Spark-Dask Project 2025

Due to March 1st 2025

I/ HADOOP, SPARK AND DASK

```
> cd TP2/Docker
```

```
> git clone https://github.com/cluster-apps-on-docker/spark-standalone-cluster-on-docker.git
```

```
> curl -LO https://raw.githubusercontent.com/cluster-apps-on-docker/spark-standalone-cluster-on-docker/master/docker-co
```

```
> docker-compose up
```

Connect to JupiterLab NoteBook at http://localhost:8888

In the notebook environnement, do not forget to install dask (pip install dask) for the dask part of the project

II/ Project Description :

In the BigDataHadoopSpakDaskCourse git project, find the following files :

-Tps---|-data----------|iris.csv

|-TP5----------|iris_ml.py |

|tools----------|launch_cmd_local.sh

A/ Spark ML : Iris classification

Data : iris.csv

In the python iris_ml.py script, a DecisionTreeClassifier is used to predict the flower species.

1/ Write a Spark Pipeline to transform and process the data before applying the machine learning process.

2/ Write two other versions of this test using :

- The Random Forest Classifier ;

- The Gradient Boosted Tree Classifier (transform the binary Classifier in a multi-class classifier using a tree classifier : for (A,B,C,D,…) classes, use (A, not A), (B, not B), (C, not C), (D, not D),... binalry classifiers.

Compare the performance of the three machine learning models.

B/ Dask ML : Iris classification

Data : iris.csv

In the python iris_ml.py script, a DecisionTreeClassifier is used to predict the flower species.

1/ Write a Dask Pipeline to transform and process the data before applying the machine learning process.

2/ Write two other versions of this test using :

- The Random Forest Classifier ;
- The Gradient Boosted Tree Classifier.

Compare the performance of the three machine learning models.

C/ Spark : parallelisation of the image processing algorithm « MedianFilter »

The median filter consists in replace the value of a pixel p[i,j] by the median value of the list :

[p[i-1,j-1],p[i-1,j],p[i-1,j+1],p[i,j-1],p[i,j],p[i,j+1],p[i+1,j-1],p[i+1,j],p[i+1,j+1]]

Complete the file « median_filter.py » to write a spark parallel median_filter.py

Execute the script on the « lena_noizy.jpg » image and generate the new file « lena_filter.jpg »

Send me the script completed and te new generated file « lena_filter.jpg »

Attention :

Take care of overlap issues : make sure that regarding how the data is split that everything is ok on each partition bondaries.

D/ Dask : parallelisation of the image processing algorithm « MedianFilter »

The median filter consists in replace the value of a pixel p[i,j] by the median value of the list :

[p[i-1,j-1],p[i-1,j],p[i-1,j+1],p[i,j-1],p[i,j],p[i,j+1],p[i+1,j-1],p[i+1,j],p[i+1,j+1]]

Complete the file « median_filter.py » to write a dask parallel median_filter.py

Execute the script on the « lena_noizy.jpg » image and generate the new file « lena_filter.jpg »

Send me the script completed and te new generated file « lena_filter.jpg »

Attention :

Take care of overlap issues : make sure that regarding how the data is split that everything is ok on each partition bondaries.