# Performance Analysis

This document provides an analysis of performance metrics for Pandas and Dask.

## Execution Time Comparison

|  |  |  |
| --- | --- | --- |
| Process | Pandas Time (s) | Dask Time (s) |
| Load time | 1.12 seconds | 1.74 seconds |
| Preprocessing time | 0.01 seconds | 0.11 seconds |
| Data exploration time | 0.48 seconds | 1.72 seconds |
| Preprocessing (train/test) time | 0.01 seconds | 3.18 seconds |
| Modelling time | 0.12 seconds | 1.53 seconds |
| Total time (measured) | 1.91 seconds | 14.00 seconds |

## Memory Usage

|  |  |  |
| --- | --- | --- |
| Dataset | Pandas (MB) | Dask (MB) |
| Clinical | 0.08 MB | 0.08 MB |
| Expression (partitioned) | N/A | 7.49 MB |
| Expression | 45.54 MB | 45.54 MB |

As we can see in the tables above, the execution of transformations in Pandas takes less time than in Dask, but Dask can store information in memory partially and process it in parallel and only as much as needed, which allows it to handle large datasets. Probably the amount of data is too small to clearly demonstrate the speed of parallel operation.

## Model Performance Comparison

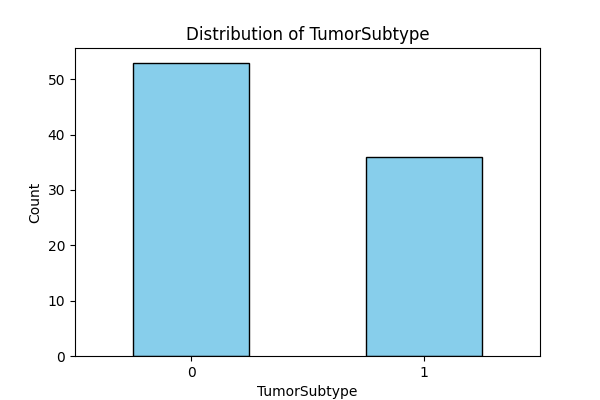
|  |  |  |
| --- | --- | --- |
| Metric | Pandas | Dask |
| Accuracy | 0.83 | 0.85 |
| Precision | 0.78 | 0.75 |
| Recall | 0.78 | 0.90 |
| F1-score | 0.78 | 0.82 |
| Auc-roc | 0.84 | 0.89 |

The models results show that despite the longer train/predict Dask model time, many indicators are higher for this model.  
However, here we should make some remarks: the function dask\_ml.model\_selection.train\_test\_split does not have an argument stratification, which could affect the final result because our dataframe is unbalanced. The main work was done on a Windows computer, but "Windows is not officially supported for dask/xgboost".

|  |  |
| --- | --- |
|  |  |

Finally, if your dataset is not too large and your computer's capabilities allow you to work with it in pandas, you should think twice before switching to Dask. Some pandas features are not implemented in Dask (for example, transpose or axis operations), which can lead to difficulties and the necessity to look for solutions. However, when it comes to really large amounts of data that don't fit in RAM, Dask becomes a great solution, allowing you to work with the data more efficiently through parallel computation and lazy loading.

Group patients by clinical variables and compute the average expression of selected genes



## Average expression of top 10 variable genes within TumorSubtype groups

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TumorSubtype | merck2-AA622265\_at | merck2-NM\_005554\_at | merck-NM\_003125\_at | merck-NM\_000424\_at | merck2-NM\_006945\_x\_at | merck2-NM\_005555\_at | merck-CR749383\_at | merck2-NM\_000424\_at | merck2-NM\_175868\_s\_at | merck-NM\_005362\_s\_at |
| 0 | 5.53 | 6.07 | 5.5 | 5.81 | 5.39 | 3.73 | 6.18 | 5.86 | 5.56 | 5.17 |
| 1 | 5.92 | 11.74 | 10.08 | 11.94 | 10.17 | 9.62 | 4.15 | 11.81 | 6.24 | 5.83 |