Assignment #1

This assignment can be completed individually or by a team (maximum of 3 members).
Total score: 90
Due date: refer to the Canvas page

Objectives

To perform basic machine learning modeling and benchmark computations using GPUs for parallel processing

Only Python programs written using Python 3.0 or higher will be accepted. **NO Jupyter notebook or any other Python variant will be accepted** for efficient grading.

Required activities

Write a concise report presenting the results of machine learning models for regression, classification, and clustering, both without GPU acceleration and with GPU-accelerated packages such as **CuPy** and **NumBa**, along with a comparison of the performance benchmarks obtained using your local computer (or any locally available computer) and the Nautilus GPU cluster.

The report should include:

- Team member names, optionally specifying each member's percent contribution or stating "Everyone contributed equally" if applicable. If your team does not reach an agreement on individual contribution, briefly write a task description for each member. Different grades may be assigned based on individual contributions.
- The results of modeling and computational performance based on the tasks outlined below

You may reuse publicly available source codes from the Internet or get help from Large Language Model; however, sharing your code with other teams or students in the class is strictly prohibited. Any student or team violating this policy will receive a zero for this assignment and may face penalties on all the remaining assignments.

Regression

1. Choose a linear regression method to learn the best polynomial function of order 3 from the dataset "Household Electric Power Consumption" and compare the computational performance between your local computer and the GPU cluster by completing the following tasks. [20]

More information about the dataset and Python code to import it can be found at (https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption).

(a) Briefly describe any data preprocessing methods you applied before modeling, with justification in no more than 1 – 2 paragraphs.

(b) Split the dataset into 80% for training and 20% for testing, perform the modeling using three approaches – without GPU acceleration and with two GPU computing packages, and compare the results based on the following metrics.

Coefficient vector, training RMSE and R^2, testing RMSE and R^2

(c) Briefly describe the algorithm you used to maximize computing resource utilization on both your local machine and the GPU cluster, focusing on parallelization to speed up the computations. When you use the GPU cluster, containerize your programs and deploy them using Kubernetes command line interface.

(d) Specify the operating system and provide the hardware specifications, including:

- CPU name, total number of CPU cores, CPU clock speed, RAM size
- GPU name, number of GPUs, GPU clock speed, number of CUDA and/or tensor cores, GPU memory size (VRAM). If your computer lacks a GPU, simply state “no GPUs available.”

If utilizing the GPU cluster, the YAML file should clearly define the computing resources.

(e) Compare the computation time for the modeling with three approaches – without GPU acceleration and with two GPU computing packages–between your local computer and the GPU cluster.

If the modeling on your computer takes more than 30 minutes, you can stop it and indicate that it took >30 minutes.

Python uses 64 bits by default to represent floating-point numbers. Is the computation performance improved if you change it to 16bits, e.g., np.float16()?

Classification

2. Use the Logistic Regression method to learn the classifier from the dataset "CDC Diabetes Health" and compare the computational performance between your local computer and the GPU cluster by completing the following tasks. [20]

More information about the dataset and Python code to import it can be found at (https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators).

(a) A brief description of preprocessing method as specified in 1.(a)

(b) The modeling results as specified in 1.(b) based on the following metrics.

Confusion matrix, % accuracy, and precision

(c) A brief description of the algorithm for parallel processing as specified in 1.(c).

(d) Performance comparison as specified in 1.(e).

Clustering

3. Use K-means method to cluster the same dataset "CDC Diabetes Health" used for classification by ignoring the dependent (label) variable, and compare the computational performance between your local computer and the GPU cluster by completing the following tasks. [20]

(a) Define the similarity measure(s), perform clustering using three approaches – without GPU acceleration and with two GPU computing packages, and compare the results based on the following metrics.

Internal index using RMSE and external index using the class labels in the dataset

(b) A brief description of the algorithm for parallel processing as specified in 1. (c).

(c) Performance comparison as specified in 1. (e).

What and how to submit the assignment

- One report in Word or PDF format per team
- Only the program file(s) created by the team (excluding any third-party packages).
- YAML file(s)
- Dockerfile(s) (or equivalent container configuration files)
- Upload each file separately. DO NOT submit a ZIP file as Canvas cannot open ZIP files.

Grading criteria

- 90% based on the overall quality of work, including the analysis process, modeling results, code implementation, the depth of understanding demonstrated in the report
- 10% based on the effort reflected in the report and program development