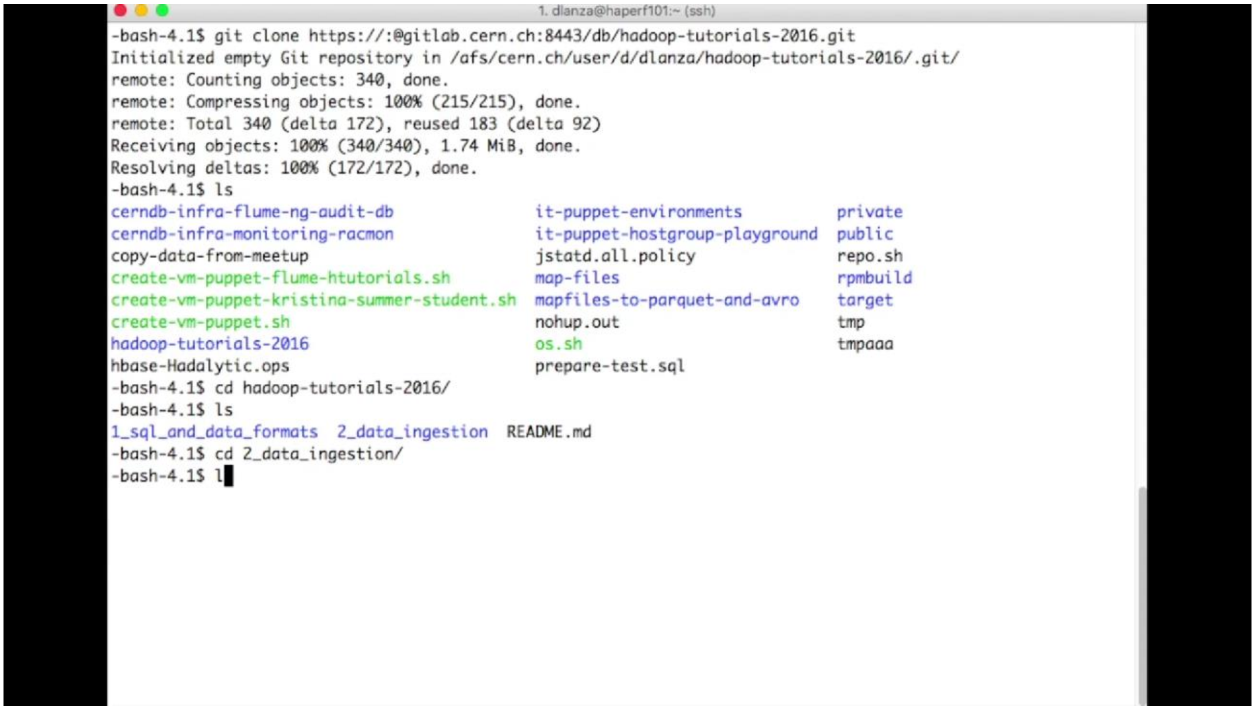
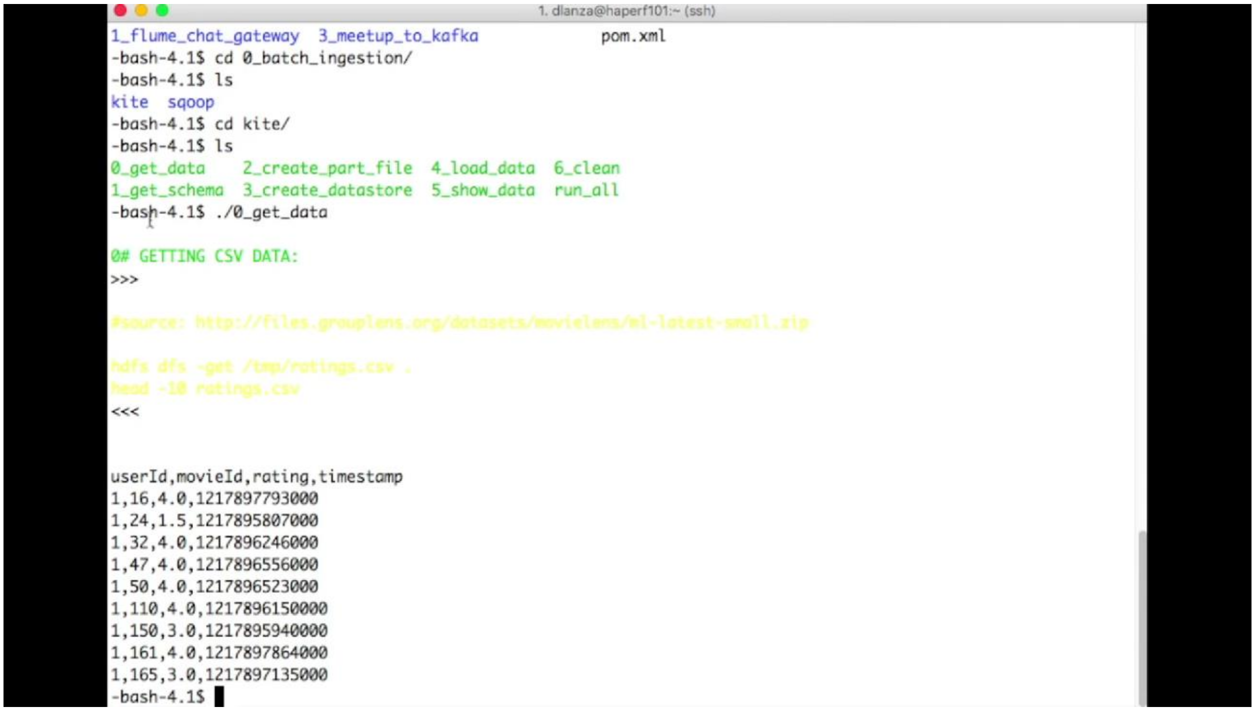
GROUP MEMBERS

