BIG DATA – SAT 5165

1. **Oscar Odera**
2. **Miltone Awiti**
3. **Eric Fosu-Kwabi**

Spark for Big Data Preprocessing

We focused on key preprocessing tasks to ensure the dataset containing 292,364 records was clean and ready for future analysis. The main tasks involved handling missing values, encoding categorical variables, and selecting relevant features for gender-based mental health treatment patterns.

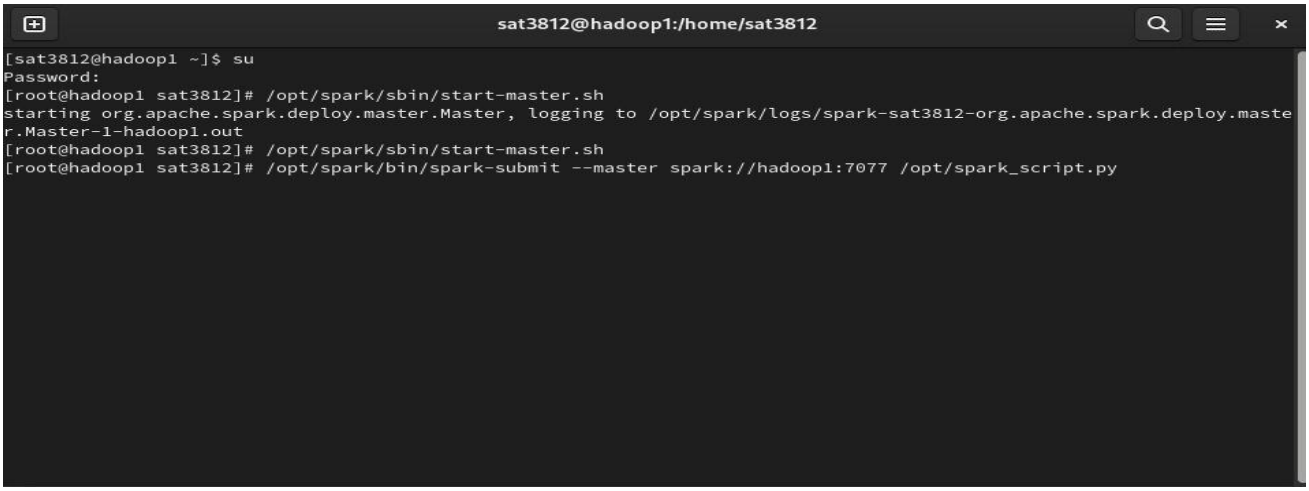
The dataset was loaded into Spark using PySpark's DataFrame API, which allows for distributed data processing across multiple virtual machines. This initial exploration allowed us to inspect the structure of the dataset and ensure that the appropriate data types were inferred.One of the key preprocessing steps was to handle missing values in the dataset. We filled missing numerical values, such as age, using mean imputation and filled categorical values with "Unknown."This ensured that the dataset did not have gaps that could lead to issues during analysis.

Categorical variables were encoded using *StringIndexer* to assign a numerical index to each unique category. Following this, *OneHotEncoder* was applied to convert these indices into binary vectors. This transformation was necessary to convert string-based categorical data into a format suitable for machine learning models.

The final step in preprocessing was assembling the features into a single vector column. This assembled the encoded categorical variables and the numerical Age variable into a feature vector.

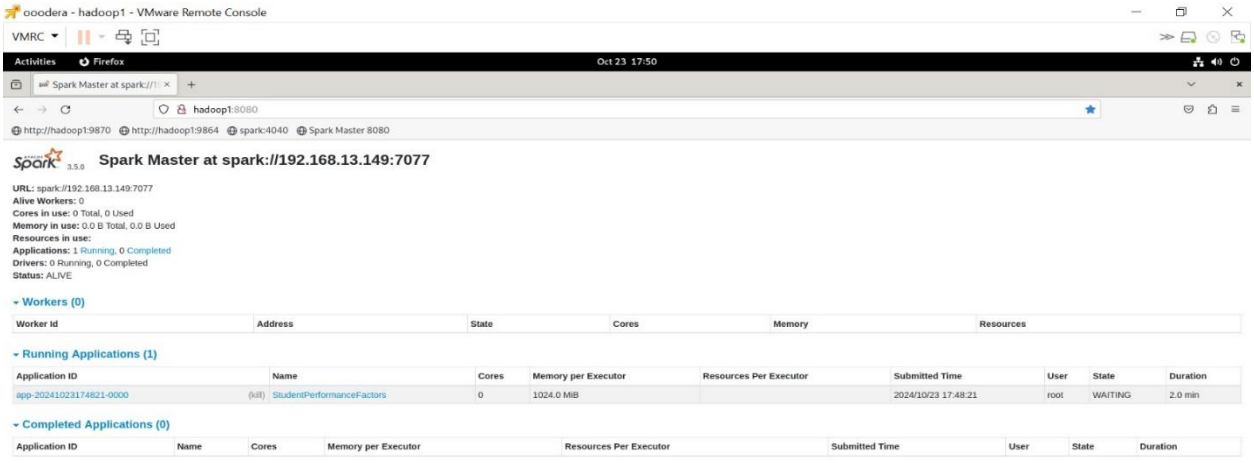
Performance Comparison

The preprocessing performance was tested under different conditions to assess how efficiently the task could be completed in a distributed environment:

