CS 442/642

Cloud Computing - Fall '21

Programming Assignment 2 – Due 12/7, 5PM (no extensions)

Goal: The purpose of this individual assignment is to learn how to develop parallel machine learning (ML) applications. Specifically, you will learn: (1) how to use Spark's MLlib to develop and use an ML model; (2) How to use Docker to create a container for your ML model to simplify model deployment.

Description: You have to build a wine quality prediction ML model in Spark (You are welcome to use AWS if you want, but using personal computer is also fine.) Then, you need to save and load the model in an application that will perform wine quality prediction. The assignment must be implemented in Java, Scala, or Python on Ubuntu Linux. The details of the assignment are presented below:

1. Input Data: I shared a dataset (winequality-white.csv) with you for your ML model. Each row in a dataset is for one specific wine, and it contains some physical parameters of the wine as well as a quality score. It is available in Canvas, under Programming Assignment 2. You will have to split this dataset into two as training (80%) and test (20%), so that you will use training split to train your models and test split to evaluate them.
2. Output: The output of your application will be a measure of the prediction performance, specifically the F1 score, which is available in MLlib.
3. Model Implementation: You have to develop a Spark application that uses MLlib to train for wine quality prediction using the training dataset. You will use the test dataset to check the performance of your trained model and to potentially tune your ML model parameters for best performance. You will develop at least two ML models (linear regression and random forest regression) and tune their hyperparameters to get the best performance out of them. As part of submission, you will present the f-1 scores for linear regression and random forest models after tuning their hyperparameters. Note: there will be extra-credit for the top 3 applications/students in terms of prediction performance (see below for grading). This can be achieved either tuning random forest and liner regression models or developing other models such as XGBoost or Random Forest classifier. For classification models, you can use 10 classes (the wine scores are from 1 to 10).
4. Docker container: You have to build a Docker container for your prediction application. In this way, the prediction model can be quickly deployed across many different environments. Since Docker will be covered in class later, we posted the slides that will help you get started with Docker under Programming Assignment 2 in Canvas.

Submission: You will submit in Canvas, under Programming Assignment 2,

1. A text/Word/pdf file that contains:
    - A link to your code in GitHub. The code includes the code for parallel model training and the code for the prediction application.
    - A link to your container in Docker Hub.
    - Step-by-step instructions on how to download your image from Docker Hub and run it as well as steps to run your code in the container

2. A video recording (at most 10 minutes) to cover followings:
    - Briefly explain your code
    - Download your image from docker hub and run it (First show that the image is not locally available i.e., docker image ls) then download and run it
    - Run your prediction code – Since training models can take long time, you are expected to save your model beforehand and load the pre-trained model in the demo instead of re-training it.

Grading:

- Implementation of wine quality prediction model in Spark – 30 points
- Hyperparameter tuning for the model – 20 points
- Docker container for prediction application – 50 points
- Extra-credit for top 3 prediction performance – 20 points