Three Management Layers

1-Operational Management Layer

The operational management layer is the core store for the data science ecosystem's complete processing capability. The layer stores every processing schedule and workflow for the all-inclusive ecosystem. This area enables you to see a singular view of the entire ecosystem. It reports the status of the processing. The operations management layer is the layer where we record the following.

1.1-Processing-Stream Definition and Management:

The processing-stream definitions are the building block of the data science ecosystem. I store all my current active processing scripts in this section. Definition management describes the workflow of the scripts through the system, ensuring that the correct execution order is managed, as per the data scientists' workflow design.

1.2-Parameters

The parameters for the processing are stored in this section, to ensure a single location for all the system parameters.

In my production system, for each customer, we place all these parameters in a single location and then simply call the single location.

Two main designs are used:

1. A simple text file that we then import into every Python script
2. A parameter database supported by a standard parameter setup script that we then include into every script.

I will also admit to having several parameters that follow the same format as the preceding examples, and I simply collect them in a section at the top of the code.

1.3-Scheduling:

The scheduling plan is stored in this section, to enable central control and visibility of the complete scheduling plan for the system. In my solution, I use a Drum-Buffer-Rope (Figure 6-1) methodology. The principle is simple.

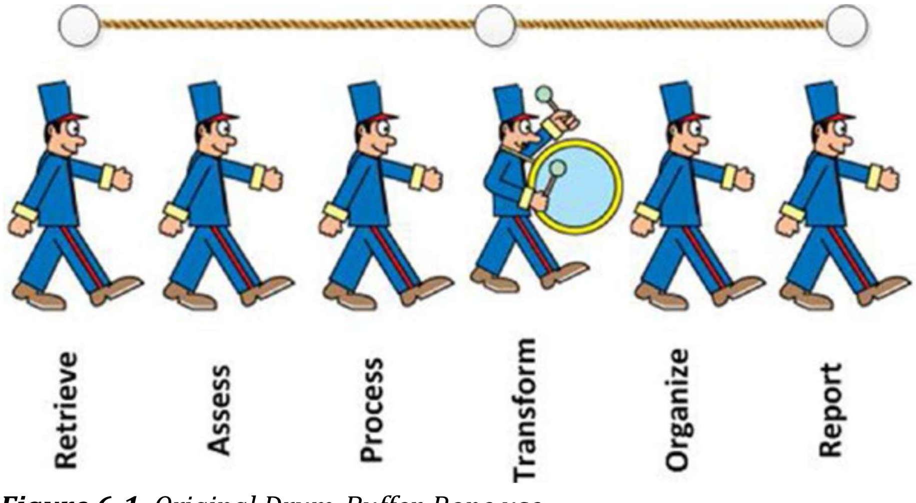

Figure 6-1. *Original Drum-Buffer-Rope use*

Similar to a troop of people marching, the Drum-Buffer-Rope methodology is a standard practice to identify the slowest process and then use this process to pace the complete pipeline. You then tie the rest of the pipeline to this process to control the eco-system's speed.

So, you place the "drum" at the slow part of the pipeline, to give the processing pace, and attach the "rope" to the beginning of the pipeline, and the end by ensuring that no processing is done that is not attached to this drum. This ensures that your processes complete more efficiently, as nothing is entering or leaving the process pipe without been recorded by the drum's beat.

I normally use an independent Python program that employs the directed acyclic graph (DAG) that is provided by the network libraries' DiGraph structure. This automatically resolves duplicate dependencies and enables the use of a topological sort, which ensures that tasks are completed in the order of requirement.

1.4-Monitoring

The central monitoring process is in this section to ensure that there is a single view of the complete system. Always ensure that you monitor your data science from a single point. Having various data science processes running on the same ecosystem without central monitoring is not advised.

e-Communication

All communication from the system is handled in this one section, to ensure that the system can communicate any activities that are happening. We are using a complex communication process via Jira, to ensure we have all our data science tracked.

I have found that the internal communication channel in any company is driven by the communication tools it uses. The only advice I will offer is to *communicate*! You would be alarmed if at least once a week, you lost a project owing to someone not communicating what they are running.

