Course Projects for Data Engineering-II Course

Projects

For this course, we provide three projects applied to real-world data, which will help you improve your understanding of the concepts learned during this course. Furthermore, you will learn how large amounts of data is collected and analyzed.

Prerequisite for projects

These projects aim to test student's fundamental data engineering and analysis skills which include the following;

1. Reading and understanding of APIs (Project-1, Project-2, Project-3).
2. Developing algorithms for handling large data from external APIs (Project-1, Project-2, Project-3).
3. Basic knowledge about system architecture design and deployment strategies using contextualization, and orchestration (Project-1, Project-2, Project-3).
4. An understanding of machine learning models, which will help identify a model suitable for prediction (Project-3).
5. A good understanding of the streaming framework as well as container orchestration techniques (Project-2, Project-1).

Visibility of the source code

The project should be kept ***private*** on a repository service website (e.g., GitHub). It is important because students' GitHub credentials will be required in order to access the GitHub API (e.g., GitHub authentication token in their source code).

Demonstration / Presentation

You are required to present the work on 31.05.2021.

Group

A group can consist of a maximum of 4 students.

Resource allocation

4 to 8 VMs can be used by each group

Best of Luck!

Project 3: Evaluating the accuracy of prediction of stargazers in open source projects

---

With the advent of social media, the idea of popularity is used to identify who is performing better than the others. For example, YouTube subscribers can be used as a direct measure for a channel's popularity. Similarly, GitHub users show their appreciation for a project by putting stargazer (or star) to the project. Therefore, the number of stargazers indicate the popularity of a GitHub project.

In this mini research project, your task is to study which prediction model has the highest accuracy in predicting the number of stars for a GitHub repository. For this task, several steps are required to complete this project.

Dataset: The set will consist of historical data on top 1000 open source projects with at least 50 stars (or based on your own observation after looking at the star count of repositories). The data can be collected by using GitHub API. Please look into the following links of how to use the API and query GitHub for getting 1000 projects by star count:

https://docs.github.com/en/rest
https://docs.github.com/en/rest/overview/resources-in-the-rest-api#pagination
https://docs.github.com/en/github/searching-for-information-on-github/searching-for-repositories#search-by-number-of-stars

Feature extraction: GitHub API will return a number of activities and information about a repository e.g., commits, forks, watchers etc. Use these activities as features to predict the star count of a given GitHub repository. Select as many features as possible. (**Optional:** perform feature engineering if possible).

Prediction technique: Apply different prediction models and train the models.

Accuracy of prediction: Use R-squared to evaluate the accuracy of the prediction.

Star Predictor App: Build an execution pipeline using GitHooks. The pipeline will provide continuous integration and delivery of machine learning models to a production environment. Therefore, create and start two VMs with Ansible orchestration environment. One server will be used for development and the other will be for production. Model training and comparison between models will be done on the development server. The model with the best accuracy will be pushed to the production server. Run your star predictor application (i.e., the selected model) on the production server and provide some activities e.g., commits, forks, watchers etc. of 5 GitHub repositories to the model which will rank the repositories according to their star count.

Project deliverable

a. Provide Docker containers and scripts for running the project.
b. Four-page scientific report having the following section:
    1. Introduction
    2. Related work
    3. System architecture

4. Results
    1. Scalability analysis
    2. Compare the models you have chosen by reporting their accuracy.