Semester Project

Description

This semester project requires analysis of (large) data sets, applying data science processing techniques The tools that will be used in the project are Apache Hadoop (version>=3.0) and Apache Spark (version>=3.5).

You are invited to use the resources in the specially configured environment that has been granted to you in the AWS cloud. In summary, the purpose of the project is:

- The familiarization and development of students' skills in the installation and management of the distributed Apache Spark and Apache Hadoop systems.
- The use of modern techniques through the Spark APIs for the analysis of volume data.
- The understanding of the capabilities and limitations of these tools in relation to the available resources and the settings that have been selected.

Datasets

This section will present the data that you will be asked to use in the context of the six-month project. These are publicly available and free datasets that have been collected from different sources. For your convenience, all necessary datasets are accessible in the following S3 bucket of the AWS cloud: s3://initial-notebook-data-bucket-dblab-905418150721/

Data set	**S3 URI**
Los Angeles Crime Data (2010-2019)	s3://initial-notebook-data-bucket-dblab-905418150721/CrimeData/Crime_Data_from_2010_to_2019_20241101.csv
Los Angeles Crime Data (2020-)	s3://initial-notebook-data-bucket-dblab-905418150721/CrimeData/Crime_Data_from_2020_to_Present_20241101.csv
LA Police Stations	s3://initial-notebook-data-bucket-dblab-905418150721/LA_Police_Stations.csv
Median Household Income by Zip Code	s3://initial-notebook-data-bucket-dblab-905418150721/LA_income_2015.csv
2010 Census Blocks	s3://initial-notebook-data-bucket-dblab-905418150721/2010_Census_Blocks.geojson
Race and Ethnicity Codes	s3://initial-notebook-data-bucket-dblab-905418150721/RE_codes.csv

The main dataset used in this work comes from the public data repository of the United States government. Specifically, it includes crime data for Los Angeles from 2010 to the present. The data is available in csv file format at the following links:

• https://data.lacity.org/Public-Safety/Crime-Data-from-2010-to-2019/63jg-8b9z

• https://data.lacity.org/Public-Safety/Crime-Data-from-2020-to-Present/2nrs-mtv8

The same links provide descriptions for each of the 28 fields in the data.

In addition to the above data, smaller datasets will be used (also available in public repositories or sources):

2010 Census Blocks (Los Angeles County): A dataset that presents census data for Los Angeles County for the year 2010 in geojson format. It is accompanied by a file with descriptions of its fields (2010_Census_Blocks_fields.csv). It is available at the following link:

• https://data.lacounty.gov/maps/lacounty::2010-census-blocks

Median Household Income by Zip Code (Los Angeles County): Another small dataset that contains data on the median income per household and ZIP Code in Los Angeles County. For convenience, the data has been collected and stored in csv file format. This dataset was produced based on the results of the 2015 census and is available at the following link:

• http://www.laalmanac.com/employment/em12c_2015.php

LA Police Stations: A small dataset containing data on the location of the 21 police stations located in the city of Los Angeles. The data is from a public data repository of the City of Los Angeles and is available in csv file format at the following link:

• https://geohub.lacity.org/datasets/lahub::lapd-police-stations/explore

Race and Ethnicity codes: A small dataset containing the full descriptions that correspond to the racial profile coding used in the main dataset.

Queries

Query 1

Sort, in descending order, the age groups of victims in incidents involving any form of “aggravated assault” (i.e., include this term in the relevant description). Consider the following age groups:

Children: < 18, Young adults: 18 – 24, Adults: 25 – 64, Elderly: >64

Query 2

Find, for each year, the 3 Police Departments with the highest percentage of closed cases. Print the year, the names (locations) of the departments, their percentages as well as their ranking. The results are given in ascending order by year and ranking (see example below).

year	precinct	closed_case_rate	#
2010	West Valley	30.57974335472044	1
2010	N Hollywood	29.23808669119627	2
2010	Mission	27.58372669119627	3

Query 3

Using the 2010 Census population data and the 2015 Census household income data, calculate the following for each area of Los Angeles: The average annual income per person and the ratio of total crimes per person. The results should be summarized in a table.