1.5-Alerting

The alerting section uses communications to inform the correct person, at the correct time, about the correct status of the complete system. I use Jira for alerting, and it works well. If any issue is raised, alerting provides complete details of what the status was and the errors it generated.

2-Audit, Balance, and Control Layer

The audit, balance, and control layer controls any processing currently under way. This layer is the engine that ensures that each processing request is completed by the ecosystem as planned. The audit, balance, and control layer is the single area in which you can observe what is currently running within your data scientist environment.

It records

- Process-execution statistics
- Balancing and controls
- Rejects- and error-handling
- Fault codes management

The three subareas are utilized in following manner.

2.1-Audit

An audit is a systematic and independent examination of the ecosystem. The audit sublayer records the processes that are running at any specific point within the environment. This information is used by data scientists and engineers to understand and plan future improvements to the processing.

My experience shows that a good audit trail is extremely crucial. The use of the built-in audit capability of the data science technology stack's components supply you with a rapid and effective base for your auditing. I will discuss what audit statistics are essential to the success of your data science.
In the data science ecosystem, the audit consists of a series of observers that record preapproved processing indicators regarding the ecosystem. I have found the following to be good indicators for audit purposes.

2.1.1-Built-in Logging

I advise you to design your logging into an organized preapproved location, to ensure that you capture every relevant log entry. I also recommend that you do not change the internal or built-in logging process of any of the data science tools, as this will make any future upgrades complex and costly. I suggest that you handle the logs in same manner you would any other data source.

Normally, I build a controlled systematic and independent examination of all the built-in logging vaults. That way, I am sure I can independently collect and process these logs across the ecosystem. I deploy five independent watchers for each logging location, as logging usually has the following five layers.

a-Debug Watcher:
This is the maximum verbose logging level. If I discover any debug logs in my ecosystem, I normally raise an alarm, as this means that the tool is using precise processing cycles to perform low-level debugging.

b-Information Watcher:
The information level is normally utilized to output information that is beneficial to the running and management of a system. I pipe these logs to the central Audit, Balance, and Control data store, using the ecosystem as I would any other data source.

c-Warning Watcher:
Warning is often used for handled "exceptions" or other important log events. Usually this means that the tool handled the issue and took corrective action for recovery. I pipe these logs to the central Audit, Balance, and Control data store, using the ecosystem as I would any other data source. I also add a warning to the Performing a Cause and Effect Analysis System data store.

d-Error Watcher:
Error is used to log all unhandled exceptions in the tool. This is not a good state for the overall processing to be in, as it means that a specific step in the planned processing did not complete as expected. Now, the ecosystem must handle the issue and take corrective action for recovery. I pipe these logs to the central Audit, Balance, and Control data store, using the ecosystem as I would any other data source. I also add an error to the Performing a Cause and Effect Analysis System data store.

e-Fatal Watcher:
Fatal is reserved for special exceptions/conditions for which it is imperative that you quickly identify these events. This is not a good state for the overall processing to be in, as it means a specific step in the planned processing has not completed as expected. This means the ecosystem must now handle the issue and take corrective action for recovery.
Once again, I pipe these logs to the central Audit, Balance, and Control data store, using the ecosystem as I would any other data source. I also add an error to the Performing a Cause and Effect Analysis System data store. I have discovered that by simply using built-in logging and a

good cause-and-effect analysis system, I can handle more than 95% of all issues that arise in the ecosystem.

f-Basic Logging:

This logging enables you to log everything that occurs in your data science processing to a central file, for each run of the process.

2.1.2-Process Tracking:

I normally build a controlled systematic and independent examination of the process for the hardware logging. There is numerous server-based software that monitors temperature sensors, voltage, fan speeds, and load and clock speeds of a computer system. I suggest you go with the tool with which you and your customer are most comfortable. I do, however, advise that you use the logs for your cause-and-effect analysis system.

2.1.3-Data Provenance:

Keep records for every data entity in the data lake, by tracking it through all the transformations in the system. This ensures that you can reproduce the data, if needed, in the future and supplies a detailed history of the data's source in the system.

2.1.4- Data Lineage:

Keep records of every change that happens to the individual data values in the data lake. This enables you to know what the exact value of any data record was in the past. It is normally achieved by a valid-from and valid-to audit entry for each data set in the data science environment.

