Práctica 5

1. Crea un notebook de **Sagemaker** (ver capturas más adelante)
2. Instala **openjdk** y **pyspark** desde la terminal (ver capturas más adelante)
3. Sube el archivo **STAR.csv**
4. Crea una base de datos en formato delta y asóciala a un dataframe de Spark (usa el notebook **P5-ayuda.ipynb** proporcionado en el material de la práctica)
5. Cambia a double el tipo de las columnas read/math de la forma siguiente:

   ```
   from pyspark.sql.types import DoubleType
   df = df.withColumn("readk",df["readk"].cast(DoubleType()))
   df = df.withColumn("read1",df["read1"].cast(DoubleType()))
   df = df.withColumn("read2",df["read2"].cast(DoubleType()))
   df = df.withColumn("read3",df["read3"].cast(DoubleType()))
   df = df.withColumn("mathk",df["mathk"].cast(DoubleType()))
   df = df.withColumn("math1",df["math1"].cast(DoubleType()))
   df = df.withColumn("math2",df["math2"].cast(DoubleType()))
   df = df.withColumn("math3",df["math3"].cast(DoubleType()))
   df = df.withColumn("experiencek",df["experiencek"].cast(DoubleType()))
   df = df.withColumn("experience1",df["experience1"].cast(DoubleType()))
   df = df.withColumn("experience2",df["experience2"].cast(DoubleType()))
   df = df.withColumn("experience3",df["experience3"].cast(DoubleType()))
   ```

6. Codifica las variables categóricas de la forma siguiente (este es un ejemplo para el jardín de infancia, hay que adaptarlo también a primero, segundo y tercero):

   ```
   from pyspark.ml import Pipeline
   from pyspark.ml.feature import StringIndexer,OneHotEncoder
   colk =
   ['gender','ethnicity','birth','stark','lunchk','schoolk','degreek','ladder
   k','tethnicityk','systemk','schoolidk']
   colkI =
   ['genderI','ethnicityI','birthI','starkI','lunchkI','schoolkI','degreekI',
   'ladderkI','tethnicitykI','systemkI','schoolidkI']
   colkE =
   ['genderE','ethnicityE','birthE','starkE','lunchkE','schoolkE','degreekE',
   'ladderkE','tethnicitykE','systemkE','schoolidkE']
   indexer = StringIndexer(inputCols=colk, outputCols=colkI)
   encoder = OneHotEncoder(inputCols=colkI, outputCols=colkE)
   pipeline = Pipeline(stages=[indexer,encoder])
   df = pipeline.fit(df).transform(df)
   ```

7. Dibuja cuatro diagramas de cajas que relacionen el número de alumnos en el aula con los niveles de lectura para jardín de infancia, primer curso, segundo curso y tercer curso. A la vista de los diagramas, ¿crees que el tamaño de la clase influye en el nivel de lectura para todos los cursos?
8. Dibuja cuatro diagramas de cajas que relacionen el número de alumnos en el aula con los niveles de matemáticas para jardín de infancia, primer curso, segundo curso y tercer curso. A la vista de los diagramas, ¿crees que el tamaño de la clase influye en el nivel de matemáticas para todos los cursos?
9. Haz un modelo de regresión lineal que relacione el nivel de lectura en el jardín de infancia con las variables de la lista siguiente:

   ```
   colinputk = ['genderI','ethnicityI','birthI','starkI','lunchkI',
   'schoolkI','degreekI','ladderkI','tethnicitykI']
   ```

Para realizar este modelo, haz los siguientes pasos:

- Divide con `randomSplit` los datos en dos partes: 75% para entrenamiento y 25% para test
- Crea un pipeline con tres etapas y aplícasela al conjunto de entrenamiento:
  - Etapa 1: con `pyspark.ml.feature.Imputer` rellena los valores perdidos de la variable de salida (`strategy="mean"`) en una variable llamada `myreadk_imputado`"
  - Etapa 2: con `pyspark.ml.feature.VectorAssembler` combina todas las variables de entrada en una columna llamada "`myfeatures`"
  - Etapa 3: con `pyspark.ml.regression.LinearRegression` ajusta un modelo a las variables de entrada "`myfeatures`" y de salida "`myreadk_imputado`"

Por último, calcula error cuadrático medio y R2 en el conjunto de test.