WINNIE NJERI MICHINO	SCT221-0213/2021
FRANCIS IRAKI	SCT221-0404/2021
FAITH NAZI	SCT221-0489/2021
CHRIS MUNENE	SCT221-0098/2021
IAN MICHENI	SCT221-0446/2021

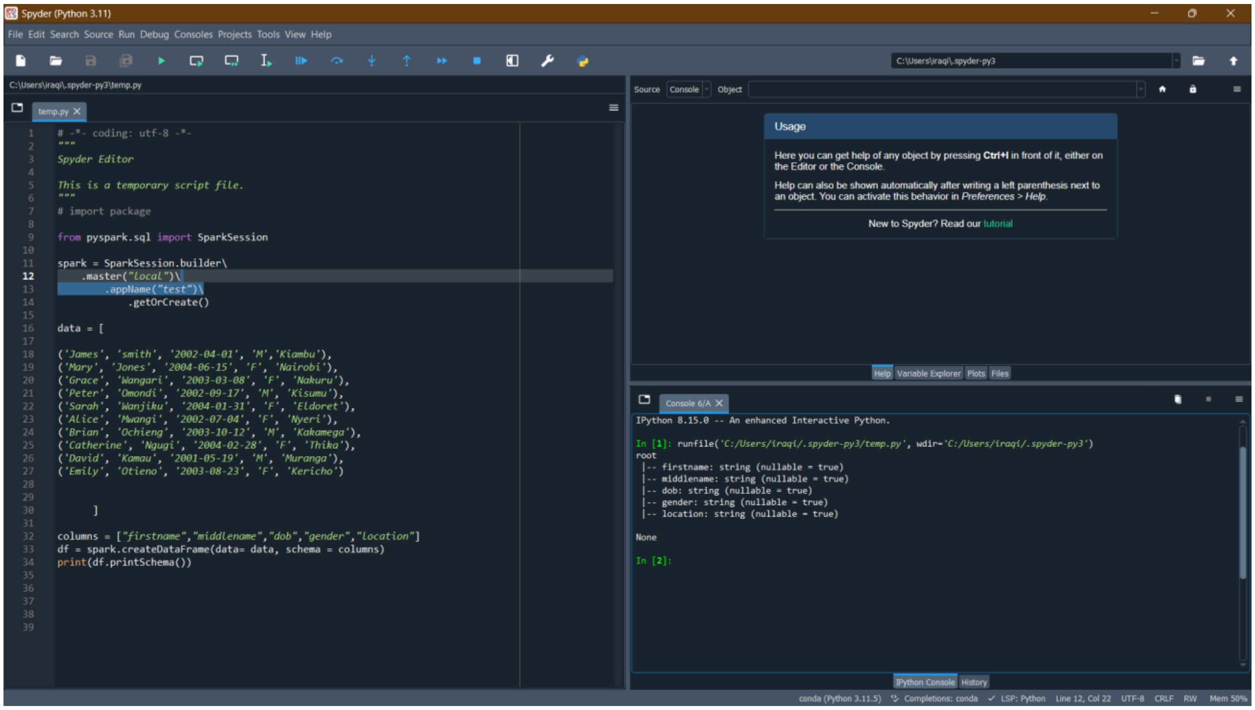
UNIT: ADVANCED PROGRAMMING.



```
-bash-4.1$ git clone https://:@gitlab.cern.ch:8443/db/hadoop-tutorials-2016.git
Initialized empty Git repository in /afs/cern.ch/user/d/dlanza/hadoop-tutorials-2016/.git/
remote: Counting objects: 340, done.
remote: Compressing objects: 100% (215/215), done.
remote: Total 340 (delta 172), reused 183 (delta 92)
Receiving objects: 100% (340/340), 1.74 MiB, done.
Resolving deltas: 100% (172/172), done.
-bash-4.1$ ls
cerndb-infra-flume-ng-audit-db               it-puppet-environments          private
cerndb-infra-monitoring-racmon               it-puppet-hostgroup-playground  public
copy-data-from-meetup                        jstatd.all.policy               repo.sh
create-vm-puppet-flume-htutorials.sh         map-files                       rpmbuild
create-vm-puppet-kristina-summer-student.sh  mapfiles-to-parquet-and-avro    target
create-vm-puppet.sh                          nohup.out                       tmp
hadoop-tutorials-2016                        os.sh                           tmpaaa
hbase-Hadalytic.ops                          prepare-test.sql
-bash-4.1$ cd hadoop-tutorials-2016/
-bash-4.1$ ls
1_sql_and_data_formats  2_data_ingestion  README.md
-bash-4.1$ cd 2_data_ingestion/
-bash-4.1$ l
```



```
1_flume_chat_gateway  3_meetup_to_kafka             pom.xml