2.2-Balance:

The balance sublayer ensures that the ecosystem is balanced across the accessible processing capability or has the capability to top up capability during periods of extreme processing. The processing on-demand capability of a cloud ecosystem is highly desirable for this purpose.

By using the audit trail, it is possible to adapt to changing requirements and forecast what you will require to complete the schedule of work you submitted to the ecosystem. I have found that deploying a deep reinforced learning algorithm against the cause-and-effect analysis system can handle any balance requirements dynamically.

2.3-Control:

The control sublayer controls the execution of the current active data science. The control elements are a combination of the control element within the Data Science Technology Stack's individual tools plus a custom interface to control the overarching work.

The control sublayer also ensures that when processing experiences an error, it can try a recovery, as per your requirements, or schedule a clean-up utility to undo the error. The cause-and-effect analysis system is the core data source for the distributed control system in the ecosystem.

I normally use a distributed yoke solution to control the processing. I create an independent process that is created solely to monitor a specific portion of the data processing ecosystem control. So, the control system consists of a series of yokes at each control point that uses Kafka messaging to communicate the control requests. The yoke then converts the requests into a process to execute and manage in the ecosystem.

3- Yoke Solution

The yoke solution is a custom design I have worked on over years of deployments. Apache Kafka is an open source stream processing platform developed to deliver a unified, high-throughput, low-latency platform for handling real-time data feeds.

Kafka provides a publish-subscribe solution that can handle all activity-stream data and processing.

The Kafka environment enables you to send messages between producers and consumers that enable you to transfer control between different parts of your ecosystem while ensuring a stable process. Simple example of the type of information you can send and receive.

3.1- Producer:

The producer is the part of the system that generates the requests for data science processing, by creating structures messages for each type of data science process it requires. The producer is the end point of the pipeline that loads messages into Kafka.

3.2 - Consumer

The consumer is the part of the process that takes in messages and organizes them for processing by the data science tools. The consumer is the end point of the pipeline that offloads the messages from Kafka.

3.3- Directed Acyclic Graph Scheduling

This solution uses a combination of graph theory and publish-subscribe stream data processing to enable scheduling. You can use the Python NetworkX library to resolve any conflicts, by simply formulating the graph into a specific point before or after you send or receive messages via Kafka. That way, you ensure an effective and an efficient processing pipeline.

4- Cause-and-Effect Analysis System

The cause-and-effect analysis system is the part of the ecosystem that collects all the logs, schedules, and other ecosystem-related information and enables data scientists to evaluate the quality of their system.

5-Functional Layer:

The functional layer of the data science ecosystem is the largest and most essential layer for programming and modeling. Any data science project must have processing elements in this layer. The layer performs all the data processing chains for the practical data science.

.

6-Data Science Process:

Following are the five fundamental data science process steps that are the core of my approach to practical data science.

- **Start with a What-if Question:** Decide what you want to know, even if it is only the subset of the data lake you want to use for your data science, which is a good start. For example, let’s consider the example of a small car dealership. Suppose I have been informed that Bob was looking at cars last weekend. Therefore, I ask: “What if I know what car my customer Bob will buy next?”
- **Take a Guess at a Potential Pattern:** Use your experience or insights to guess a pattern you want to discover, to uncover additional insights from the data you already have. For example, I guess Bob will buy a car every three years, and as he currently owns a three-year-old Audi, he will likely buy another Audi. I have no proof; it’s just a guess or so-called gut feeling. Something I could prove via my data science techniques.

- **Gather Observations and Use Them to Produce a Hypothesis:** So, I start collecting car-buying patterns on Bob and formulate a hypothesis about his future behavior. For those of you who have not heard of a hypothesis, it is a proposed explanation, prepared on the basis of limited evidence, as a starting point for further investigation. "I saw Bob looking at cars last weekend in his Audi" then becomes "Bob will buy an Audi next, as his normal three-year buying cycle is approaching."
- **Use Real-World Evidence to Verify the Hypothesis:** Now, we verify our hypothesis with real-world evidence. On our CCTV, I can see that Bob is looking only at Audis and returned to view a yellow Audi R8 five times over last two weeks. On the sales ledger, I see that Bob bought an Audi both three years previous and six previous. Bob's buying pattern, then, is every three years. So, our hypothesis is verified. Bob wants to buy my yellow Audi R8.
- **Collaborate Promptly and Regularly with Customers and Subject Matter Experts As You Gain Insights:** The moment I discover Bob's intentions, I contact the salesperson, and we successfully sell Bob the yellow Audi R8.

