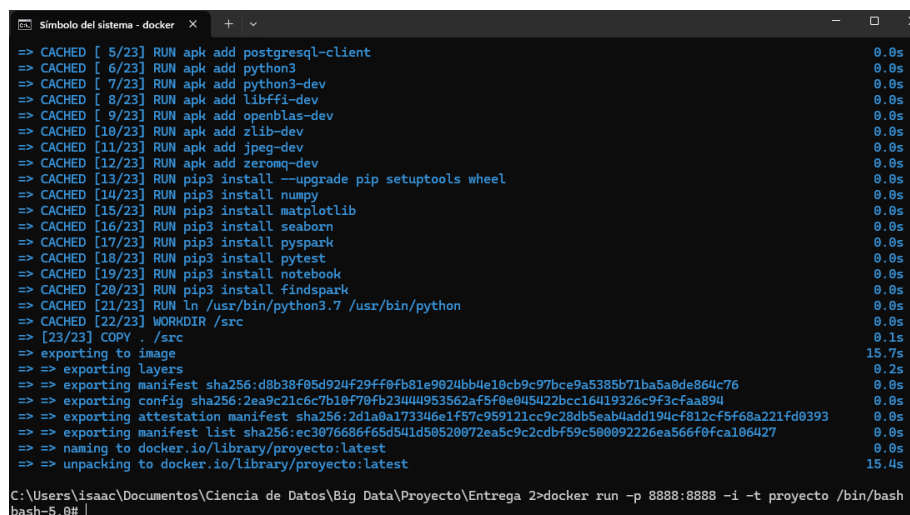


Guía para el Código Principal

Isaac Brenes

1. Descargar y descomprimir el zip en un directorio local
2. Abra el command prompt del sistema y busque la dirección en la que guardo la carpeta. Debe nombrarla “proyecto”
3. Debe crear la imagen en Docker. Utilice el comando **<docker build --tag proyecto .>**. Este comando también está disponible en el archivo “build_image.sh”.
4. Debe correr la imagen. Utilice el comando **<docker run -p 8888:8888 -i -t proyecto /bin/bash>** o bien utilice el archivo “run_image.sh”. No cierre este prompt ya que será utilizado luego.



```
=> CACHED [ 5/23] RUN apk add postgresql-client 0.0s
=> CACHED [ 6/23] RUN apk add python3 0.0s
=> CACHED [ 7/23] RUN apk add python3-dev 0.0s
=> CACHED [ 8/23] RUN apk add libffi-dev 0.0s
=> CACHED [ 9/23] RUN apk add openblas-dev 0.0s
=> CACHED [10/23] RUN apk add zlib-dev 0.0s
=> CACHED [11/23] RUN apk add jpeg-dev 0.0s
=> CACHED [12/23] RUN apk add zeromq-dev 0.0s
=> CACHED [13/23] RUN pip3 install --upgrade pip setuptools wheel 0.0s
=> CACHED [14/23] RUN pip3 install numpy 0.0s
=> CACHED [15/23] RUN pip3 install matplotlib 0.0s
=> CACHED [16/23] RUN pip3 install seaborn 0.0s
=> CACHED [17/23] RUN pip3 install pyspark 0.0s
=> CACHED [18/23] RUN pip3 install pytest 0.0s
=> CACHED [19/23] RUN pip3 install notebook 0.0s
=> CACHED [20/23] RUN pip3 install findspark 0.0s
=> CACHED [21/23] RUN ln /usr/bin/python3.7 /usr/bin/python 0.0s
=> CACHED [22/23] WORKDIR /src 0.0s
=> [23/23] COPY . /src 0.1s
=> exporting to image 15.7s
=> exporting layers 0.2s
=> exporting manifest sha256:d8b38f05d924f29ff0fb81e9024bb4e10cb9c97bce9a5385b71ba5a0de864c76 0.0s
=> exporting config sha256:2ea9c21c6c7b10f70fb23444953562af5f0e045422bcc16419326c9f3cfaa894 0.0s
=> exporting attestation manifest sha256:2d1a0a173346e1f87c9959121cc9c28db5eab4add194cf812cf5f68a221fd0393 0.0s
=> exporting manifest list sha256:ec3076686f65d541d50520072ea5c9c2cdbc59c500092226ea566f0fca106427 0.0s
=> naming to docker.io/library/proyecto:latest 0.0s
=> unpacking to docker.io/library/proyecto:latest 15.4s

C:\Users\isaac\Documentos\Ciencia de Datos\Big Data\Proyecto\Entrega 2>docker run -p 8888:8888 -i -t proyecto /bin/bash
bash-5.0#
```