12. Repite el modelo para primer curso, segundo y tercero, usando las variables necesarias para ello
13. Haz cuatro modelos lineales que relacionen el nivel de matemáticas con las variables correspondientes en los cuatro cursos
14. Repite los apartados 11 a 13 con Random Forest y compara los resultados de la regresión lineal y del random forest.

Parte opcional:

15. ¿Podrías ordenar las variables según su importancia en relación con las capacidades de lectura y escritura en jardín de infancia, primero, segundo y tercero? La solución más sencilla consiste en pasar el dataframe Spark a Pandas y hacer el cálculo en un nodo; ¿se te ocurre cómo hacerlo sin convertir el dataframe a Pandas, para poder aplicar la solución a una base de datos de gran tamaño?

Práctica 5

Anexo: Información acerca del dataset STAR

Project STAR: Student-Teacher Achievement Ratio

Description

The Project STAR public access data set, assessing the effect of reducing class size on test scores in the early grades.

A data frame containing 11,598 observations on 47 variables.

gender
factor indicating student's gender.

ethnicity
factor indicating student's ethnicity with levels "cauc" (Caucasian), "afam" (African-American), "asian" (Asian), "hispanic" (Hispanic), "amindian" (American-Indian) or "other".

birth
student's birth quarter (of class yearqtr).

stark
factor indicating the STAR class type in kindergarten: regular, small, or regular-with-aide. NA indicates that no STAR class was attended.

star1
factor indicating the STAR class type in 1st grade: regular, small, or regular-with-aide. NA indicates that no STAR class was attended.

star2
factor indicating the STAR class type in 2nd grade: regular, small, or regular-with-aide. NA indicates that no STAR class was attended.

star3
factor indicating the STAR class type in 3rd grade: regular, small, or regular-with-aide. NA indicates that no STAR class was attended.

readk
total reading scaled score in kindergarten.

read1
total reading scaled score in 1st grade.

read2
total reading scaled score in 2nd grade.

read3
total reading scaled score in 3rd grade.

mathk
total math scaled score in kindergarten.

math1
total math scaled score in 1st grade.

math2

total math scaled score in 2nd grade.

math3
total math scaled score in 3rd grade.

lunchk
factor indicating whether the student qualified for free lunch in kindergarten.

lunch1
factor indicating whether the student qualified for free lunch in 1st grade.

lunch2
factor indicating whether the student qualified for free lunch in 2nd grade.

lunch3
factor indicating whether the student qualified for free lunch in 3rd grade.

schoolk
factor indicating school type in kindergarten: "inner-city", "suburban", "rural" or "urban".

school1
factor indicating school type in 1st grade: "inner-city", "suburban", "rural" or "urban".

school2
factor indicating school type in 2nd grade: "inner-city", "suburban", "rural" or "urban".

school3
factor indicating school type in 3rd grade: "inner-city", "suburban", "rural" or "urban".

degreek
factor indicating highest degree of kindergarten teacher: "bachelor", "master", "specialist", or "master+".

degree1
factor indicating highest degree of 1st grade teacher: "bachelor", "master", "specialist", or "phd".

degree2
factor indicating highest degree of 2nd grade teacher: "bachelor", "master", "specialist", or "phd".

degree3
factor indicating highest degree of 3rd grade teacher: "bachelor", "master", "specialist", or "phd".

ladderk
factor indicating teacher's career ladder level in kindergarten: "level1", "level2", "level3", "apprentice", "probation" or "pending".

ladder1
factor indicating teacher's career ladder level in 1st grade: "level1", "level2", "level3", "apprentice", "probation" or "noladder".

ladder2
factor indicating teacher's career ladder level in 2nd grade: "level1", "level2", "level3", "apprentice", "probation" or "noladder".