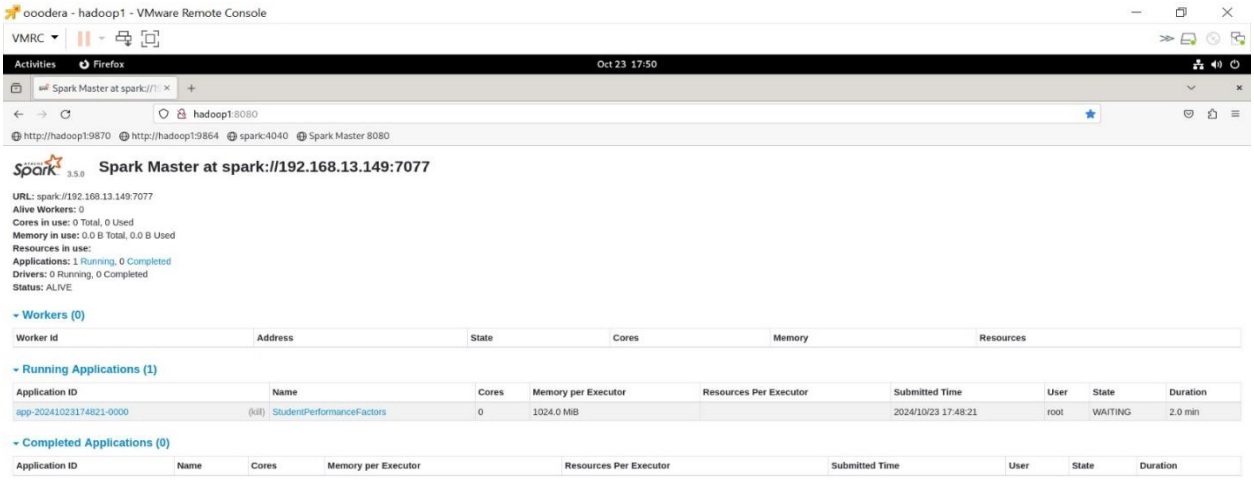


Using the Master Node Only

We started with master node to check on the performance



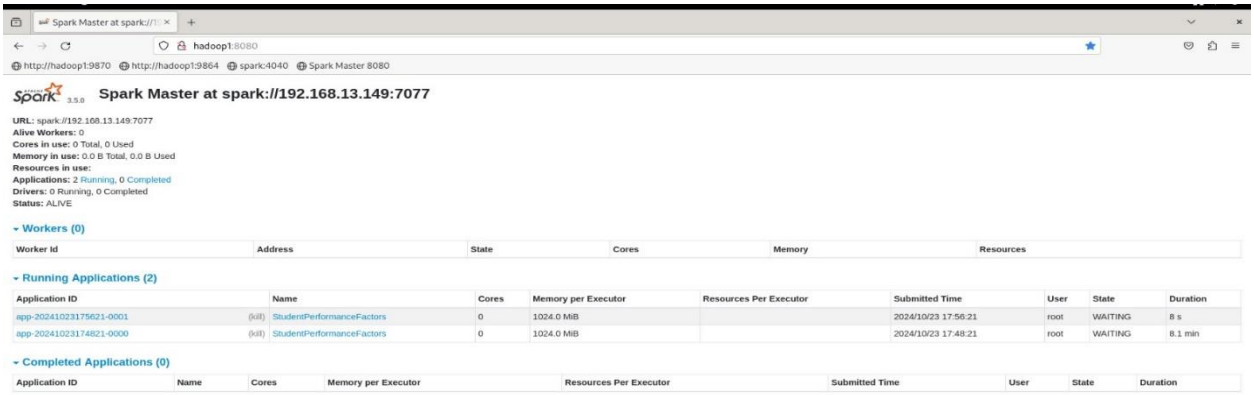
- **Execution Time**: 2 minutes
- **Observation**: Running the preprocessing tasks on a single master node resulted in a relatively longer execution time due to the workload being handled by one machine.



