**Task 3 Submission: Code submissions**

**Project Group Partners:  
Muddasir Rizwan -24494  
Muhammad Hadi- 25244  
Ahmed Raza-25255**

**Assessment of Execution Time: Synchronous Implementation:**

The procedure for the undistributed technique entails going through a number of processes to analyse a dataset of movie reviews, which include data cleansing, tokenization of words, Word2Vec feature extraction, and RandomForestClassifier sentiment classification. The size of the dataset and the system both affect how long the full procedure takes to execute. But when dealing with big datasets, processing jobs one after the other in a sequential fashion might cause delays.

**Implementation in Parallel:**

Utilising distributed computing, the workflow is modified to execute on Apache Spark in the parallel way. This method drastically cuts down on execution time by dividing the dataset into training and testing sets, parallelizing the processing jobs over several Spark nodes, and using PySpark features. The time needed to process each stage—tokenization, feature extraction, and classification—decreases due to distributed computing's parallel execution capabilities. For example, while the sequential approach may take several minutes to process, the parallel approach can cut this time down by running tasks concurrently.

**Changes for the Parallel Approach: The main adjustments consist of:**

1. Distributed Computing: By switching from Pandas to PySpark, it is possible to divide the processing burden across several nodes. This makes it possible to handle more data at once and speeds up each step.
2. Communication and data transmission between nodes are made possible by the introduction of a SparkSession, which establishes a Spark context.
3. Pipeline: Tokenization, stopword removal, Word2Vec vectorization, and classification processes may be streamlined by organising the workflow in a linear fashion using the Spark MLlib pipeline feature.
4. Training of models in parallel: Depending on the number of nodes, this decreases the time complexity from linear to almost constant.
5. Scalability: By expanding over more nodes, parallel processing can handle larger amounts of data and more complicated calculations..

**Future Parameter Considerations:**

1. Data Partitioning: Vary the train-test split ratio to track variations in model execution time and performance.
2. Pipeline Configuration: Test various feature engineering and classification methods to observe how they affect processing accuracy and performance.
3. Node Configuration: To test the impact of changing node settings (memory, CPUs) on execution time, increase the number of nodes in the Spark cluster.
4. Hyperparameters of the model: To maximise processing, adjust hyperparameters like Word2Vec window size and RandomForest model depth.

**Conclusion:**The substantial performance gains from distributed computing are seen when synchronous and parallel execution are compared:  
  
Execution Time: By cutting down on total execution time, the parallel technique speeds up the processing of bigger datasets.  
Project Objectives: By effectively analysing text sentiment, parallel processing helps the project achieve its objectives and opens the door for further in-depth studies in later iterations.  
Future Development: The knowledge gathered will direct changes to data processing pipelines in the future, guaranteeing scalable and effective solutions for text categorization jobs.