7- Functional layer of the framework

It consists of several structures, as follows:

• ***Data schemas and data formats*****:** Functional data schemas and data formats deploy onto the data lake's raw data, to perform the required schema-on-query via the functional layer.

• ***Data models*****:** These form the basis for future processing to enhance the processing capabilities of the data lake, by storing already processed data sources for future use by other processes against the data lake.

• ***Processing algorithms*****:** The functional processing is performed via a series of well-designed algorithms across the processing chain.

• ***Provisioning of infrastructure*****:** The functional infrastructure provision enables the framework to add processing capability to the ecosystem, using technology such as Apache Mesos, which enables the dynamic previsioning of processing work cells.

The processing algorithms and data models are spread across six supersteps for processing the data lake.

1. **Retrieve:** This superstep contains all the processing chains for retrieving data from the raw data lake into a more structured format.
2. **Assess:** This superstep contains all the processing chains for quality assurance and additional data enhancements.
3. **Process:** This superstep contains all the processing chains for building the data vault.
4. **Transform:** This superstep contains all the processing chains for building the data warehouse from the core data vault.
5. **Organize:** This superstep contains all the processing chains for building the data marts from the core data warehouse.
6. **Report:** This superstep contains all the processing chains for building virtualization and reporting of the actionable knowledge.

8- Retrieve Super step

The Retrieve superstep is a practical method for importing completely into the processing ecosystem a data lake consisting of various external data sources. The Retrieve superstep is the first contact between your data science and the source systems.

The successful retrieval of the data is a major stepping-stone to ensuring that you are performing good data science. Data lineage delivers the audit trail of the data elements at the lowest granular level, to ensure full data governance. Data governance supports metadata management for system guidelines, processing strategies, policies formulation, and implementation of processing. Data quality and master data management helps to enrich the data lineage with more business values, if you provide complete data source metadata.

The Retrieve superstep supports the edge of the ecosystem, where your data science makes direct contact with the outside data world. I will recommend a current set of data structures that you can use to handle the deluge of data you will need to process to uncover critical business knowledge.

8.1- Data Lakes:

The Pentaho CTO James Dixon is credited with coining the term *data lake*.

"If you think of a datamart as a store of bottled water—cleansed and packaged and structured for easy consumption—the data lake is a large body of water in a more natural state."

I describe the data lake as follows:

"A company's data lake covers all data that your business is authorized to process, to attain an improved profitability of your business's core accomplishments. The data lake is the complete data world your company interacts with during its business life span. In simple terms, if you generate data or consume data to perform your business tasks, that data is in your company's data lake."

So, as a lake needs rivers and streams to feed it, the data lake will consume an unavoidable deluge of data sources from upstream and deliver it to downstream partners. I had a business customer who received the entire quarterly shipping manifests of his logistics company on a DVD but merely collected them in a plastic container for five years, because he assumed they were superfluous to requirements, as he had the electronic e-mails for each shipment online. Fortunately, one of my interns took the time to load a single DVD and noticed that the complete supply chain routing path and coding procedure were also on the DVD. This new information enriched the data lake by more than 30% in real-value logistics costs, by adding this new data to the company's existing data lake. Do not limit the catchment area of your data lake, as the most lucrative knowledge is usually in the merging of data you never before accepted as important.

8.2-Data Swamps:

Data swamps are simply data lakes that are not managed. They are not to be feared. They need to be tamed.

Following are four critical steps to avoid a data swamp.

a-Start with Concrete Business Questions

Simply dumping a horde of data into a data lake, with no tangible purpose in mind, will result in a big business risk. Countless times, I have caught sight of data-loading agreements that state that they want to load all data.

My recommendation is to perform a comprehensive analysis of the entire set of data you have and then apply a metadata classification for the data, stating full data lineage for allowing it into the data lake. The data lake must be enabled to collect the data required to answer your business questions.

