Big Data Hackathon 2017 Program

The goal is to take you through all the steps of a small big data project in a single day and give you some hands-on experience on a cutting-edge big data platform. The steps below are merely some guidelines that you can follow – you can depart from them at any point and originality in what you do may bring rewards! **Above all, try to explore and have fun**!

Step 1: Understand the business case

We can use satellite data (remote sensing) to make decisions that confront climate change. Can satellite data help people decide what regions will continue to be safe to live in and which should be avoided? Where it's best to grow crops? To answer questions like these, we can use terrabytes of landsat satellite data sets that are publicly available on aws.

Step 2: Get your data

Mount your amazon s3-bucket: https://aws.amazon.com/public-datasets/landsat/ . The information on the cloudcover can be found in the csv file at the bottom of the page.

Step 3: Explore your data

You will want to look at your images now, but they might be cloudy. You can use one of Spark's APIs to explore the scenes csv file and gain insight into the cloud cover of the > 1 million pics available. The location of the landsat data is expressed in paths/rows, how does this relate to longitude/latitude? Here are some suggested steps for exploration of the available data:

- Download scenes file, unzip it and rename to csv, make it available on hdfs and load this into a spark dataframe
- Cloudcover is the probability that a pixel is covered by the clouds. What values do you expect? Are there any values outside of this range? Does the data contain duplicates? Clean your data if necessary.
- Would it not be easier if we could just run a query on our data to get to know it better? Try to turn it into a query-friendly format!
- The landsat 8 project started in 2013. How far back in time does the data go? Does it look like all the files are here?
- It is hard to find good satellite pictures of Belgium...always so cloudy! Find the least cloudy day in Belgium in the last 3 years.
- Time for a vacation! Select the average cloudcover in Belgium and compare it to that of the Costa Blanca. You can use for instance Benidorm's coordinates.
- July in Belgium is the sunniest month, averaging 7 hours of sunshine per day... Can you figure out the average cloudcover in the month of July using all of the available data?
- Can you figure out the cloudcover averages for all months...in just 1 command line? To test your results, try to answer the following question: in Southern California, two months (outside of the wintermonths December through February) are known to have a weather pattern that results in cloudy, overcast skies. This gives rise to appropriate nicknames for

those months. Can you show the monthly average cloud covers, and find out what those two months are (nickname included!) using the dataset available for the coastal region around the city of San Diego?

Step 4: Store your findings

The EMR cluster will be terminated from time to time to avoid unnecessary charges when idle. It seems a good idea to store our newly found results. This can be done using AWS' noSQL DynamoDB. You can start by manually creating a table for cloudcover and add some records. Then see if you can automate this and write the whole dataframe containing the monthly averages to the DB using the batch_write() function. Check out the tutorials http://boto3.readthedocs.io/en/latest/guide/dynamodb.html for inspiration!

Step 5: Build Visualizations

Now you can admire the place of your dreams on the planet. Use the info from the previous step to show the place of your dreams on a day with clear skies. Make use of the Python script to plot Landsat images described in : https://nex.nasa.gov/nex/projects/1217/wiki/access_and_visualize_landsat_data_in_the_geotiff_format_on_aws/ Task 3- Step 3. You can fine-tune this to show the spectral bands of your choice.

Step 6: Get predictive

Water is key in growing crops. Can you find a way to determine the water content of an image? You could start with one image and try to calculate the probability a pixel will be water. Can you automate this and take into account more images? Based on the data available, make a prediction of where you would migrate. What other kinds of questions might this data help us answer?

Step 7: Present your results

Showcase your findings of the previous day – try to be original. Even if your code is not working 100% properly, try to come up with a good plan of attack and show off your presentation skills!

Do you like what you see?

Check out the following things if you want more information:

https://aws.amazon.com/	Secure cloud computing environment, suitable for big data projects.
https://spark.apache.org/	Cluster-computing framework for big data processing. Several online courses are available on EdX and Coursera.
www.bigindustries.be/	The coolest big data consultancy company on the block.

Appendix: Set-up information

1. Required software installation when working under windows: Xming (X server to enable graphics) and putty to connect to the EMR master node. Start Xming.
2. We will need to add your IP address to allow inbound traffic to the EMR master node. Please check your IP address e.g. through http://checkip.amazonaws.com/ .
3. Log on to the EMR's master node using putty. Check under connection -> SSH -> X11 the "enable X11 forwarding" box. Under connection ->ssh ->Auth , browse on your pc to select the private key for authentication, using the ppk key that is provided (htf2017.ppk). Your host name will look like hadoop@ec2-34-242-12-133.eu-west-1.compute.amazonaws.com although the exact address will differ from the address above. Port is 22 and connection type ssh. To make use of Zeppelin notebook, in ssh choose "connection ->ssh->tunnel and fill in source port = 8890; destination =master dns followed by ":8890" (something like ec2-52-51-187-82.eu-west-1.compute.amazonaws.com:8890)".
4. Once logged in, you will find a directory /home/Hadoop/<teamname>. Store all files your files in this directory.
5. Then go to http://localhost:8890/ in a browser and Zeppelin should appear. Change the settings to make use of the correct python version : (right upper corner) interpreter -> spark -> properties -> zeppelin.pyspark.python = /home/hadoop/miniconda2/bin/python and save. Now start a new note (notebook +) and let every command proceed by '%spark.pyspark'.