Práctica 5

ladder3
factor indicating teacher's career ladder level in 3rd grade: "level1", "level2", "level3", "apprentice", "probation" or "noladder".

experiencek
years of teacher's total teaching experience in kindergarten.

experience1
years of teacher's total teaching experience in 1st grade.

experience2
years of teacher's total teaching experience in 2nd grade.

experience3
years of teacher's total teaching experience in 3rd grade.

tethnicityk
factor indicating teacher's ethnicity in kindergarten with levels "cauc" (Caucasian) or "afam" (African-American).

tethnicity1
factor indicating teacher's ethnicity in 1st grade with levels "cauc" (Caucasian) or "afam" (African-American).

tethnicity2
factor indicating teacher's ethnicity in 2nd grade with levels "cauc" (Caucasian) or "afam" (African-American).

tethnicity3
factor indicating teacher's ethnicity in 3rd grade with levels "cauc" (Caucasian), "afam" (African-American), or "asian" (Asian).

systemk
factor indicating school system ID in kindergarten.

system1
factor indicating school system ID in 1st grade.

system2
factor indicating school system ID in 2nd grade.

system3
factor indicating school system ID in 3rd grade.

schoolidk
factor indicating school ID in kindergarten.

schoolid1
factor indicating school ID in 1st grade.

schoolid2
factor indicating school ID in 2nd grade.

schoolid3
factor indicating school ID in 3rd grade.

Details

Práctica 5

Project STAR (Student/Teacher Achievement Ratio) was a four-year longitudinal class-size study funded by the Tennessee General Assembly and conducted in the late 1980s by the State Department of Education. Over 7,000 students in 79 schools were randomly assigned into one of three interventions: small class (13 to 17 students per teacher), regular class (22 to 25 students per teacher), and regular-with-aide class (22 to 25 students with a full-time teacher's aide). Classroom teachers were also randomly assigned to the classes they would teach. The interventions were initiated as the students entered school in kindergarten and continued through third grade.

The Project STAR public access data set contains data on test scores, treatment groups, and student and teacher characteristics for the four years of the experiment, from academic year 1985–1986 to academic year 1988–1989. The test score data analyzed in this chapter are the sum of the scores on the math and reading portion of the Stanford Achievement Test.

Stock and Watson (2007) obtained the data set from the Project STAR Web site.

The data is provided in wide format. Reshaping it into long format is illustrated below. Note that the levels of the degree, ladder and tethnicity variables differ slightly between kindergarten and higher grades.

Source

Online complements to Stock and Watson (2007).

References

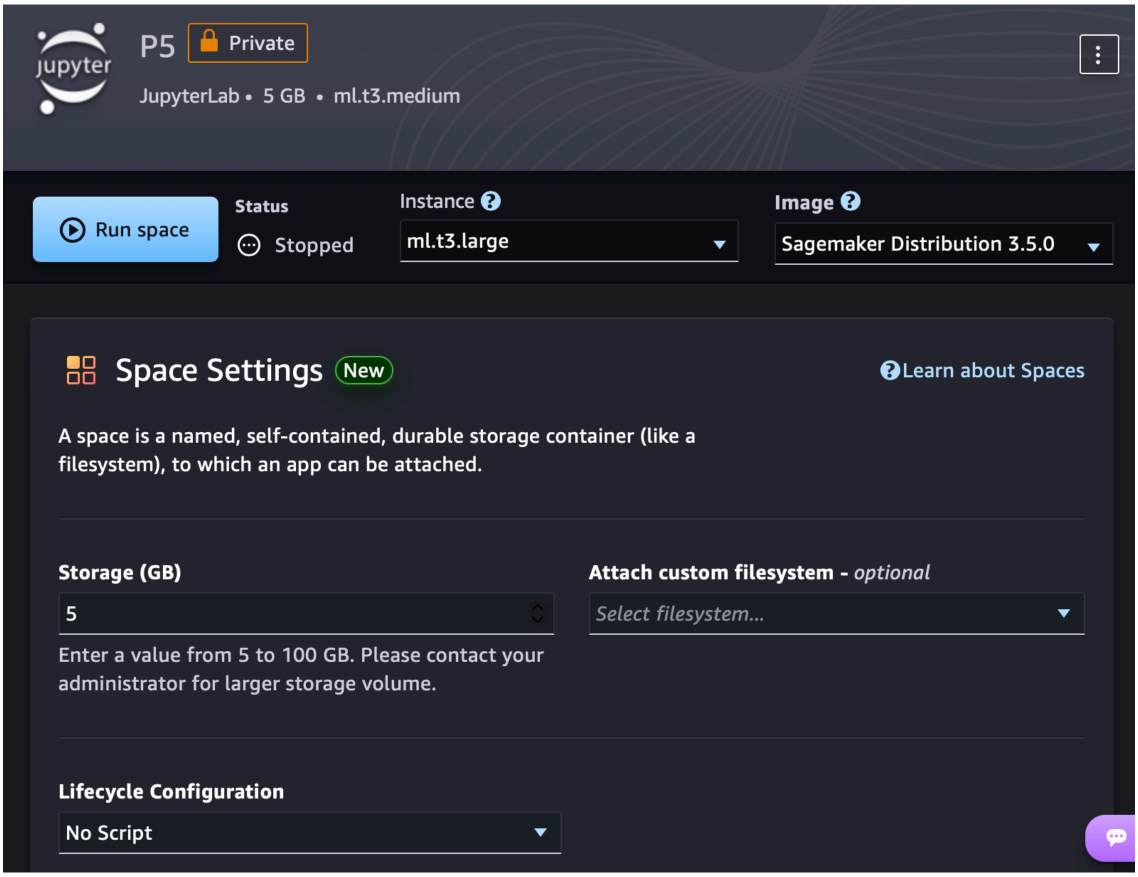
Stock, J.H. and Watson, M.W. (2007). Introduction to Econometrics, 2nd ed. Boston: Addison Wesley.