-bash-4.1$ cd 0_batch_ingestion/
-bash-4.1$ ls
kite  sqoop
-bash-4.1$ cd kite/
-bash-4.1$ ls
0_get_data    2_create_part_file  4_load_data  6_clean
1_get_schema  3_create_datastore  5_show_data  run_all
-bash-4.1$ ./0_get_data

0# GETTING CSV DATA:
>>>

#source: http://files.grouplens.org/datasets/movielens/ml-latest-small.zip

hdfs dfs -get /tmp/ratings.csv .
head -10 ratings.csv
<<<


userId,movieId,rating,timestamp
1,16,4.0,1217897793000
1,24,1.5,1217895807000
1,32,4.0,1217896246000
1,47,4.0,1217896556000
1,50,4.0,1217896523000
1,110,4.0,1217896150000
1,150,3.0,1217895940000
1,161,4.0,1217897864000
1,165,3.0,1217897135000
-bash-4.1$
```



```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
# import package

from pyspark.sql import SparkSession

spark = SparkSession.builder\
    .master("local")\
        .appName("test")\
            .getOrCreate()

data = [

('James', 'smith', '2002-04-01', 'M','Kiambu'),
('Mary', 'Jones', '2004-06-15', 'F', 'Nairobi'),
('Grace', 'Wangari', '2003-03-08', 'F', 'Nakuru'),
('Peter', 'Omondi', '2002-09-17', 'M', 'Kisumu'),
('Sarah', 'Wanjiku', '2004-01-31', 'F', 'Eldoret'),
('Alice', 'Mwangi', '2002-07-04', 'F', 'Nyeri'),
('Brian', 'Ochieng', '2003-10-12', 'M', 'Kakamega'),
('Catherine', 'Ngugi', '2004-02-28', 'F', 'Thika'),
('David', 'Kamau', '2001-05-19', 'M', 'Muranga'),
('Emily', 'Otieno', '2003-08-23', 'F', 'Kericho')


      ]

columns = ["firstname","middlename","dob","gender","location"]
df = spark.createDataFrame(data= data, schema = columns)
print(df.printSchema())
```