If, for example, you wanted to calculate the shipping patterns for the last five years, you would require all the data for the last five years related to shipping to be available in the data lake. You would also need to start a process of collecting the relevant data in the future, at agreed intervals, to ensure that you could use the same data science in a year's time, to evaluate the previous five years' data. The data lineage ensures that you can reference the data sources once you have your results.

b- Data Governance

The role of data governance, data access, and data security does not go away with the volume of data in the data lake. It simply collects together into a worse problem, if not managed.

Spend the time on data governance up front, as recovery is not easy. I typically spend 70%+ of the project's time on governance and classification of data sources.

c- Data Source Catalog

Metadata that link ingested data-to-data sources are a must-have for any data lake. I suggest you note the following as general rules for the data you process.

- *Unique data catalog number:* I normally use YYYYMMDD/ NNNNNN/NNN. E.g. 20171230/000000001/001 for data first registered into the metadata registers on December 30, 2017, as data source 1 of data type 1. This is a critical requirement.
- *Short description (keep it under 100 characters):* Country codes and country names (Country Codes—ISO 3166)
- *Long description (keep it as complete as possible):* Country codes and country names used by VKHC as standard for country entries
- Contact information for external data source: *ISO 3166-1:2013 code lists from* www.iso.org/iso-3166-country-codes.html
- *Expected frequency:* Irregular (i.e., no fixed frequency, also known as ad hoc). Other options are near-real-time, every 5 seconds, every minute, hourly, daily, weekly, monthly, or yearly.
- *Internal business purpose:* Validate country codes and names.

I have found that if your data source catalog is up to date, the rest of the processing is stress-free.

d- Business Glossary

The business glossary maps the data-source fields and classifies them into respective lines of business. This glossary is a must-have for any good data lake.

Create a data-mapping registry with the following information:

- *Unique data catalog number: I* normally use YYYYMMDD/ NNNNNN/NNN.
- *Unique data mapping number: I* normally use NNNNNNN/ NNNNNNNNN. E.g., 0000001/000000001 for field 1 mapped to internal field 1
- *External data source field name:* States the field as found in the raw data source
- *External data source field type:* Records the full set of the field's data types when loading the data lake

- *Internal data source field name:* Records every internal data field name to use once loaded from the data lake
- *Internal data source field type:* Records the full set of the field's types to use internally once loaded
- *Timestamp of last verification of the data mapping: I* normally use YYYYMMDD-HHMMSS-SSS that supports timestamp down to a thousandth of a second.

The business glossary records the data sources ready for the retrieve processing to load the data. In several of my customer's systems, I have had to perform this classification process, using machine learning with success.

This simply points the classification bots at a data lake and finds any new or changed metadata in that manner. Once newly identified metadata is collected, the human operators have only to deal with the exceptions.

8.3-Analytical Model Usage

Data tagged in respective analytical models define the profile of the data that requires loading and guides the data scientist to what additional processing is required.

Data Field Name Verification
I use this to validate and verify the data field's names in the retrieve processing in an easy manner.

Unique Identifier of Each Data Entry
Allocate a unique identifier within the system that is independent of the given file name. This ensures that the system can handle different files from different paths and keep track of all data entries in an effective manner. Then allocate a unique identifier for each record or data element in the files that are retrieved.

Data Type of Each Data Column
Determine the best data type for each column, to assist you in completing the business glossary, to ensure that you record the correct import processing rules.

Histograms of Each Column
I always generate a histogram across every column, to determine the spread of the data value.

Minimum Value
Determine the minimum value in a specific column.

Maximum Value
Determine the maximum value in a specific column.

Mean
If the column is numeric in nature, determine the average value in a specific column.

Median
Determine the value that splits the data set into two parts in a specific column.

Mode
Determine the value that appears most in a specific column.

Range
For numeric values, you determine the range of the values by taking the maximum value and subtracting the minimum value.

Quartiles

Quartiles are the base values dividing a data set into quarters. Simply sort the data column and split it into four groups that are of four equal parts.

Standard Deviation

the *standard deviation* is a measure of the amount of variation or dispersion of a set of values.

