**Extract-Transform-Load**

In any machine learning problem, the first questions that must be asked concern *data*. Where does it come from? How is it structured? What transformations must be applied to convert it into a form suitable for a machine learning model? How should it be stored?

The answers to these questions fall under the purview of the Extract-Transform-Load (ETL) pipeline. [[1]](#footnote-1)

Answering these questions is the role of the Extract-Transform-Load pipeline. Each step belies considerable complexity.

Or the Extract-Load-Transform pipeline. The sole difference is where the transformation takes place. In ELT, the transformation occurs in the target data store, instead of using a separate transformation engine. This simplifies the architecture by removing the transformation engine from the pipeline. Scaling the target data store also scales the ELT pipeline performance. However, ELT only works well when the target system is powerful enough to transform efficiently.

Such a distinction is largely academic for this dissertation; the process with henceforth be referred to as “ETL”.

Typically ELT use cases fall within the big data realm.

Benefits of ETL: helps to analyse business data for taking critical business decisions.

The question this dissertation is answering: can medium-term train delays be predicted?

Transactional databases cannot answer complex business questions that can be answered by ETL. A data warehouse provides a common data repository.

ETL is data pipeline.

Helps migrate data into a data warehouse. Convert to the various formats and types to adhere to one consistent system. A predefined proess for accessing and manipulating source data into the target database.

Is it better to explain the dataset

Provides a method of moving the data from various sources into a data warehouse. Allows verification of data transformation, aggregation and calculations rules. Sample data comparison between the source and target system.

When creating a data warehouse, it is common for disparate sources to be brought together into one place so that it can be analysed for patterns and insights. Ideally data would have a compatible schema from the outset, but this is rarely the case. ETL takes data that is heterogenous and makes it homogeneous.

# Okay. That’s helpful. Let’s start from the top and work our way down, referencing as we go.

**Extraction**

*Define*

Extraction is the process of collecting data, often from multiple different sources, and moving it to a *staging area*.

“Data extraction presupposes a selection process” (Ponniah, pg. 282). Data must be selected based on requirements.

As Ponniah (xxxx, pg. 284) notes, “it is not uncommon for a project team to spend as much as 50% to 70% of project effort on ETL functions”. Sadly, this was the case in this dissertation.

Problems are numerous and varied.

Data is extracted from the source system into the staging area. If corrupted data is copied directly from the source into data warehouse database, rollback will be a challenge. Validations can be done during extraction: reconcile records with source data. Make sure that no spam / unwanted data is loaded. Data type check. Removal of all types of duplicate / fragmented data. Check whether keys are in place or not.

Extracts the data from different source systems.

The collection of data from various sources.

Retrieving data from external storage or transmission sources

Most data storage projects integrate data received from various source systems. Each individual system may employ a separate data organisation or format. Each individual system may employ a separate data organisation or format. Common data source structures are relational databases and pure data files.

Someone in the organisation identifies the desired data sources and the rows, columns, and fields to be extracted from those sources. These sources likely include transactional databases hosted on-site and in the cloud.

The process of reading data from a database. In this stage, data is collected, often from multiple and different types of sources.

Extracts the data from different source systems.

Part of the planning for this stage should include estimating data volumes for each data source. Data must be extracted in a way that doesn’t have a negative impact on source systems or response times.

Retrieving data from external data storage or transmission sources.

Target sources may include ERP, CRM, streaming sources, other enterprise systems, and data from third-party sources.

The main challenge is integrating with APIs. Integration is different for every application; not every product provides a vanilla REST API. Some REST APIs are surprisingly convoluted, and some applications are still stuck in protocols like SOAP. Many APIs are not rigorously and accurately documented. An API may have a large surface area with dozens of built-in resource endpoints. APIs are constantly changing and breaking.

The first phase of ETL. Data is collected from one or more data sources and held in temporary storage where the subsequent two phases can be executed. During extraction, validation rules are applied to test whether data has expected values essential to the data warehouse. Data that fails the validation is rejected and further processed to discover why it failed validation, allowing remediation if possible.

# <https://www.webopedia.com/TERM/E/ETL.html>

# <https://www.sas.com/en_us/insights/data-management/what-is-etl.html>

# <https://databricks.com/glossary/extract-transform-load>

# <https://www.stitchdata.com/etldatabase/etl-extract/>

# <https://www.informatica.com/services-and-training/glossary-of-terms/extract-transform-load-definition.html>

# <https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl>

# <https://www.techopedia.com/definition/24170/extract-transform-load-etl>

**Extraction for this dissertation**

There are many desired data sources.

