EXPLORIUM

Making Sense of Data Prep:

ETL, Wrangling, Data Enrichment

Table of contents

Getting your Data Ready for ML — Data Preparation

Data preparation is an essential, if sometimes overlooked, part of any machine learning (ML) lifecycle. It's not that data scientists ignore it, but it's easy to think that sorting data into a database and running a few Python functions will do the trick. You may be right if you're working with a small dataset, or if your models are simply an academic exercise, but what if you're dealing with production-ready models or datasets that have hundreds of columns and thousands of rows?

Let's put it another way. Imagine you're cooking a meal, and you've gone through the trouble of raiding your pantry and going to the store to get all the ingredients you need. Do you simply toss everything into a pot and hope for the best? Probably not, but let's even take it a step further. Maybe you even peel some of the vegetables and take things out of their packaging. Is that enough? Possibly.

But what if instead of simply slicing a few things up and tossing it all in together, you take the time to prepare it the right way, cutting ingredients uniformly and adding just the right amount? You'll probably end up with a great meal. This is the core of data preparation. Before you get great insights from your models, you need to make sure your data is ready to deliver the goods. Let's dive deeper into how you can prepare your data for maximum efficiency.

In this whitepaper, we'll break down what you need to do to prepare your datasets for the best results in machine learning. We'll discuss the ETL process in-depth, as well as the concept of data wrangling, and the challenges you might face at each turn. We'll also discuss some ways you can speed up the process.

Getting your data ready for machine learning

External data can greatly enrich your internal datasets and provide answers you simply couldn't get on your own. At the same time, it's important to appreciate that onboarding external data is a hefty task in its own right. You don't simply purchase or acquire external data and that's the end of the matter. You still need to integrate it, clean it, and make sure it's relevant.

> "Fully 80 percent of credit unions believe the inaccuracies have affected their bottom line, causing an average 13 percent hit on revenue. Additionally, 70 percent of financial institutions blame poor data quality for ongoing problems with their loyalty efforts"
>
> - Deloitte Research

Cleaning your data

You need to clean up and prepare all your data to make sure it's properly organized, free from errors and omissions, and ready for use by your models. This is especially important when you're using external datasets, which may use different formatting conventions or be incompatible in other ways with your existing data.

These are just some of the issues that could impact your data integration process when looking at an external dataset:

- It’s not labeled properly
- It contains null or empty values
- Columns are mistitled
- The dataset is unstructured

The ETL process

ETL stands for extract-transform-load and refers to the process of migrating, or drawing out and copying, data from one source to another. That could be because you are moving data from one database to another in a different format. It could also be because you are dealing with high volumes of data, originating from multiple source systems, including data marts or data warehouses, all of which need to be consolidated.

Here's what each of the ETL functions does in a little more detail:

Extract

This is when data is read (collected) from the database or a range of sources. The extraction process sounds simple, but how you extract your data will also be important in how it interacts with your models later on.

> The extraction process sounds simple, but how you extract your data will also be important in how it interacts with your models later on.

The first step in the extraction process is to understand your data sources, and how they'll impact the speed and latency of your data transfer. You can achieve this during your data auditing process (you can read more about data auditing in part one of our series), by understanding how the data is stored at the source, and how it will interact with your internal data structures.

This will also give you an idea of how to handle the loading process later on. Data that is being constantly loaded and transferred might create bottlenecks if you don't use the right extraction process, and could impact your production models' efficiency down the line.

Once you're ready to start extracting data, you can do so in full (as a single extraction), or incrementally (partial extraction), though the latter means you'll need a mechanism to update you on any changes made to the data source.

It’s also worth noting that while extraction seems like a straightforward process — simply taking data from somewhere else and putting it into your dataset — there are some challenges that could impact the process. Some of these include:

- Transferring data from an unstructured data source (such as a data lake)
- Changing data formats over time
- Shifts in data velocity and volume
- Null values in your data sources

These complications mean that to ensure your extraction goes smoothly, you need to have data cleansing processes in place. These steps include:

- Detecting errors and inconsistencies in your data source (or sources)
- Correcting mismatches and ensuring columns are organized in the same sequence in your source and destination datasets
- Ensuring your data columns use the same format (for instance, if you’re using time series data or currencies, you should make sure values are written in the same style)

Once your data is ready, you'll likely extract it to a staging area before you place it in your destination to give yourself a chance to transform it without impacting your overall dataset. Now, it's ready for you to shape it into a better form.