Práctica 5

Capturas de pantalla con instrucciones:

1) Creación de notebook

2)



P5 Private

JupyterLab • 5 GB • ml.t3.medium

Status: Stopped

Instance: ml.t3.large

Image: Sagemaker Distribution 3.5.0

Space Settings New

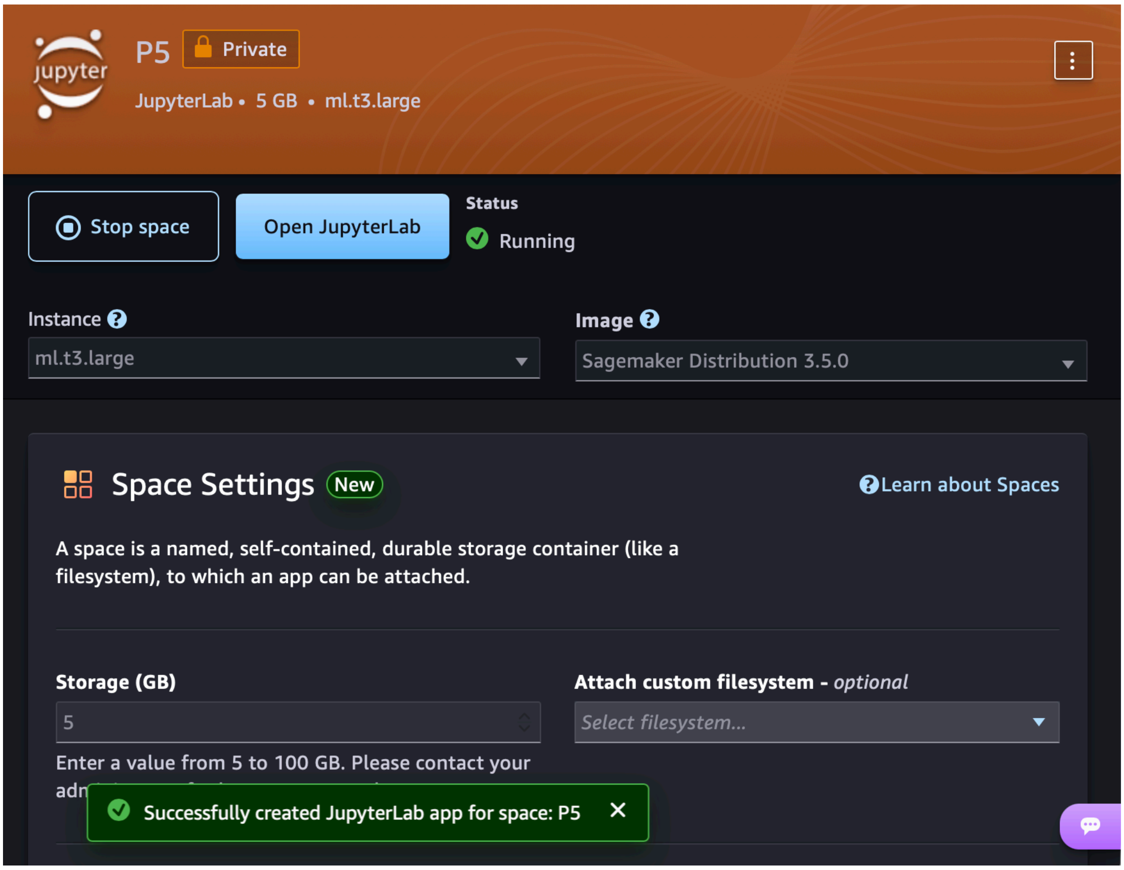
A space is a named, self-contained, durable storage container (like a filesystem), to which an app can be attached.