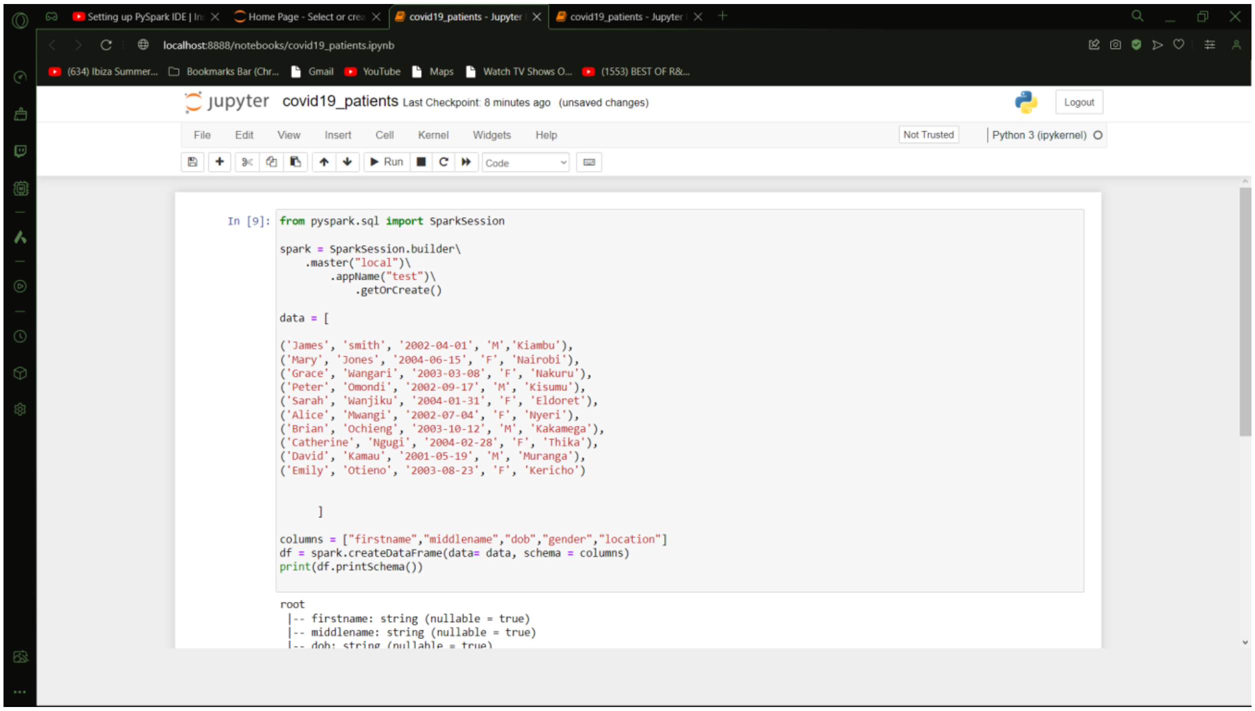
Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