Skewness

Skewness describes the shape or profile of the distribution of the data in the column.

Missing or Unknown Values

Identify if you have missing or unknown values in the data sets.

Data Pattern

I have used the following process for years, to determine a pattern of the data values themselves. Here is my standard version:

Replace all alphabet values with an uppercase case A, all numbers with an uppercase N, and replace any spaces with a lowercase letter *b* and all other unknown characters with a lowercase u. As a result, “Good Book 101” becomes "AAAAbAAAAbNNNu.” This pattern creation is beneficial for designing any specific assess rules.

8.4-Data Quality

More data points do not mean that data quality is less relevant. Data quality can cause the invalidation of a complete data set, if not dealt with correctly.

8.5-Audit and Version Management

Up to now, I have simply allowed you to save and work with data in a use-once method. That is not the correct way to perform data science. You must always report the following:

- Who used the process?
- When was it used?
- Which version of code was used?

8.6-Training the Trainer Model

To prevent a data swamp, it is essential that you train your team also. Data science is a team effort. People, process, and technology are the three cornerstones to ensure that data is curated and protected. You are responsible for your people; share the knowledge you acquire from this book. The process I teach you, you need to teach them. Alone, you cannot achieve success.

Technology requires that you invest time to understand it fully. We are only at the dawn of major developments in the field of data engineering and data science. Remember: A big part of this process is to ensure that business users and data scientists understand the need to start small, have concrete questions in mind, and realize that there is work to do with all data to achieve success.

8.7- Actionable Business Knowledge from Data Lakes

I will guide you through several actionable business processes that you can formulate directly from the data in the sample data set.

Engineering a Practical Retrieve Superstep

I have explained the various aspects of the Retrieve superstep, I will explain how to assist our company with its processing.

The means are as follows:

- Identify the data sources required.

- Identify source data format (CSV, XML, JSON, or database).
- Data profile the data distribution (Skew, Histogram, Min, Max).
- Identify any loading characteristics (Columns Names, Data Types,
    - Volumes).

- Determine the delivery format (CSV, XML, JSON, or database).

Vermeulen PLC

The company has two main jobs on which to focus your attention:

- **Designing a Routing Diagram for the Company:** A network routing diagram is a map of all potential routes through the company's network, like a road map.
- **Building a Diagram for the Scheduling of Jobs:** You can extract core routers locations to schedule maintenance jobs. Now that we know where the routers are, we must set up a schedule for maintenance jobs that have to be completed every month. To accomplish this, we must retrieve the location of each router, to prepare a schedule for the staff maintaining them.

Hillman Ltd

The company has four main jobs requiring your attention:

- *Planning the locations of the warehouses:* Hillman has countless UK warehouses, but owing to financial hardships, the business wants to shrink the quantity of warehouses by 20%.
- *Planning the shipping rules for best-fit international logistics:* At Hillman Global Logistics' expense, the company has shipped goods from its international warehouses to its UK shops. This model is no longer sustainable. The co-owned shops now want more feasibility regarding shipping options.
- *Adopting the best packing option for shipping in containers:* Hillman has introduced a new three-size-shipping-container solution. It needs a packing solution encompassing the warehouses, shops, and customers.
- *Creating a delivery route:* Hillman needs to preplan a delivery route for each of its warehouses to shops, to realize a 30% savings in shipping costs.

Plan the locations of the warehouses. I will assist you in retrieving the warehouse locations.

Important Shipping Information

Shipping Terms

These determine the rules of the shipment, the conditions under which it is made. Normally, these are stated on the shipping manifest. Currently, Hillman is shipping everything as DDP.

Seller

The person/company sending the products on the shipping manifest is the *seller.* In our case, there will be warehouses, shops, and customers. Note that this is not a location but a legal entity sending the products.

Carrier

The person/company that physically carries the products on the shipping manifest is the *carrier.* Note that this is not a location but a legal entity transporting the products.

Port

A *port* is any point from which you have to exit or enter a country. Normally, these are shipping ports or airports but can also include border crossings via road. Note that there are two ports in the complete process. This is important. There is a port of exit and a port of entry.

