Project phase 1
Data Analysis with Spark

Due date:

12 March 2018 23:59

The project of the course TDT4305 consists of two phases. This document describes the first one which focuses on learning how to perform data analysis on a large dataset. You will work with the Apache Spark framework[1] and you can choose freely which of the Spark-compatible languages you prefer: Python, Scala or Java. You can work in groups of max. two people.

1 Exploratory Analysis of Twitter Dataset

Twitter[2] is a social media platform where users can post and interact with messages, known as "tweets". In this task you will work with a preprocessed dataset downloaded from the publicly available Twitter Api. The aim of this phase is to learn the differences and the best practices in usage of various Spark functions.

1.1 Data

A link to the dataset is available on Blackboard, under the "Prosjekt\Project" tab. It is a zip file that you need to unzip it to use it, and it contains the following:

geotweets.tsv

It is represented in TSV format and contains the following columns:

1. **utc_time** – check-in time on the server represented by a UNIX timestamp (in msec).
2. **country_name**
3. **country_code**
4. **place_type**
5. **place_name**

[1] http://spark.apache.org/
[2] https://twitter.com/

6. **language**
7. **username**
8. **user_screen_name** – the name of the user displayed on Twitter account.
9. **timezome_offset** – in seconds
10. **number_of_friends**
11. **tweet_text**
12. **latitude**
13. **longitude**

stop_words.txt

It this file the stop-words that will be used are listed one per line.

1.2 Notes before you start

- Try to find appropriate Spark functions with respect to the tasks you are trying to accomplish.
- Use only data/columns needed for each task to make your solution efficient.
- During experiments on your local machine, use a sample of the tweets dataset to speed-up the processing time. To sample 10% of the whole dataset without replacement and with a seed number 5 (every time you get the same sample), in Python call:
  rdd_sample = rdd.sample(false, 0.1, 5)
- To output results into a file, use saveAsTextFile function, and format your output as a TSV (tab-separated) file.
- To avoid multiple files when exporting results into a file, decrease number of data partitions using coalesce(1) or repartition(1) functions. In your report, write down if you used any, which one and why.
- Words in the tweet texts are separated by the space character ' '.
- All stop words are in lowercase, so you must convert to lowercase the tweet texts and use case-insensitive comparison.

1.3 RDD API Tasks

On your local machine install Spark, download above mentioned datasets, load the dataset into an RDD, and using suitable Spark functions (such as map, reduce, reduceByKey, sortByKey, etc.) perform these tasks:

1. Explore and briefly describe the dataset.
   (a) How many tweets are there?
   (b) How many distinct users (username) are there?
   (c) How many distinct countries (country_name)

(d) How many distinct places (place_name) are there?

(e) In how many languages users post tweets?

(f) What is the minimum latitude?

(g) What is the minimum longitude?

(h) What is the maximum latitude?

(i) Waht is the maximum longitude?

(j) What is the average length of a tweet text in terms of characters?

(k) What is the average length of a tweet text in terms of words?

Write a code (named "task_1") that writes these results into a TSV file (one per line) with name "result_1.tsv" and include the results in your report.

2. Find the total number of tweets posted from each country and sort them in descending order of tweet counts. For countries with equal number of tweets, sorting must be in alphabetical order. Write a code (named "task_2") that writes the results in a TSV file in the form of <country_name>tab<tweet_count> and name it "result_2.tsv".

3. For each country that has more than 10 tweets, find its geographical centroid.

   (a) Write a code (named "task_3") that outputs in a TSV file the latitude and longitude of the centroids and the names of the countries, in the form of <country_name>tab<latitude>tab<longitude>

   (b) Visualize the results in CartoDB

   (more information below)

4. Calculate local time for each tweet (UTC time + timezone offset) and find the 1-hour interval with maximum number of tweets for each country in the form of <country_name>tab<begining_hour>tab<tweet_count> [3]. Times should be rounded down to the hour, in 24 hour scale, so that a tweet posted between [13:00,14:00) should be interpreted as posted at 13. Write a code (named "task_4") that writes the results in a TSV file and name it "result_4.tsv".

