New York Parking Tickets

Big Data Project 2023/24

General instructions

Your task is to perform all steps of comprehensive data analysis on a moderately sized dataset (in terms of big data). You should adhere to the CRISP-DM methodology [1] and structure your report accordingly. You must work in teams of **2 students**.

Tools

Select the appropriate tools from the Python data science ecosystem, e. g. Dask, Pandas, Numpy, Matplotlib, Faust, Streamz, Parquet, DuckDB, Kafka, … Where required, your experiments must be performed and evaluated on Arnes HPC cluster.

Data

The initial dataset is the New York City open data - Parking Violations [2] for years 2014 – 2024 that you will augment with additional information from the data sources of your choice (as per instructions below). Your analysis should in all cases focus on

a. Full data (2014-2024)
b. Boroughs (see for example [6])
c. An "interesting" subset of streets (e.g., most problematic streets)

You can download the CSV-formatted data from the source [2], or from the Arnes cluster (directory `/d/hpc/home/mkukar/bigdata/data/NYTickets`).

Tasks

Your project consists of the following tasks.

T1. Import CSV datasets and store them in (a) Parquet format and (b) HDF5 format. Compare the datasets in terms of file sizes. Choose appropriate partitioning where applicable.

T2. Augment the original Parking violations data with sources of additional information:
   a. Weather information
   b. Vicinity/locations of primary and high schools
   c. Information about events in vicinity
   d. Vicinity/locations of major businesses
   e. Vicinity/locations of major attractions

   You can find many (but not all) sources in the New York City open data repository. You will need to link the data with respect to location (location data or street names) and time (where applicable, e. g. for weather and event data).

T3. Programmatically perform the introductory exploratory data analysis. Select and calculate appropriate data aggregates. Determine and visualize how good your data augmentation is. When using DuckDB on Parquet files, try to balance between SQL and Dask processing.

T4. Perform the data analysis in a "streaming" manner (treat data as streaming). Use Kafka as a message broker and write a custom producer (reading data from files) and a custom consumer (for processing data). Since data is (with few exceptions, most likely errors) ordered by the "summons number" you can assume that the lines in CSV files are in chronological order (important for producing Kafka messages). Decide if using different topics can help you. Show rolling descriptive statistics (mean, standard deviation, …, think of at least three more) for all data, boroughs, and for the 10 most interesting streets (with highest numbers of tickets overall, or by your qualified choice). For the same data, choose, implement, and apply a stream clustering algorithm (preferably spatial clustering) of your choice [7].
Note: stream processing is better not performed in the Arnes cluster.

T5. Perform the data analysis in a "batch" manner using machine learning to predict events such as days with high number of tickets (think of and implement at least one additional interesting learning problem). You will need to appropriately transform the augmented data. Ensure that each single worker **will not have** enough memory to store and process the entire dataset (e.g., 8GB per worker). Use at least three kinds of supervised machine learning algorithms:

a. One of the existing distributed algorithms from Dask-ML
b. A sophisticated third-party algorithm which "natively" supports distributed computing (such as XGBoost or LightGBM)
c. One of the common scikit-learn algorithms utilizing partial_fit.

For all three scenarios compare performance in terms of loss (error), scalability, time, and total memory consumption.
Note: scalability must be tested in the Arnes cluster by increasing the number of workers and observing the total processing time.

T6. Repeat the tasks T2, T3 and T5 with different data formats and scenarios: Dask on Parquet files, then Dask+DuckDB on Parquet files, and finally Dask on HDF5 files.). Evaluate their qualitative (subjective) and quantitative (scalability, measured time, data size) advantages and disadvantages.

T7. (optional) Visualize your results from T3 on a map (Google Maps, Open Street Maps, or a map of your choice); see for example prettymaps [4, 5].

Reporting

Prepare a PDF report of at least 10 pages and as well as all code files (packed in a ZIP file). Include as many interesting and relevant visualizations as necessary. Find and include appropriate precise references.

Each team will present their preliminary results in a short 5-minute presentation at the end of the semester.

References

[1] CRISP-DM, http://lyle.smu.edu/~mhd/8331f03/crisp.pdf

[2] New York City open data - Parking Violations Issued: https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2022/pvqr-7yc4; for previous years see https://catalog.data.gov/dataset/parking-violations-issued-fiscal-year-2023 (you can change the year from 2014 to 2023)

[3] Data folder on Arnes HPC cluster: `/d/hpc/projects/FRI/bigdata/data/NYTickets`

[4] Marcelo de Oliveira Rosa Prates, Prettymaps, https://github.com/marceloprates/prettymaps

[5] Prettymaps, New York example, https://www.reddit.com/r/prettymaps_/comments/qpl31e/code_for_local_use_new_york_example/

[6] Red Zone, Blue Zone: Discovering Parking Ticket Trends in New York City, https://newyorkparkingticket.com/wp-content/uploads/2016/11/NYC-Parking-Ticket-Report_parking_Samuel_Ackerman5.pdf

[7] Data stream clustering, https://en.wikipedia.org/wiki/Data_stream_clustering