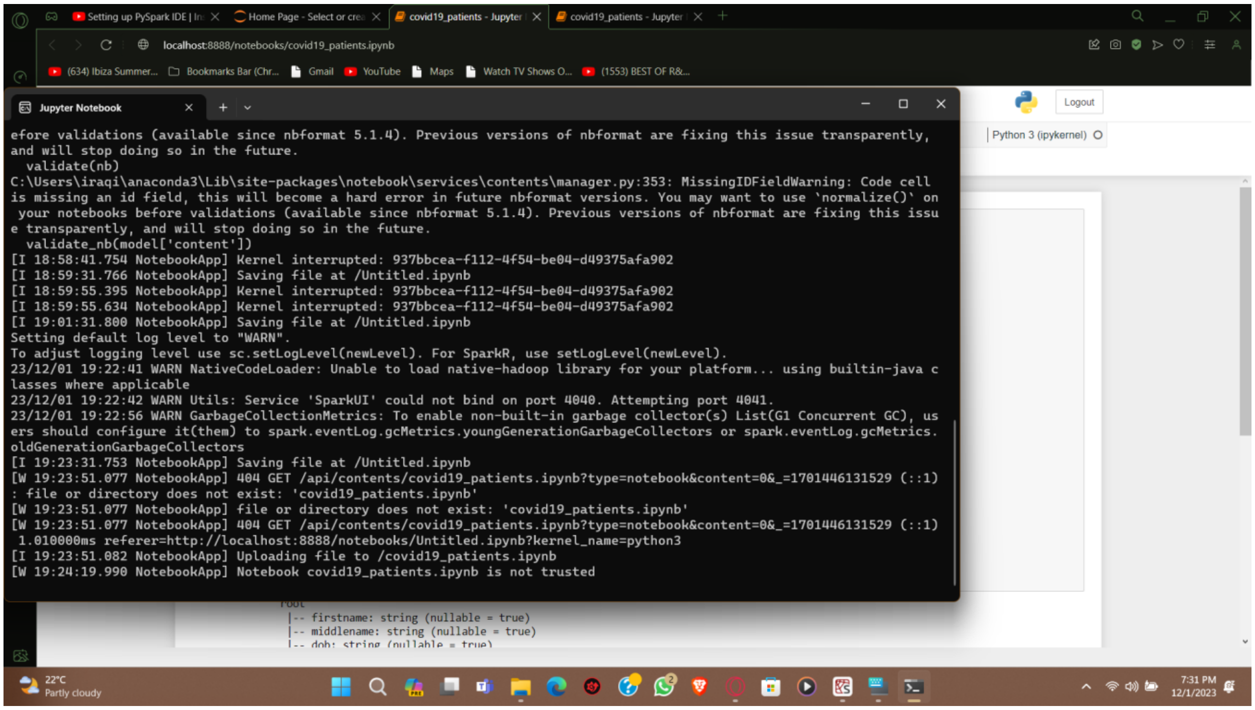
New to Spyder? Read our tutorial

```
IPython 8.15.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/iraqi/.spyder-py3/temp.py', wdir='C:/Users/iraqi/.spyder-py3')
root
 |-- firstname: string (nullable = true)
 |-- middlename: string (nullable = true)
 |-- dob: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- location: string (nullable = true)

None

In [2]:
```


Setting up PySpark IDE | In
Home Page - Select or crea
covid19_patients - Jupyter
covid19_patients - Jupyter
localhost:8888/notebooks/covid19_patients.ipynb
(634) Ibiza Summer...
Bookmarks Bar (Chr...
Gmail
YouTube
Maps
Watch TV Shows O...
(1553) BEST OF R&...
jupyter
covid19_patients Last Checkpoint: 8 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```python
In [9]: from pyspark.sql import SparkSession

        spark = SparkSession.builder\
            .master("local")\
                .appName("test")\
                    .getOrCreate()

        data = [

        ('James', 'smith', '2002-04-01', 'M','Kiambu'),
        ('Mary', 'Jones', '2004-06-15', 'F', 'Nairobi'),
        ('Grace', 'Wangari', '2003-03-08', 'F', 'Nakuru'),
        ('Peter', 'Omondi', '2002-09-17', 'M', 'Kisumu'),
        ('Sarah', 'Wanjiku', '2004-01-31', 'F', 'Eldoret'),
        ('Alice', 'Mwangi', '2002-07-04', 'F', 'Nyeri'),
        ('Brian', 'Ochieng', '2003-10-12', 'M', 'Kakamega'),
        ('Catherine', 'Ngugi', '2004-02-28', 'F', 'Thika'),
        ('David', 'Kamau', '2001-05-19', 'M', 'Muranga'),
        ('Emily', 'Otieno', '2003-08-23', 'F', 'Kericho')


