Systems and Toolchains Course Project – Option 2

Deadline: November 16th, 11:59pm ET

Accept this assignment by accessing GitHub classroom via the following URL: https://classroom.github.com/a/yDpbj8_M

In this project, you will use MQTT Dataset available on Kaggle

https://www.kaggle.com/datasets/cnrieiit/mqttset

This dataset contains machine learning techniques on MQTT. In this project, you will need the dataset to conduct the following tasks.

General Expectations

- Follow coding best practices with well-documented code.
- Add your dataset under a new **data folder** on GitHub repository
- You may choose 1 peer to work with on this project.
- If you choose to work with peer, write the name of your peer in the Canvas submission. If you fail to do so, your peer will not get the grade.
- We will not handle cases where students forget to submit the name of their peers.

Task-I: Build and populate necessary tables (30% of course project grade)

- Ingest both train and test data into one Postgres Database table. Use the augmented datasets that are provided under Final CSV folder.
- Add a field to your database table that distinguishes between train and test datasets.
- Identify constraints as needed and document them in your Readme.md file.
- Your tables should be created in schema with the name “mqtt”.
- In your ReadMe.md, add a description for the features in the dataset.
- Use the reduced version of the data if your laptop’s memory can’t handle the original dataset.

Task-II: Conduct analytics on your dataset (20% of course project grade)

Develop Python functions that run Spark to answer the following questions. All of the core analysis and data ingestion should be conducted via PySpark. Ingest all the data to answer the following questions from the Postgres Database table.

1. What is the average length of an MQTT message captured in the training dataset?
2. For each target value, what is the average length of the TCP message? (Conduct this process programmatically and don't hardcode any of the target values in your command)
3. Build a Python function that uses PySpark to list the **most frequent X** TCP flags where **X** is a user-provided parameter.
   - Make sure to handle this scenario as well: if the user requests 5 most frequent TCP flags but there are 3 Flags that share the same count at rank number 5, please include all of them in your output.
4. Among the listed targets, what is the most popular target on Google News? (Use 5-minutes Google News feed to justify your answer).
   - Use this query: https://news.google.com/rss/search?q=popular+cyber+attacks
   - You may find yourself in need to decrypt the target values in the dataset to proper English equivalent. For example, "bruteforce" to "brute force".

Task- III Machine Learning Modeling (30% of course project grade)

- Build machine learning models that can predict whether there is an attack and the type of the attack
  - Use proper feature engineering principles (including data cleaning and data engineering)
  - Build two versions: one in Spark and the other one in PyTorch or Tensorflow .
  - For each version, choose two different classifiers/regressors. You can use the same two choices for Spark and PyTorch/Tensorflow, and neural networks of substantial different structures (deep vs shallow, MLP vs CNN) count as two different classifiers/regressors. For each classifier/regressor, identify a few tunable parameters for your model and tune the parameters (using proper metric(s)). Then, run the best model (after tuning) on the test data set and record the test accuracy.
  - In your ReadMe file, explain why you chose the classifiers/regressors and provide comments on the impact of the tunable parameters on the accuracy. Also, compare the selected models.

Task- IV Deploy your code to the Cloud: (10% of the course grade)

- Run a version of your code for the three tasks above on the cloud.
- In this version, you may skip the creation of the Database on the cloud (i.e. on the cloud version, you don't need to write data to table for simplicity). You may ingest the data from CSVs directly.
- If you run the PostgreSQL on the cloud: you will receive **10% extra-credit**.

Course Project Checkpoint is planned for **October 12th** and will constitute **10% of the course project grade**. In your course project checkpoint, submit **Task I** and **Task II**.

Submission Guidelines:

- You MUST use the GitHub classroom URL to create your repository. Post your GitHub repository's URL created via GitHub classroom to Canvas. Use the starter code that is provided above as the starter for your code.
- Your GitHub repository should have a ReadMe.md file that lists the "exact" steps on how to run your code and the input configurations associated with it. I will follow the steps in your ReadMe file and if I can't get it running on my machine, I will deduct considerable number of points from your project grade.
- **You should record a video demonstrating two elements:**
  1. Code Walkthrough while you are explaining your code changes.
  2. Demoing the running application while you are navigating through EVERY functionality that is working in your application. I will use this video to help assessing your grade. You may lose points for the functionalities that are not demonstrated in the demo.
- Your video size may be large to be uploaded to GitHub. You may use Box to upload the video and add the URL to your ReadMe.md file in your GitHub repository.
  1. Make sure that your video is publicly shared. Private videos won't be visible to the instructor and TAs and therefore, your project grade will be impacted

Grading Notes:

- Unlike HW-assignments that has 20% late penalty, late project submissions on Canvas or GitHub **will receive 0 points** (won't be graded)
- **Not submitting the GitHub video (for both code walkthrough and functionality demo): you will get up to 80% of the maximum grade.**
- Not providing clear details in the ReadMe file on how to run the application (or any variables that need to be updated/replaced): **you will get up to 90% of the maximum grade.**

Refer to Course Syllabus for planned course project checkpoints.