Assignment 5

Course: *Big Data*
Due date: *May 19th, 2024*

Assignment

In this assignment you will learn about Apache Kafka and stream processing with Faust.

Setup your Kafka instance in Docker.

- Take the docker-compose.yml file from Učilnica and run it. Check if you have two containers running (kafka and zookeeper).
- You can create topics in Kafka with:

  ```
  docker exec kafka /usr/bin/kafka-topics --create --topic Tematika --partitions 1 --replication-factor 1 --bootstrap-server localhost:9092
  ```

  but the configuration of Kafka in this container lets you create topics inside Python.
- Login to kafka:

  ```
  docker exec -it kafka bash
  ```

- Install the $confluent - kafka$, (or the $kafka - pythion$ library) and $faust$ library inside the container. To use Jupyter notebook inside Docker refer to `https://www.docker.com/blog/supercharging-ai-ml-development-with-jupyterlab-and-docker/`

Download subhourly data for three different stations from:`https://www.ncei.noaa.gov/pub/data/uscrn/products/subhourly01/2021/`.

The structure of the data is described in `https://www.ncei.noaa.gov/pub/data/uscrn/products/subhourly01/`.

Write a Kafka Producer and consumer.

You can use the $confluent - kafka$ or the $kafka - pythion$ library. You can refer to `https://docs.confluent.io/kafka-clients/python/current/overview.html`.

Check out the *Faust* library.

- Start with `https://abhishekbose550.medium.com/basic-stream-processing-using-kafka-and-faust-7de07ed0ea77`

- Watch the Introduction to Kafka Stream Processing in Python using Faust `https://www.youtube.com/watch?v=Nt96udaC5Zk`.
- Read `https://towardsdatascience.com/stream-processing-with-python-kafka-faust-a11740d0910c`

Use Faust for stream processing.

Do not use the true timestamp of the data but give them the current timestamp (turns out that Faust has problems with processing data with historic timestamps). You also do not have to wait for 24 hours to calculate one hourly value. Just wait to stream 12 values to calculate the hourly value.

- Compute the hourly temperature (hourly mean) for each station.
- Stream temperature data from the three stations and report the station with the highest hourly temperature (use the subhourly data).
- Implement an algorithm that detects outliers in the temperature data stream.

Write the processe data (hourly temperature, the highest temperature and outliers) back to Kafka.

The configuration of producer and consumer are given on Učilnica in the file *konfiguracija.py*.

Submit a Jupyter notebook with your code and "report".

Your Jupyter notebook should contain: problem description, short description of the solution, and conclusions.