Storage (GB): 5

Enter a value from 5 to 100 GB. Please contact your administrator for larger storage volume.

Attach custom filesystem - *optional*: *Select filesystem...*

Lifecycle Configuration: No Script



P5 Private

JupyterLab • 5 GB • ml.t3.large

Status: Running

Instance: ml.t3.large

Image: Sagemaker Distribution 3.5.0

Space Settings New

A space is a named, self-contained, durable storage container (like a filesystem), to which an app can be attached.

Storage (GB): 5

Attach custom filesystem - *optional*: *Select filesystem...*

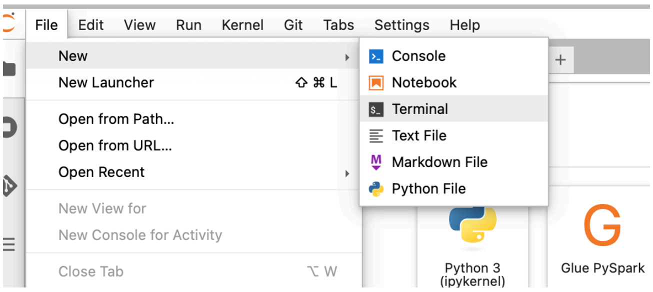
Enter a value from 5 to 100 GB. Please contact your adm

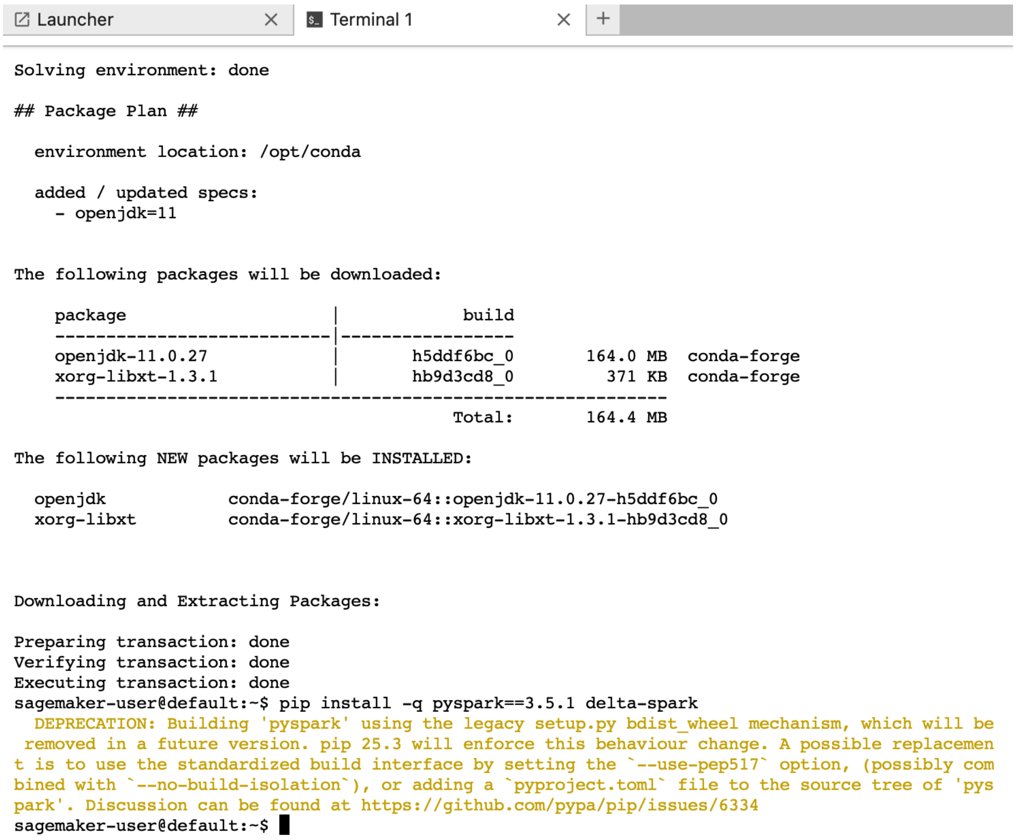
Successfully created JupyterLab app for space: P5