              ]

        columns = ["firstname","middlename","dob","gender","location"]
        df = spark.createDataFrame(data= data, schema = columns)
        print(df.printSchema())
```

```
root
 |-- firstname: string (nullable = true)
 |-- middlename: string (nullable = true)
 |-- dob: string (nullable = true)
```



```
efore validations (available since nbformat 5.1.4). Previous versions of nbformat are fixing this issue transparently,
and will stop doing so in the future.
  validate(nb)
C:\Users\iraqi\anaconda3\Lib\site-packages\notebook\services\contents\manager.py:353: MissingIDFieldWarning: Code cell
is missing an id field, this will become a hard error in future nbformat versions. You may want to use `normalize()` on
 your notebooks before validations (available since nbformat 5.1.4). Previous versions of nbformat are fixing this issu
e transparently, and will stop doing so in the future.
  validate_nb(model['content'])
[I 18:58:41.754 NotebookApp] Kernel interrupted: 937bbcea-f112-4f54-be04-d49375afa902
[I 18:59:31.766 NotebookApp] Saving file at /Untitled.ipynb
[I 18:59:55.395 NotebookApp] Kernel interrupted: 937bbcea-f112-4f54-be04-d49375afa902
[I 18:59:55.634 NotebookApp] Kernel interrupted: 937bbcea-f112-4f54-be04-d49375afa902
[I 19:01:31.800 NotebookApp] Saving file at /Untitled.ipynb
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/12/01 19:22:41 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java c
lasses where applicable
23/12/01 19:22:42 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
23/12/01 19:22:56 WARN GarbageCollectionMetrics: To enable non-built-in garbage collector(s) List(G1 Concurrent GC), us
ers should configure it(them) to spark.eventLog.gcMetrics.youngGenerationGarbageCollectors or spark.eventLog.gcMetrics.
oldGenerationGarbageCollectors
[I 19:23:31.753 NotebookApp] Saving file at /Untitled.ipynb
[W 19:23:51.077 NotebookApp] 404 GET /api/contents/covid19_patients.ipynb?type=notebook&content=0&_=1701446131529 (::1)
: file or directory does not exist: 'covid19_patients.ipynb'
[W 19:23:51.077 NotebookApp] file or directory does not exist: 'covid19_patients.ipynb'
[W 19:23:51.077 NotebookApp] 404 GET /api/contents/covid19_patients.ipynb?type=notebook&content=0&_=1701446131529 (::1)
 1.010000ms referer=http://localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
[I 19:23:51.082 NotebookApp] Uploading file to /covid19_patients.ipynb
[W 19:24:19.990 NotebookApp] Notebook covid19_patients.ipynb is not trusted
```

```
 |-- firstname: string (nullable = true)
 |-- middlename: string (nullable = true)
 |-- dob: string (nullable = true)
```

(iv) Describe pre-processing tasks/techniques used to prepare the data:

Handle Missing Data:

Use PySpark functions like na.drop() or na.fill() to handle missing values.

Data Cleaning and Transformation:

Remove irrelevant columns or rows using PySpark DataFrame operations.

Use functions like filter() or drop().

Feature Engineering:

Create new features or modify existing ones based on domain knowledge.

Use PySpark DataFrame transformations.

Scaling/Normalization:

If your algorithm requires it, use PySpark's StandardScaler or MinMaxScaler for feature scaling.

Reasoning:

The choice of pre-processing tasks depends on the characteristics of your data and the requirements of your predictive model.

Handling missing data is crucial to avoid biases in your analysis.

Data cleaning and transformation ensure that the data is in a suitable format for analysis.

Feature engineering enhances the model's ability to capture patterns.

Scaling is essential for algorithms sensitive to the scale of features.