Transform

Once you load your data into a staging area, you'll still be left with a lot of data points, and even whole datasets, that aren't ready or usable in machine learning models. After you run a basic cleanup of your data during the extraction process, you'll still be left with a large mass of data that needs to be turned into something you can use if you want to get the most out of it.

> After you run a basic cleanup of your data during the extraction process, you'll still be left with a large mass of data that needs to be turned into something you can use if you want to get the most out of it.

There are essentially two types of data transformations we use — the basic kind, which is a lot like cleaning data, and the advanced kind, which lets you actually **transform** your datasets.

Basic transformation looks a lot like cleaning your data, but it happens in the staging area once your new datasets have been loaded and you can see what you're working with. Some common basic transformations include:

- Format standardization to ensure every data type is uniform within its columns and rows
- Cleaning the datasets, which include proper mapping of values to standardized values (such as converting any null values to "0", or from "Male" and "Female" to "M" or "F", for example)
- Removing duplicate values and columns to avoid problems during training and testing
- Establishing key relationships across your data tables

Advanced transformation methods include:

- **Field decoding** for data that comes from multiple sources. Often, different data sources use their own encryption, validation, and data representation methods which make it hard to understand when you're dealing with unique values or duplicates, or even useless data. You'll need to convert everything into a uniform, understandable format.

- **Data merging** to avoid redundant columns and bloated datasets. If you have data fields that are similar enough, you can usually merge them into a single entity for clarity and ease of use later on. For example, if you have multiple fields related to your product (price, name, type, description, etc.), you can create a unified field to describe them.
- **Separating single** fields into multiple ones is also valuable when you're dealing with columns that are too unwieldy. For instance, you can take a single name and split it into first, middle, and last name, or do the same thing with company data that can give you more value on its own than in a single monolithic data point.

Once you have a dataset that you're happy with, and that meets your ML needs and criteria, you can load it into your actual database.

Load

This is the process used to “write” the data into the target database (or wherever it will be fed into your machine learning models). If you’ve managed to complete the previous two steps well, this last part will be a lot easier, though there are still some considerations to keep in mind.

You can either load the data into your warehouse or database incrementally or in a full set all at once. The latter will take more time (especially as your datasets increase in size) but can be a more effective way if you need to have immediate access to the full set.

On the other hand, incrementally loading your data lets you constantly apply changes as your datasets change without having to erase and refresh the entire dataset, a more convenient approach if you’re dealing with real-time data, or datasets that are updated frequently.