Práctica 5

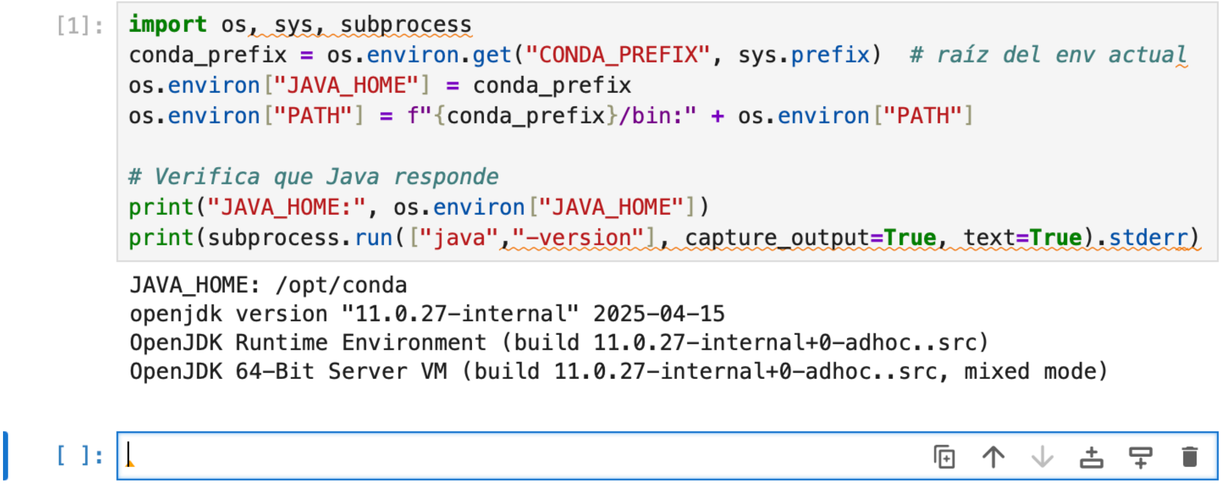
Instalación OpenJDK y PySpark:

```
conda install -y -c conda-forge openjdk=11

pip install -q pyspark==3.5.1 delta-spark
```





```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda

  added / updated specs:
    - openjdk=11


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    openjdk-11.0.27            |       h5ddf6bc_0       164.0 MB  conda-forge
    xorg-libxt-1.3.1           |       hb9d3cd8_0         371 KB  conda-forge
    ------------------------------------------------------------
                                           Total:       164.4 MB

The following NEW packages will be INSTALLED:

  openjdk            conda-forge/linux-64::openjdk-11.0.27-h5ddf6bc_0
  xorg-libxt         conda-forge/linux-64::xorg-libxt-1.3.1-hb9d3cd8_0



Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
sagemaker-user@default:~$ pip install -q pyspark==3.5.1 delta-spark
  DEPRECATION: Building 'pyspark' using the legacy setup.py bdist_wheel mechanism, which will be
 removed in a future version. pip 25.3 will enforce this behaviour change. A possible replacemen
t is to use the standardized build interface by setting the `--use-pep517` option, (possibly com
bined with `--no-build-isolation`), or adding a `pyproject.toml` file to the source tree of 'pys
park'. Discussion can be found at https://github.com/pypa/pip/issues/6334
sagemaker-user@default:~$ 
```

