto-contenido, lo que implica que cuenta con comentarios y Markdowns explicando el desarrollo de la solución.

2. Ejecución de los archivos de la tarea

Para poder ejecutar la solución, es necesario correr los siguientes comando de forma secuencial en una terminal corriendo en el directorio de esta tarea:

1. docker build –tag tarea3_yoksanvarela .
2. docker run –name tarea3-db -e POSTGRES_PASSWORD=testPassword -p 5433:5432 -d postgres
3. docker run -p 8888:8888 -i -t tarea3_yoksanvarela /bin/bash

Una vez ejecutados estos comandos, se corre el siguiente comando dentro de la terminal del contenedor base:

1. jupyter notebook –ip=0.0.0.0 –port=8888 –allow-root

Esto va a levantar Jupyter en el contenedor. Para poder accederlo, es necesario tocar el hipervínculo que aparece al final de la ejecución: http://127.0.0.1:8888/?token=5b733a89489f1ed9306587932161b6cabaf9900bbfdb0b67 (nota: no tengo certeza que este hipervínculo sea el mismo entre diferentes computadoras)

Una vez concluido con esto, ya se puede ejecutar el cuaderno de Jupyter de esta tarea:
tarea3_YoksanVarelaCambronero.ipynb

Finalmente, en otra terminal corriendo dentro del contenedor de POSTGRES (en mi caso, accedí a esta terminal directamente dentro de Docker Desktop), ejecutar lo siguiente:

1. psql -h host.docker.internal -p 5433 -U postgres (Cuando se pregunta por el password, introducir testPassword y presionar la tecla enter).

Esto es necesario para acceder a las bases de datos POSTGRES creadas durante el desarrollo de la solución, y son las solicitadas en la tarea a excepción de una extra que fue creada de mi parte: *tarea3_raw*. Esta base de datos tiene los atributos categóricos con su valor original sin codificar, para efectos de lectura consideré que es mejor contar con esa información. Para accesar cualquiera de las bases de datos, usar el comando **SELECT * FROM *nombre de la base de datos a accesar*;**. Por ejemplo, para acceder la base de datos tarea3_raw, se ejecuta dentro del contenedor POSTGRES *SELECT * FROM tarea3_raw;*.