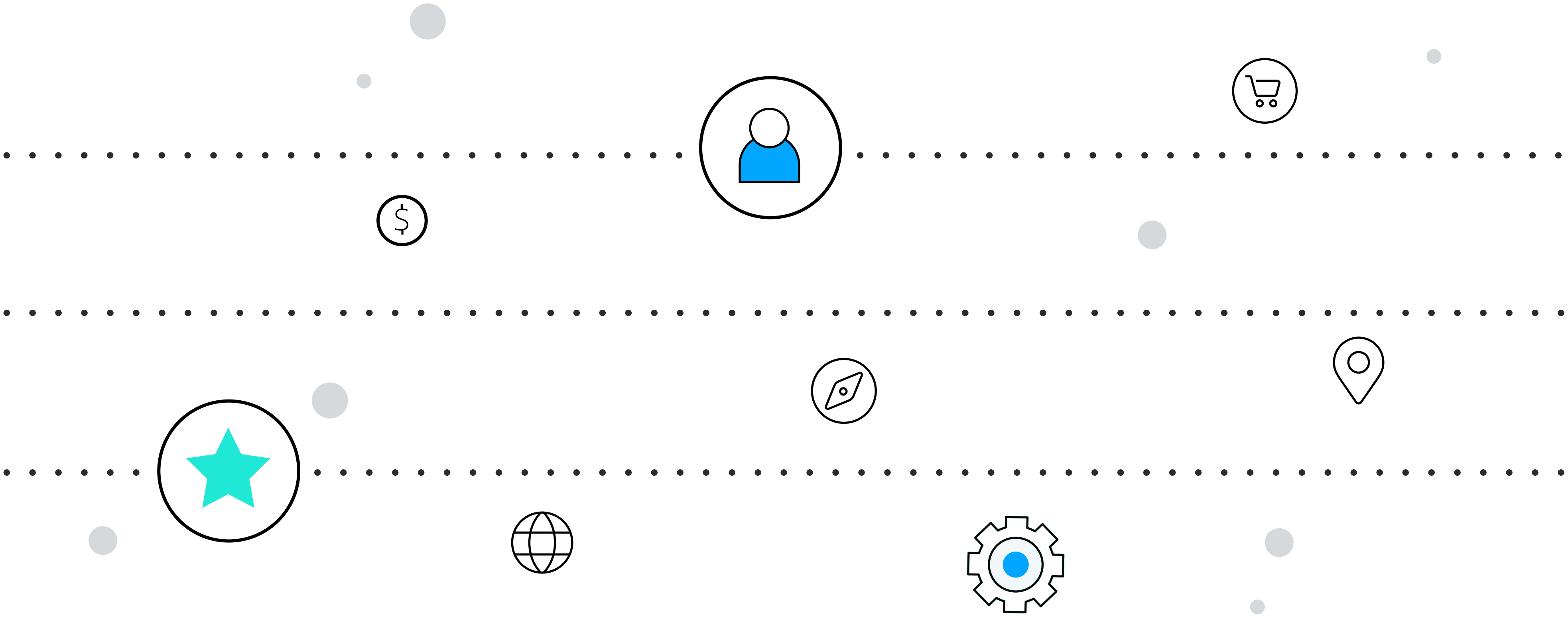
Finally, before declaring victory and moving on, you should always have some load verification mechanisms in place for a few reasons:

- Ensure that there aren’t any null or missing values in your data fields
- Make sure any combined values and calculated measures you included

are accurate

- Verify that all columns and rows are properly loaded and labeled.

Now, your data should be ready to use.



Data wrangling

Data wrangling, or data “munging” refers to all the things you need to do to your raw or unformatted data in order to map it into a format that’s easily usable for machine learning.

This can be an enormous, difficult process. Some data scientists say they spend up to three-quarters of their time on data wrangling! However, you can’t skip over it; you need to get to grips with your data before you can figure out how best to analyze it. Done properly, data wrangling reveals vital insights about what your data contains and how you might manipulate it in your machine learning project while helping you figure out what features to extract (more on that in the next section).

> Some data scientists say they spend up to three-quarters of their time on data wrangling! However, you can’t skip over it; you need to get to grips with your data before you can figure out how best to analyze it.

1. Exploring the dataset

This involves investigating the dataset to make sure it contains relevant information, and you will be able to extract value from it. That sounds simple, but it can be a complex, exhausting task, especially when you veer into the realm of exploratory data analysis (EDA).

The important thing there is to get to know the patterns and correlations in the dataset. Since these inevitably depend on the wider business or sector context, you will really need to work with a domain expert (or have significant domain expertise themselves) to do this effectively.

One important element of exploration is correlation analysis, which is fundamental to the success of your future models. This involves expunging redundant information, such as related columns containing similar, identical, or unnecessary information. In simple terms, this can serve to muddle your models later on. For example, a variable importance metric will see both inputs as important but won't have a clear way to prioritize one over the other. If you don't fix this now, your models may become confusing or overly complicated.
Structuring the data

2. Structuring the data

Data comes in all shapes and sizes. Before you start modeling, you need to get it into the right one for machine learning, getting your dataset into a table format that works for what you need to do with it later

This involves merging, ordering, and reshaping data ready for your models, including building the right columns and organizing the ontologies (labels and categories) correctly. Don't be afraid to move data around in ways that facilitate easier analysis and computation. You may find that it makes sense to split a single column into a number of rows, for example.

Cleaning your data

As we've seen, messy data is a disaster that will drag down your models' accuracy. You need to go through each dataset carefully to standardize your entries, eliminating overlaps, outliers, redundant data points, empty columns, and null values.

