Week 6 Deliverables: File Ingestion and Schema Validation

# Introduction

**File Ingestion** and **Schema Validation** are both essential steps in processing large datasets within the Ingestion pipeline.

**File Ingestion** involves:

* Taking data from an external source.
* Loading it into a system/ application for further analysis.
* The file (normally .csv, excel, .json) format should be accessible for various operations, such as analysis, transformation, and reporting.

**Schema Validation** involves:

* The process of ensuring the data being ingested and imported into a system adheres to a predefined structure or schema.
* Schemas define the expected format, data types and rules for the data within the dataset.
* Schema validation checks to see if the file imported in the File Ingestion phase matches the expected structure, verifying column names, data types etc.

The **Purpose** and **Importance** of this is:

* **Errors/ Quality Assurance**: Ensure the data adheres to the schema maintaining quality and consistency, along with also having a reduced chance of errors, such as missing/ mismatched data. It can enable data engineers to heck for any anomalies/ discrepancies.
* **Efficiency/ Documentation**: Serve as a documentation for the expected structure, making it easier for team members to work with along with saving time and resources with data pre-processing and analysis.
* **Data Integration**: Organisations use data from multiple sources, hence a data schema is useful in ensuring that data from each source aligns with its common structure.

# Computational Efficiency

## Dask

This is a parallel and distributed computing algorithm which is designed to enable scalable and flexible handling of larger datasets which take up more memory. It downloads datasets very quickly and can download much larger datasets.

## Modin

Modin is another pandas distributed and computing algorithm which enhances the performance of Pandas-based data analysis by leveraging parallel and distributed computing techniques. It is useful in speeding up data operations without the need for extensive code modifications, to work with Pandas and large datasets.

## Ray

Ray is an open-source Python framework for building distributed and parallel applications. It is a powerful choice for building scalable, high-performance applications, in the realm of distributed systems, machine learning, and data preprocessing.

Out of all the libraries, Dask had the fastest computational efficiency.

## YAML Files

The YAML library is used primarily for Schema Validation. This involves validating the number of columns and column names so that the Yaml Schema for the data matches the dataframe we want.

This is a **human-readable data serialisation format**, used in configuration files and data interchange between programming languages and different data structures. It is easy for both humans to read/write and machines to parse/generate.

YAML is commonly used for configuration files in applications, as well as defining data structures within tools like Docker Compose and Kubernetes. The simplicity and human-friendliness makes it a popular choice for data serialisation needs in software development and system administration.

It uses indentation, and simple punctuation to structure data, supporting data types, comments, nested structures, extensibility. It is also language-agnostic, so it can be used with multiple programming languages.

* The YAML configuration file serves to store settings and parameters used via the code and is a structured text file defining options for the programme.
* The YAML configuration file is used to separate configuration parameters from the code itself, making it easier to change settings.
* Once we have configuration settings set, we can construct a source file, which is a line of code which we use to read a csv file and then use a delimiter specified in configuration to parse the data.

Example of Configuration File:

A screenshot of a computer code

Description automatically generated

## GZIP Files

We put files in a **GZIP compression** algorithm, which is used as a file compression format to reduce the size of the files, especially text-based csv, json etc.

**GZIP compression** algorithms are used because:

1. Reducing the file size saving storage space and this makes the data transfer faster.
2. Easier archives in the long-run as they take up less space and better for **backup compression**.
3. Some **data pre-processing** tools can work with GZ compressed files – saving **time and storage** and they can be shared with colleagues more easily.