Each source

They can be divided into three categories: schedules, movements, and weather. Each one of this is an ordeal in of itself!

**Transform**

*Define*

Needs to be cleansed, mapped, and transformed. Apply a set of functions to extraction data. Correcting data integrity issues.

Validations done during this stage including filtering, character set conversion and encoding handling, conversion of units of measurements, data threshold validation check. Cleaning, splitting and merging columns. Transposing rows and data, and using lookups.

Data is processed to make values and structure consistent across all data. Typical transformations include things like date formatting, resorting rows or columns of data, joining data from two values into one, or splitting data from one value into two. The goal of transformation is to make all data conform to the a uniform schema.

The cleansing and aggregation

There is a lot of overlap between transform and *pre-processing*, the stage at which data is cleaned, unnecessary fields removed, columns encoded, and so on. The difference lies in ease of repetition. Transform would, ideally, be performed once. Operations are typically expensive, operating on vast quantities of data. Pre-processing, using heavily optimised code in NumPy and Pandas, can be iteratively improved; transform less so.

Applies calculations, concatenations.

Uses a series of rules or operations to retrieve pure data from the source and deliver the data in its final form for manipulation at the receiving end. Some data sources need very little, or even no, data processing. Sometimes one or more transformation may be critical to match the business and technical requirements of the target database.

Transform according to business rules. Usually involves operations such as filtering, sorting, aggregating, joining, cleaning, de-duplicating, and validating. Often run in parallel to save time.

The cleansing and aggregation that may need to happen to data to prepare it for analysis. Two approaches.

Shift in recent years toward transforming data within the warehouse rather than transforming it beforehand, primarily driven by two factors: the increased performance and scalability of the modern analytics database, and the ability of in-database transformation to be written in SQL, the data manipulation language of choice for most analysts.

Transformations can be divided into basic and advanced. Basic:

* Cleaning: mapping NULL to 0, “Male” to “M”, formatting dates
* Deduplication: identifying and removing duplicate records
* Format revision: character set conversion, unit of measurement conversion, data/time conversion
* Key restructuring: establishing key relationships across tables .

Advanced

* Derivation: applying business rules to your data that derive new calculated values from existing data
* Filtering: selecting only certain rows and / or columns
* Joining: linking data from multiple sources
* Splitting: splitting a single column into multiple columns
* Data validation: simple or complex data validation
* Summarisation
* Aggregation: aggregating data elements from multiple data sources and databases
* Integration: reconciling different data names and values for the same data element.

Transformation converts the raw data that has been extracted from the source server. It gets cleaned, mapped, and transformed, often to a specific data schema, so it will meet operational needs. Cleaning, format revision, data threshold validation checks, restructuring, de-duplication, filtering, merging, splitting, derivation, summarisation, integration, aggregation and complex data validation.

Data is not usually loaded directly into the target data warehouse, but it is common to have it uploaded into a staging database. This step ensures a quick roll in case something does not go as planned. Possibility to generate audit reports for regulatory compliance or diagnose and repair any data issues.

**Transform for this dissertation**

Unifying data via ID

Correctly identifying SCHEDULEs

Getting destination times for trains to match with schedules

Converting into a tabular format

**Load**

Loading is the simplest stage. It is simply the storing of data in a format accessible for future usage. Sends data to the receiving end, likely to be data storage. May be very simple or very complex, depending on the needs of the application. Some data storage methods may replace old data with cumulative data. Updating of extracted data is normally done on a periodic basis.

Move the transformed data into the permanent, target database. Once loaded, the ETL process is complete. In many organisations ETL is performed regularly in order to keep the data warehouse updated with the latest data.

The process of writing converted data from a staging area to a target database.

Ensure that key field data is neither missing nor null

Test modelling views based on the target tables

Check that combined values and calculated measures

**Load for this dissertation**

**Evaluation**

As previously stated, ETL took a considerable length of time. The majority of the time spent working on this dissertation, in fact.

Modern ETL systems have a great many convenience features – automated rollbacks, in particular – that would have simplified the task somewhat.

A great deal of time was spent running and re-running the same processes with small bugfixes. Even working across different machines was difficult with terabytes of data.

It was also a brand-new idea for me. Machine learning problems I’d tackled in the past had datasets fully prepared already, albeit with some pre-processing required. But ETL is much deeper. I found it very hard, in particular, to reach a stage when I could view my dataset as immutable.

1. Sometimes a different order is used: Extract-Load-Transform (ELT). The sole difference is where the transformation stage takes place. The distinction is largely academic here, so ETL is used throughout. [↑](#footnote-ref-1)