Using the Worker Node

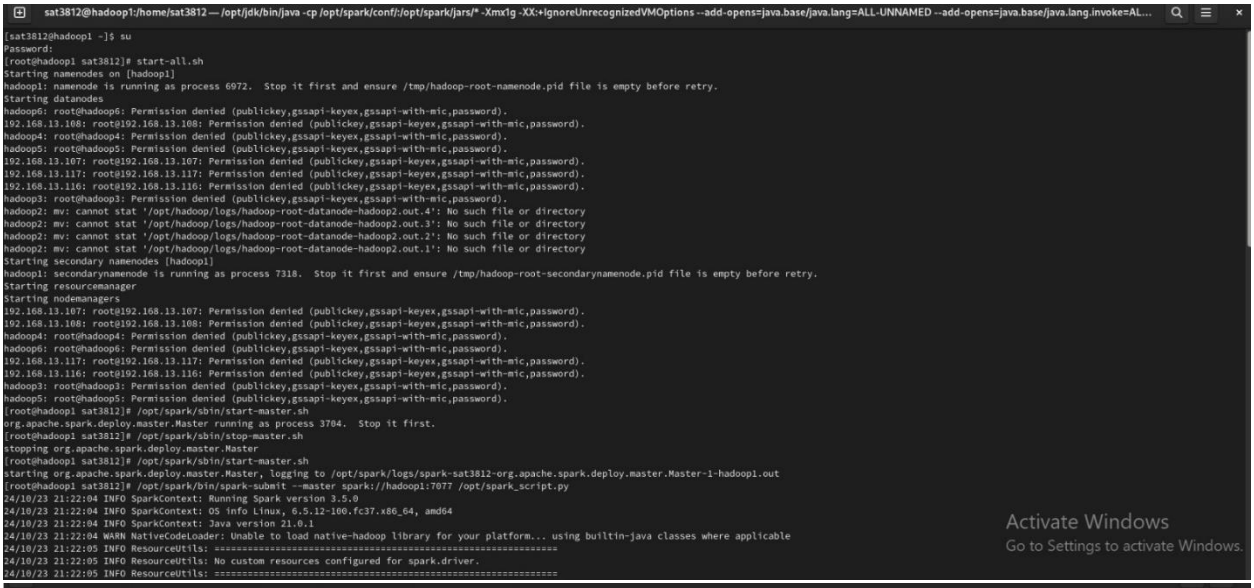
Execution Time: 8 seconds

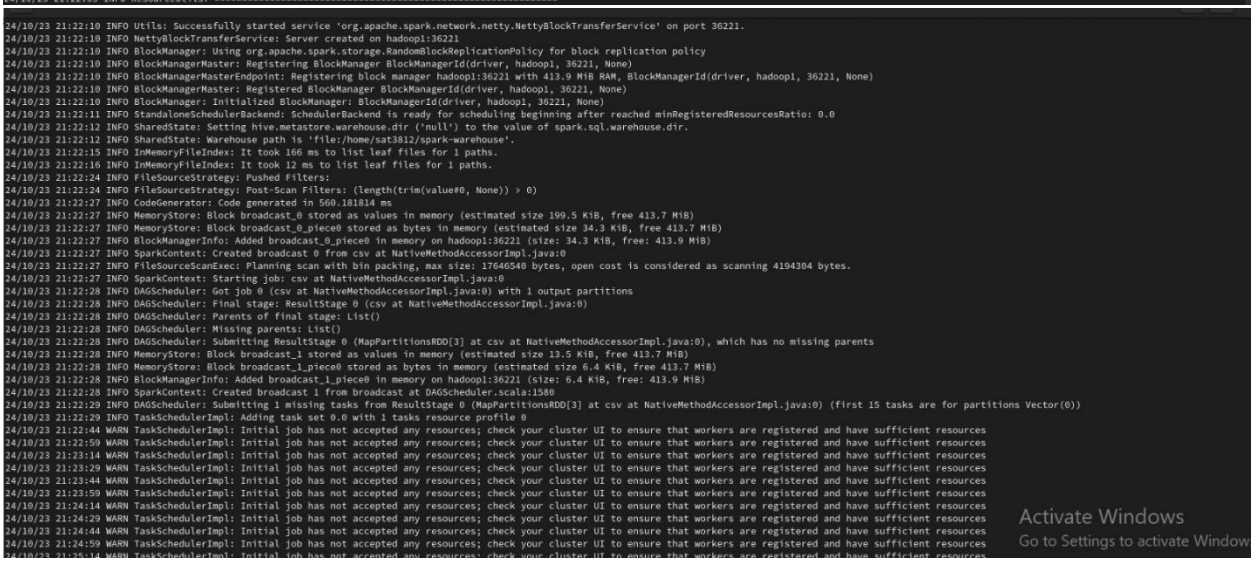
Observation: The task was completed significantly faster when distributed across the worker node, showing a drastic reduction in execution time. This demonstrates the clear benefit of distributed computing in handling large datasets and heavy preprocessing tasks.



Limitations:

- We encountered connection issues that prevented us from fully utilizing all six virtual machines. Errors occurred during the setup of some of the VMs, restricting us from running the job across multiple nodes simultaneously. Despite this, the speed improvement observed with two VMs shows the potential for even greater efficiency with more robust connections.





The project codes can be accessed on;