5. Find the number of tweets from each city in US (place_type = 'city' and country_code = 'US') in the form of <place_name>tab<tweet_count>. Write a code (named "task_5" that writes the results in a TSV file named "result_5.tsv" in descending order of tweet counts. For cities with equal number of tweets, sorting must be in alphabetical order.

6. Find the 10 most frequent words (in lowercase) and their frequencies from the US, excluding the words shorter than 2 characters (length < 2) and the words from the stop words file. Write a code (named "task_6") that writes the results in a TSV file named "result_6.tsv" in the form of <word>tab<frequency>.

7. Find the 5 cities in the US with the highest number of tweets (place_type = 'city' and country_code = 'US', ordered by their tweet counts/alphabetical). For these 5 cities, find the 10 most frequent words ordered by their frequency, ignoring the stop words from the file and excluding words shorted than 2 characters (length < 2). Write a code (named "task_7") that writes the results in a TSV file named "result_7.tsv" in the form of <place_name>tab<word1>tab<frequency1>tab<word2>tab<frequency2> ... <word10>tab<frequency10>.

[3] Python code to have a more readable time format: datetime.fromtimestamp(time_in_sec)

1.4 Dataset API Tasks

Working on Spark, load the dataset into a Dataframe, name it, rename the columns with the names of columns described before in section 1.1, and using suitable Dataframe functions perform these tasks:

8. Calculate the following using Spark SQL on your Dataframe:
   (a) Number of tweets.
   (b) Number of distinct users (username)
   (c) Number of distinct countries (country name)
   (d) Number of distinct places (place name)
   (e) Number of distinct languages users post tweets
   (f) Minimum values of latitude and longitude
   (g) Maximum values of latitude and longitude

Write a code (named "task_8") that calculates the results and prints them with the show() function in the console. Don't forget to include them in your report also.

1.5 Geographical Centroid

The Geographical Ceontroid of a list of geographical points is nothing more than the numerical average of all the latitudes and all longitudes in the list. To calculate the geographical centroid of a country from its tweets for this phase, you can use the following formula:

$$Centroid\_of\_list(lat, lon) = (\frac{\sum_{i=1}^{size(list)} lan_i}{size(list)}, \frac{\sum_{i=1}^{size(list)} lon_i}{size(list)}) \quad (1)$$

where, list is all the places that the centroid needs to be calculated, lan is their latitude, and lon is their longitude.

1.6 Visualization in Carto.com

Carto is an in-browser mapping service, which you can use to place your data on a map. Your goal is to visualize the countries with the most tweets by exporting latitude, longitude, and country name into a TSV file and uploading it to Carto.
How to visualize the check-ins:

1. Create a free account at https://carto.com/signup/
2. Upload your results in TSV format with one country per line.
3. Connect the dataset and 'DATA VIEW' tab should open.
4. Select which of your columns refer to coordinates (lat, lon). If you name your columns 'lat'/'lon', Carto will recognize them automatically.
5. Check if column with data of your interest was recognized as a date data type, otherwise change the data type.
6. Switch to 'MAP VIEW' tab.
7. In the right panel, choose 'wizard' and adjust the map type, style, labels, etc. to your needs.
8. Put your map in your report.

1.7 Delivery

You should deliver to Blackboard a ZIP file containing:

- a short report (PDF)
- source codes of your script/program with the described names (.py/.scala/ .java[4])
- output files (.tsv) with your results as described

In your report, provide answers to all the (sub)tasks and briefly describe which Spark functions you used in your solution and why. As mentioned before, you can work in groups of max. two people, providing both names in your report and both should deliver the project on Blackboard.

There will be slots for presentation (and questioning) about the project on April 12-13 and April 23-24.

[4]For java coding, include also a *RUNNABLE* jar with the compile and the spark submit parameters