Pay close attention to date values in particular. Conventions around recording dates vary from county to country, organization to organization. For example, the U.S., Canada, and the Philippines list dates in MM-DD-YY format, while almost all other nations put the day first, followed by the month. Even within a single country, different systems may prefer to use DD-MM-YYYY over DD-MM-YY, or to list the name of the month rather than a numerical value, and so on. Small issues like these grow into much bigger ones if you don't fix inconsistencies in your datasets before building your models.

Enrichment

This is where you go back to your internal datasets and figure out how to incorporate additional, external data to augment it. You add relevant data points and columns from potentially hundreds of other sources, creating a better, more comprehensive dataset.

At this stage, you should figure out what other kinds of data you can derive from the data you already have. For example, If you have a column with average monthly earnings, might it be useful to use this to calculate average yearly earnings, too, and add a column for that?

If you're using a data-as-a-service provider, or a platform that automates your external connections, this is when you would plug in all these pre-vetted external datasets to really flesh out the bare-bones information with valuable, relevant industry data from a range of other datasets.

Typically this is achieved using one of the following matching techniques:

- **Join.** This is when you are searching for an exact query, looking for an exact match. For example, you might search a company name in a local database, or put out an API query with that exact term. When the system finds an entry for that company in the database, it adds all the other information (e.g. contact details, turnover, company size, etc) into your database, too.
- **Partial matching.** This is a little less granular than joining. Rather than seeking an exact match for your search term, you look for partial matches. So, rather than a company name, you might search for information on companies working in a particular sector. This then allows you to extract relevant features, based on the partial match.

- **Regional matching.** This is when you connect a search term, in its broad sense, to a region. You aren't looking for an exact query and key to match with — rather, you're looking for useful data on a particular topic or type of business, that applies in a geographical region or a date range. That could be, for example, stock market data from June-July 2020. Or it could be traffic flows in New York. The skill is getting just the right level of granularity, so that the results you get are accurate and relevant, without being too broad.
- **Text matching.** This allows you to do things like web scraping, where you look for all instances of a term across websites, social media, blogs, and so on. You don't necessarily have to look for an exact match; you might look for different versions or spellings of the same word or term, for example. Here you're looking for similarities, with your results rates from exact matches to more contextual uses.

Be patient

Data wrangling is an extremely time-intensive process. If you're doing this manually, you will have to corroborate every source and examine every tool in turn.

Even if or when you've found a dataset that works well with your existing data, you can't expect to simply plug in an external data source and let it run without any intervention on your part. If you try to do that, your models will struggle to make sense of the mass of data coming into them and you'll end up with some bad results and dodgy predictions.

> "The reality is that as much as 80 percent of the work on which data scientists spend their time can be fully or partially automated"
>
> - Deloitte Research

Speeding up the process

While there's no way to get out of data wrangling, you can make the process faster and slicker by introducing automation to take care or repeat certain tasks across large datasets. Even automating one small, repetitive data cleaning task turns into a huge time saving when you apply it to a dataset with tens of thousands of rows. The key is, always, to find ways to make your data pipeline scalable — and automation tools are an excellent place to start.

Now that you've figured out exactly what data you have (and more importantly, what you're missing), you can begin the painstaking process of filling in the gaps by acquiring and integrating external data.

Getting your data ready for heavy lifting

You may be in a rush to get on with the business of actually building machine learning models, but if your data isn't in the right state, you may be doing a lot of work for very small returns. Data preparation is a vital step in the ML lifecycle and is essential to make sure that the rest of the steps in your data workflow function smoothly. Without it, you're just throwing ingredients into a pot and hoping for the best.

Now, it's time to see what you can do with a well prepared and structured data set. It's time to move on to the heavy lifting of data science. Keep on reading to chapter three, **Making Sense of Deployment: Feature Engineering, Training, Testing, and Monitoring**, to learn what you can do with a prepared dataset to get the most out of your ML, and to push your models into production with confidence.

Ready to move on to part three? Download Making Sense of Deployment: Feature Engineering, Training, Testing, and Monitoring now.

EXPLORIUM

About Explorium

Explorium offers a first of its kind data science platform powered by augmented data discovery and feature engineering. By automatically connecting to thousands of external data sources and leveraging machine learning to distill the most impactful signals, the Explorium platform empowers data scientists and business leaders to drive decision-making by eliminating the barrier to acquire the right data and enabling superior predictive power.

For more information,
visit www.explorium.ai