Ship

Ship is the general term for the physical transport method used for the goods. This can refer to a cargo ship, airplane, truck, or even person, but it must be identified by a unique allocation number.

Terminal

A *terminal* is the physical point at which the goods are handed off for the next phase of the physical shipping.

Named Place

This is the location where the ownership is legally changed from seller to buyer. This is a specific location in the overall process. Remember this point, as it causes many legal disputes in the logistics industry.

Buyer

The person/company receiving the products on the shipping manifest is the *buyer.* In our case, there will be warehouses, shops, and customers. Note that this is not a location but a legal entity receiving the products.

EXW—Ex Works (Named Place of Delivery)

By this term, the seller makes the goods available at its premises or at another named place. This term places the maximum obligation on the buyer and minimum obligations on the seller.

Here is the data science version: If I were to buy *Practical Data Science* at a local bookshop and take it home, and the shop has shipped it EXW—Ex Works, the moment I pay at the register, the ownership is transferred to me. If anything happens to the book, I would have to pay to replace it.

FCA—Free Carrier (Named Place of Delivery)

Under this condition, the seller delivers the goods, cleared for export, at a named place. Following is the data science version.

If I were to buy *Practical Data Science* at an overseas duty-free shop and then pick it up at the duty-free desk before taking it home, and the shop has shipped it FCA— Free Carrier—to the duty-free desk, the moment I pay at the register, the ownership is transferred to me, but if anything happens to the book between the shop and the dutyfree desk, the shop will have to pay. It is only once I pick it up at the desk that I will have to pay, if anything happens. So, the moment I take the book, the transaction becomes EXW, so I have to pay any necessary import duties on arrival in my home country.

CPT—Carriage Paid To (Named Place of Destination)

The seller, under this term, pays for the carriage of the goods up to the named place of destination. However, the goods are considered to be delivered when they have been handed over to the first carrier, so that the risk transfers to the buyer upon handing the goods over to the carrier at the place of shipment in the country of export.

The seller is responsible for origin costs, including export clearance and freight costs for carriage to the named place of destination. (This is either the final destination, such as the buyer's facilities, or a port of destination. This must be agreed upon by both seller and buyer, however.)

Now, here is the data science version: If I were to buy *Practical Data Science* at an overseas bookshop and then pick it up at the export desk before taking it home and the shop shipped it CPT—Carriage Paid To—the duty desk for free, the moment I pay at the register, the ownership is transferred to me, but if anything happens to the book between the shop and the duty desk of the shop, I will have to pay. It is only once I have picked up the book at the desk that I have to pay if anything happens. So, the moment I take the book, the transaction becomes EXW, so I must pay any required export and import duties on arrival in my home country.

CIP—Carriage and Insurance Paid To (Named Place of Destination)

This term is generally similar to the preceding CPT, with the exception that the seller is required to obtain insurance for the goods while in transit. Following is the data science version.

If I buy *Practical Data Science* in an overseas bookshop and then pick it up at the export desk before taking it home, and the shop has shipped it CPT—Carriage Paid To— to the duty desk for free, the moment I pay at the register, the ownership is transferred to me. However, if anything happens to the book between the shop and the duty desk at the shop, I have to take out insurance to pay for the damage. It is only once I have picked it up at the desk that I have to pay if anything happens. So, the moment I take the book, it becomes EXW, so I have to pay any export and import duties on arrival in my home country. Note that insurance only covers that portion of the transaction between the shop and duty desk.

DAT—Delivered at Terminal (Named Terminal at Port or Place of Destination)

This Incoterm requires that the seller deliver the goods, unloaded, at the named terminal. The seller covers all the costs of transport (export fees, carriage, unloading from the main carrier at destination port, and destination port charges) and assumes all risks until arrival at the destination port or terminal.

The terminal can be a port, airport, or inland freight interchange, but it must be a facility with the capability to receive the shipment. If the seller is not able to organize unloading, it should consider shipping under DAP terms instead. All charges after unloading (for example, import duty, taxes, customs and on-carriage costs) are to be borne by buyer.