5. Puede ejecutar las pruebas unitarias utilizando el comando **<pytest -s>**
6. Abra un **nuevo command prompt** del sistema y obtenga la dirección de la carpeta “*posgresql*” contenida dentro de la carpeta principal. Utilice el comando **<docker run --name bigdata-db -e POSTGRES_PASSWORD=testPassword -p 5433:5432 -d postgres>** o bien utilice el archivo “run_image.sh” disponible dentro de esa carpeta. Puede cerrar este prompt una vez creado el contenedor.

```
Microsoft Windows [Versión 10.0.26100.4770]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\isaac>cd C:\Users\isaac\Documentos\Ciencia de Datos\Big Data\Proyecto\Entrega 2\posgresql

C:\Users\isaac\Documentos\Ciencia de Datos\Big Data\Proyecto\Entrega 2\posgresql>docker run --name bigdata-db -e POSTGRES_PASSWORD=testPassword -p 5433:5432 -d postgres
```

7. Vuelva al prompt inicial. Utilice el comando **<psql -h host.docker.internal -p 5433 -U postgres>** para cargar posgres en el contenedor. También puede utilizar el archivo “connect_db_from_spark_container” disponible dentro de la carpeta. La contraseña es **testPassword**.

```
Dataframe esperado
+-----+-----+-----+-----+
| ID | Calidad_profesor | Actividad_fisica | Nota |
+-----+-----+-----+-----+
| 1 | 3 | 2 | 100 |
| 2 | 1 | 1 | 10 |
| 3 | 2 | 5 | 90 |
| 4 | 1 | 0 | 72 |
| 5 | 2 | 6 | 80 |
+-----+-----+-----+-----+

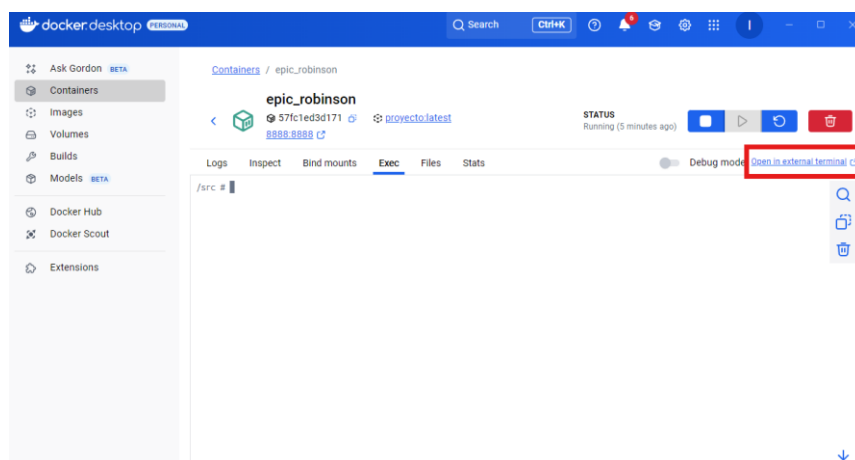
===== warnings summary =====
test_unit_tests.py::test_join_pass
/usr/lib/python3.7/site-packages/pyspark/context.py:317: FutureWarning: Python 3.7 support is deprecated in Spark 3.4.
warnings.warn("Python 3.7 support is deprecated in Spark 3.4.", FutureWarning)

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 1 passed, 1 warning in 10.86s =====

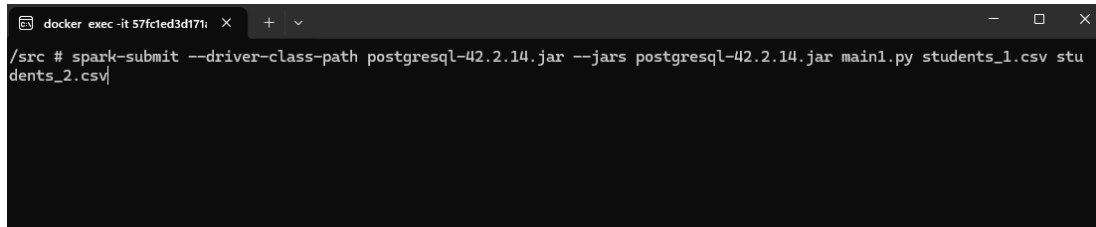
bash-5.0# psql -h host.docker.internal -p 5433 -U postgres
Password for user postgres:
psql (11.12, server 17.5 (Debian 17.5-1.pgdgl20+1))
WARNING: psql major version 11, server major version 17.
Some psql features might not work.
Type "help" for help.

postgres=#
```