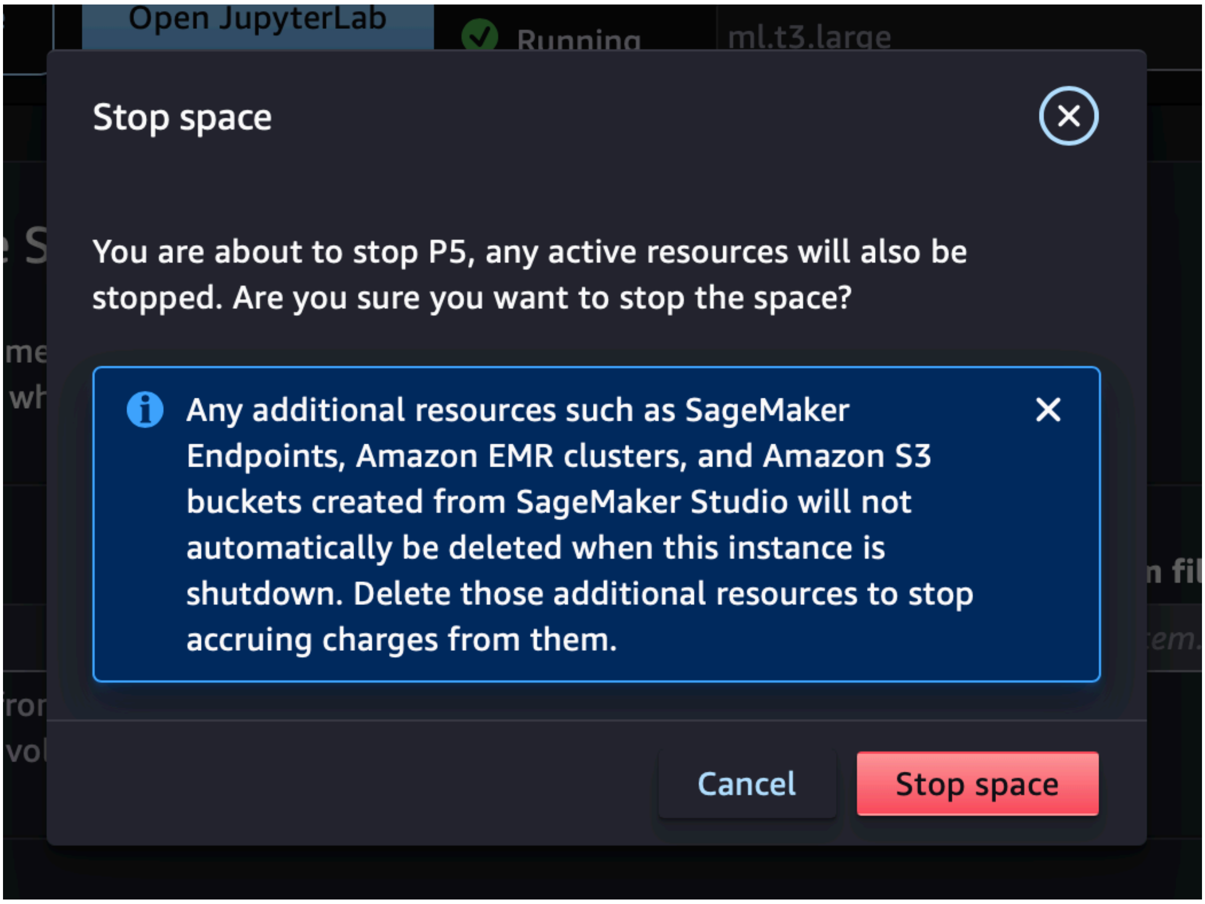


```
[1]: import os, sys, subprocess
     conda_prefix = os.environ.get("CONDA_PREFIX", sys.prefix)  # raíz del env actual
     os.environ["JAVA_HOME"] = conda_prefix
     os.environ["PATH"] = f"{conda_prefix}/bin:" + os.environ["PATH"]

     # Verifica que Java responde
     print("JAVA_HOME:", os.environ["JAVA_HOME"])
     print(subprocess.run(["java","-version"], capture_output=True, text=True).stderr)

     JAVA_HOME: /opt/conda
     openjdk version "11.0.27-internal" 2025-04-15
     OpenJDK Runtime Environment (build 11.0.27-internal+0-adhoc..src)
     OpenJDK 64-Bit Server VM (build 11.0.27-internal+0-adhoc..src, mixed mode)

[ ]: 
```

Práctica 5

3) Aunque se detenga el laboratorio, si no se detiene el espacio sigue consumiendo saldo. Debe pararse el espacio cada vez que se deje de trabajar en la práctica.

Nota: Cuando se para el espacio no se conserva la instalación de openJDK ni de PySpark, hay que repetir el paso 2 cada vez que se inicie el laboratorio



Stop space

You are about to stop P5, any active resources will also be stopped. Are you sure you want to stop the space?

Any additional resources such as SageMaker Endpoints, Amazon EMR clusters, and Amazon S3 buckets created from SageMaker Studio will not automatically be deleted when this instance is shutdown. Delete those additional resources to stop accruing charges from them.

Práctica 5