Following is the data science version. If I were to buy *Practical Data Science* overseas and then pick it up at a local bookshop before taking it home, and the overseas shop shipped it—Delivered At Terminal (Local Shop)—the moment I pay at the register, the ownership is transferred to me. However, if anything happens to the book between the payment and the pickup, the local shop pays. It is picked up only once at the local shop. I have to pay if anything happens. So, the moment I take it, the transaction becomes EXW, so I have to pay any import duties on arrival in my home.

DAP—Delivered at Place (Named Place of Destination)

According to Incoterm 2010's definition, DAP—Delivered at Place—means that, at the disposal of the buyer, the seller delivers when the goods are placed on the arriving means of transport, ready for unloading at the named place of destination. Under DAP terms, the risk passes from seller to buyer from the point of destination mentioned in the contract of delivery.

Once goods are ready for shipment, the necessary packing is carried out by the seller at his own cost, so that the goods reach their final destination safely. All necessary legal formalities in the exporting country are completed by the seller at his own cost and risk to clear the goods for export.

After arrival of the goods in the country of destination, the customs clearance in the importing country must be completed by the buyer at his own cost and risk, including all customs duties and taxes. However, as with DAT terms, any delay or demurrage charges are to be borne by the seller.

Under DAP terms, all carriage expenses with any terminal expenses are paid by seller, up to the agreed destination point. The required unloading cost at the final destination has to be accepted by the buyer under DAP terms.

Here is the data science version. If I were to buy 100 copies of *Practical Data Science* from an overseas web site and then pick up the copies at a local bookshop before taking them home, and the shop shipped the copies DAP-Delivered At Place (Local Shop)— the moment I paid at the register, the ownership would be transferred to me. However, if anything happened to the books between the payment and the pickup, the web site owner pays. Once the copies are picked up at the local shop, I have to pay to unpack them at bookshop. So, the moment I take the copies, the transaction becomes EXW, so I will have to pay costs after I take the copies.

DDP—Delivered Duty Paid (Named Place of Destination)

By this term, the seller is responsible for delivering the goods to the named place in the country of the buyer and pays all costs in bringing the goods to the destination, including import duties and taxes. The seller is not responsible for unloading. This term places the maximum obligations on the seller and minimum obligations on the buyer. No risk or responsibility is transferred to the buyer until delivery of the goods at the named place of destination.

The most important consideration regarding DDP terms is that the seller is responsible for clearing the goods through customs in the buyer's country, including both paying the duties and taxes, and obtaining the necessary authorizations and registrations from the authorities in that country.

Here is the data science version. If I were to buy 100 copies of *Practical Data Science* on an overseas web site and then pick them up at a local bookshop before taking them home, and the shop shipped DDP—Delivered Duty Paid (my home)—the moment I pay at the till, the ownership is transferred to me. However, if anything were to happen to the books between the payment and the delivery at my house, the bookshop must replace the books as the term covers the delivery to my house.

Connecting to Other Data Sources

The following are a few common data stores

SQLite

This requires a package named sqlite3.

Microsoft SQL Server

Microsoft SQL server is common in companies, and this connector supports your connection to the database. Via the direct connection, use

```
from sqlalchemy import create_engine
engine = create_engine('mssql+pymssql://scott:tiger@hostname:port/vermeulen')
```

Oracle

Oracle is a common database storage option in bigger companies. It enables you to load data from the following data source with ease:

```
from sqlalchemy import create_engine
engine = create_engine('oracle://andre:vermeulen@127.0.0.1:1521/vermeulen')
```

MySQL

MySQL is widely used by lots of companies for storing data. This opens that data to your data science with the change of a simple connection string.
There are two options. For direct connect to the database, use

```
from sqlalchemy import create_engine
engine = create_engine('mysql+mysqldb://scott:tiger@localhost/vermeulen')
```

Apache Cassandra

Cassandra is becoming a widely distributed database engine in the corporate world.
To access it, use the Python package cassandra.

```
from cassandra.cluster import Cluster
cluster = Cluster()
session = cluster.connect(‘vermeulen’)
```

Apache Hadoop

Hadoop is one of the most successful data lake ecosystems in highly distributed data Science. The pydoop package includes a Python MapReduce and HDFS API for Hadoop.

Pydoop

is a Python interface to Hadoop that allows you to write MapReduce applications and interact with HDFS in pure Python