**CSYE7105 - High Performance Parallel Machine Learning & AI, Fall 2024**

**Sentiment Analysis on Amazon Customer Reviews Using Parallel Deep Learning**

**Team 4:** Anshul Chaudhary and Dev Mithunisvar Premraj

**Instructor:** Prof. Handan Liu

**Introduction**

**Background:**

In today's digital marketplace, Amazon reviews significantly influence consumer purchasing decisions and determine product success. These reviews provide a wealth of information, encompassing customer experiences, product strengths, and areas for improvement. With over 3.5 million Amazon reviews sourced from the Amazon Reviews dataset on Kaggle, there is a substantial opportunity to leverage this data for valuable insights. Each review record includes essential details such as product reviews and ratings offering a comprehensive foundation for in-depth analysis.

**Motivation:**

Analyzing Amazon reviews for sentiment is pivotal in understanding customer satisfaction, preferences, and emerging trends. However, processing and extracting meaningful sentiment from such a large and unstructured dataset presents significant challenges. Efficient data cleaning and transformation are critical to preparing the data for accurate sentiment classification. Although the current data preprocessing using Pandas takes approximately 2 minutes for 3.5 million records, the data's growing volume and complexity necessitate further optimization. Additionally, the subsequent sentiment analysis process can extend beyond several hours due to the dataset's size and complexity. To address these challenges, implementing parallel processing techniques is essential to enhance processing speeds and ensure scalability for even larger datasets

**Goal:**

Our primary objective is to harness parallel computing to enhance the speed and efficiency of sentiment analysis on Amazon reviews. Specifically, we aim to:

1. **Maintain and Optimize Preprocessing Efficiency:** Ensure that data cleaning and transformation remain swift for 3.5 million records while preparing the system to handle larger datasets seamlessly. Integrate Dask, a flexible parallel computing library, alongside Pandas and sklearn preprocessing to distribute computational tasks across multiple cores, thereby reducing preprocessing times further.
2. **Enhance System Scalability and Efficiency:** Continuously experiment with and refine parallel computing methodologies to achieve optimal speed-up and resource utilization. Identify and tune the most effective parallel computing strategies and parameters to maximize performance and accommodate future data growth.

By achieving these goals, we aim to build a robust and scalable sentiment analysis system capable of efficiently handling large-scale Amazon review data. This system will provide timely and accurate sentimental insights, enabling informed business strategies and fostering improved customer satisfaction.

**Methodology**

To achieve our objectives, we will employ a structured methodology encompassing data preprocessing, parallel processing optimization, and sentiment analysis. The key components of our approach are outlined below:

1. **Data Cleaning and Transformation:**

* **Libraries:** Pandas, NumPy, and Dask.
* **Data Splitting:** Each input line in the dataset represents a single review. We will split each entry into a label field (for sentiment) and a text field (for review content).
* **Label Assignment:** Reviews are classified as either negative or positive sentiments based on star ratings: \_\_label\_\_1 for 1-2 stars (negative) and \_\_label\_\_2 for 4-5 stars (positive).
* **Special Character Removal:** Review text will be cleaned by removing special characters, ensuring a consistent format for model input and improving downstream performance.
* **Tokenization:** Text tokenization will transform cleaned reviews into numerical sequences suitable for model input.
* **Optimization with Dask:** Using Dask, we will parallelize each preprocessing step by breaking the dataset into smaller, independent chunks processed simultaneously. Dask's flexible parallel computing allows us to utilize multiple cores, significantly speeding up processing and reducing memory overhead, ensuring scalability even if the dataset grows beyond in-memory limitations. We aim to reduce the current 2-minute processing time by optimizing chunk size and parallel parameters, enhancing scalability for future data expansion.

1. **Sentiment Analysis with LSTM:**

* **Libraries and Frameworks:** PyTorch, Dask.
* **Model Development:** We will build a Long Short-Term Memory (LSTM) neural network, as LSTMs are well-suited for sequential data and can effectively capture dependencies in textual data, essential for sentiment classification tasks.
* **Training Process:** The LSTM model will be trained on the preprocessed dataset to classify sentiments expressed in reviews. To handle the high memory and computational demands of the large dataset, we will leverage DDP (Distributed Data Parallel) in PyTorch for parallelized training.
* **Parallelized Model Training with PyTorch DDP:**
  + **DDP Implementation:** PyTorch's Distributed Data Parallel (DDP) framework will allow us to distribute the LSTM model across multiple GPUs or CPUs. During training, DDP synchronizes gradients across devices after each batch, ensuring that model parameters are updated consistently across all parallel instances. This approach is more memory-efficient and faster than traditional training, significantly reducing training time.
  + **Gradient Synchronization and Scalability:** With DDP, each device processes a mini batch of data, followed by gradient synchronization to ensure that model updates are consistent across all devices. This parallelized training strategy not only enhances speed but also provides scalability, allowing us to handle larger datasets without resource bottlenecks.

1. **Model and Parallel Computing Performance:**
   * **Accuracy and F1 Score:** We will evaluate model performance using accuracy and F1 score, critical for assessing the LSTM’s effectiveness in classifying sentiment across a diverse dataset. The F1 score is valuable for this imbalanced dataset as it balances precision and recall.
   * **Training and Validation Loss:** Loss curves will be tracked to monitor model convergence, identifying any trends of overfitting or underfitting during training.
   * **Speedup and Efficiency of Parallelization:** To quantify the effectiveness of our parallelization efforts, we will measure processing and training time reduction compared to non-parallelized baselines. Using Dask and PyTorch DDP, we aim to achieve significant speedup in both the preprocessing and training stages as dataset size and available resources increase, ensuring the system remains robust and scalable.

**Dataset Description**

The dataset, sourced from Kaggle, consists of a few million Amazon customer reviews and their associated star ratings, which serve as output labels. The size of the dataset is around (516.93 MB) The data has been preprocessed for fastText format, with reviews classified as either \_\_label\_\_1 (negative: 1-2 stars) or \_\_label\_\_2 (positive: 4-5 stars). Neutral reviews (3-star) are excluded, ensuring a clear separation between positive and negative sentiment.

Train and Test data shape:

* Train data shape (3599999, 1)
* Test data shape (399999, 1)

**Data Source**

https://www.kaggle.com/datasets/bittlingmayer/amazonreviews/data

**Reference (optional)**

https://www.kaggle.com/code/reemmuharram/amazon-reviews-sentiment-ananlysis-lstm#1--Libraries