8. Ahora dirijase al contenedor en Docker y abra la terminal (externa si lo desea).

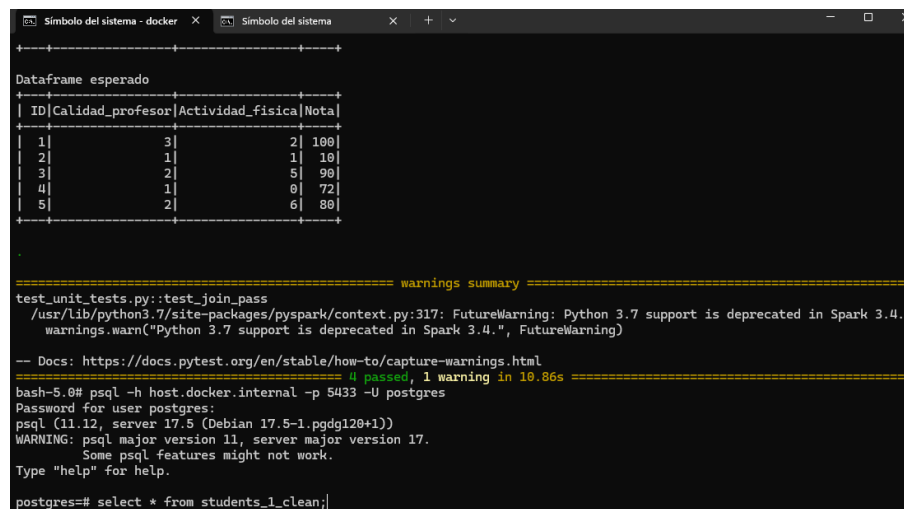


9. Para ejecutar el archivo main1.py en donde se realiza la limpieza de los dos conjuntos de datos y se suben a postgres, ejecute este comando en la terminal que acaba de abrir **<spark-submit --driver-class-path postgresql-42.2.14.jar --jars postgresql-42.2.14.jar main1.py students_1.csv students_2.csv>**



```
docker exec -it 57fcted3d174 /src # spark-submit --driver-class-path postgresql-42.2.14.jar --jars postgresql-42.2.14.jar main1.py students_1.csv students_2.csv
```

10. Si desea verificar los datos en postgres, vaya a la primera terminal que abrió y que está conectada con postgres, y ejecute **<select * from students_1_clean;>** para abrir el primer dataset preprocesado, o **<select * from students_2_clean;>** para ver el segundo.



```
Simbolo del sistema - docker x Simbolo del sistema x + v

Dataframe esperado
+-----+
| ID | Calidad_profesor | Actividad_fisica | Nota |
+-----+
| 1 | 3 | 2 | 100 |
| 2 | 1 | 1 | 10 |
| 3 | 2 | 5 | 90 |
| 4 | 1 | 0 | 72 |
| 5 | 2 | 6 | 80 |
+-----+

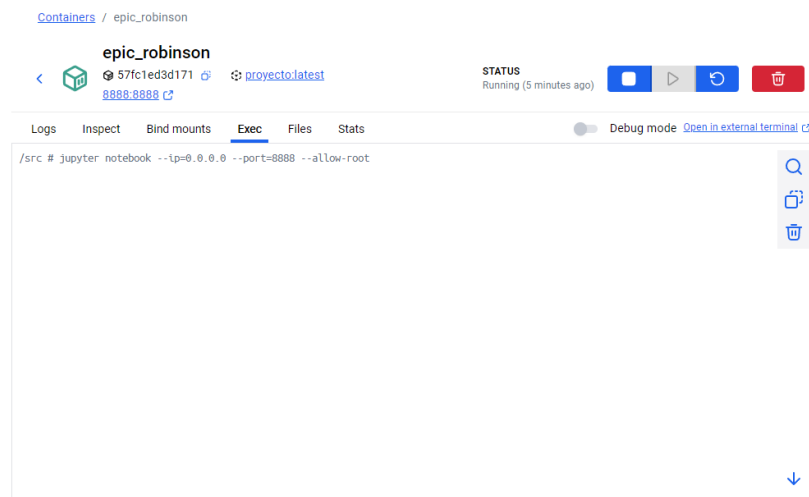
===== warnings summary =====
test_unit_tests.py::test_join_pass
/usr/lib/python3.7/site-packages/pyspark/context.py:317: FutureWarning: Python 3.7 support is deprecated in Spark 3.4.
warnings.warn("Python 3.7 support is deprecated in Spark 3.4.", FutureWarning)

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 0 passed, 1 warning in 10.86s =====

bash-5.0# psql -h host.docker.internal -p 5433 -U postgres
Password for user postgres:
psql (11.12, server 17.5 (Debian 17.5-1.pgdg120+1))
WARNING: psql major version 11, server major version 17.
Some psql features might not work.
Type "help" for help.

postgres=# select * from students_1_clean;
```

11. Para ejecutar los modelos de ML, debe abrir el Jupyter Notebook. Puede utilizar el comando en la terminal del contenedor **<jupyter notebook --ip=0.0.0.0 --port=8888 --allow-root>** también disponible en el archivo "load_jupyter_notebook". Debe acceder utilizando alguno de los enlaces.



12. Abra el archivo llamado “*ML.ipynb*” en donde se termina de preparar los datos para comenzar con los entrenamientos de los modelos. Al final se encuentra el análisis de resultados.