Query 4

Find the racial profile of registered crime victims (`Vict Descent`) in Los Angeles for the year 2015 in the 3 areas with the highest per capita income. Do the same for the 3 areas with the lowest income. Use the mapping of the descent codes to the full description from the Race and Ethnicity codes dataset. The results should be printed in two separate tables from highest to lowest number of victims per racial group (see example result below).

Victim Descent	#
White	413
Black	274
Unknown	132
Hispanic/Latin/Mexican	12

Query 5

Calculate, for each police station, the number of crimes that took place closest to it, as well as its average distance from the locations where the specific incidents occurred. The results should be displayed sorted by number of incidents, in descending order (see example below).

division	average_distance	#
77TH STREET	2.208	7045
RAMPART	2.009	4595
FOOTHILL	3.597	3047
PACIFIC	2.739	2132

Tips:

1. To implement queries that include geospatial analytics, you should use the Apache Sedona library (version 1.6.1), which has been installed in your work environment. You are provided with a user guide in a related notebook that you can find in the corresponding section of your account. More information can be found in the documentation and the website: https://sedona.apache.org/1.6.1/.
2. Assume that the areas of Los Angeles are defined by the COMM column of the **2010 Census Blocks**.
3. Some records in the basic dataset incorrectly refer to Null Island (0,0). They should be filtered out and not taken into account in the calculation of distances.

Questions

1. Implement Query 1 using the DataFrame and RDD APIs. Run both implementations with 4 Spark executors. Is there a performance difference between the two APIs? Justify your answer. (20%)
2. a) Implement Query 2 using the DataFrame and SQL APIs. Report and compare the execution times between the two implementations. (10%)

   b) Write Spark code that converts the main data set to parquet file format and stores a single .parquet file in your team's S3 bucket. Choose one of the two implementations of subquery a) (DataFrame or SQL) and compare the execution times of your application when the data is imported as .csv and as .parquet. (10%)
3. Implement Query 3 using DataFrame or SQL API. Use hint & explain methods to find out which join strategies the catalyst optimizer uses. Experiment by forcing Spark to use different strategies (between BROADCAST, MERGE, SHUFFLE_HASH, SHUFFLE_REPLICATE_NL) and comment on the results you observe. Which of the available Spark join strategies is (are) the most appropriate and why? (20%)
4. Implement Query 4 using the DataFrame or SQL API. Execute your implementation by scaling the total computational resources you will use: Specifically, you are asked to execute your implementation with 2 executors with the following configurations:

   • 1 core/2 GB memory

   • 2 cores/4GB memory

   • 4 cores/8GB memory

   Comment on the results. (20%)
5. Implement Query 5 using the DataFrame or SQL API. Execute your implementation using a total of 8 cores and 16GB of memory with the following configurations:

   • 2 executors × 4 cores/8GB memory

   • 4 executors × 2 cores/4GB memory

   • 8 executors × 1 core/2 GB memory

   Comment on the results. (20%)

Deliverables – Submission Terms

- The assignment must be completed in groups of (maximum) 2.
- The submission deadline will be set in https://helios.ntua.gr/course/view.php?id=887 in a link that will open soon.
- The project constitutes 30% of the total course grade. In order to be graded, each team must submit a report and successfully pass the mandatory oral examination on the subject of the

assignment. The examination will take place after the assignment is submitted (a relevant schedule will also be posted on helios site).

- The delivered report must be a single .pdf file named after the IDs of the group members separated by an underscore (or the ID of the student in the case of a single-member group), e.g., 03100000.zip, or 03100000_03100001.zip (depending on the number of people in the group). The file will contain a report (strictly according to what is requested here) which will include the answers to the questions as well as a link to a repository (github, gitlab, bitbucket, etc.) with the codes you have implemented, as well as possible scripts/howtos for executing your code. All submissions are strictly subject to the academic ethics code of NTUA and the School of ECE. **Your code must not be changed from the day the report is submitted until the grading of the course. If this happens, your grade will be ZERO (0).**
- Each group can implement its code in Scala, Java or Python. In addition, you are given the opportunity to use your own resources (e.g., personal PC, VM on another cloud provider), as long as the requirements of the assignment are met. In any case, the examination will require a live demonstration of your code.
- Questions/explanations about the assignment will be given via the forum on the course page at helios. Do not send questions to the teachers/assistants' email accounts.