Chapter 2

Why Data Preparation is So Important

On a predictive modeling project, machine learning algorithms learn a mapping from input variables to a target variable. The most common form of predictive modeling project involves so-called structured data or tabular data. This is data as it looks in a spreadsheet or a matrix, with rows of examples and columns of features for each example. We cannot fit and evaluate machine learning algorithms on raw data; instead, we must transform the data to meet the requirements of individual machine learning algorithms. More than that, we must choose a representation for the data that best exposes the unknown underlying structure of the prediction problem to the learning algorithms in order to get the best performance given our available resources on a predictive modeling project.

Given that we have standard implementations of highly parameterized machine learning algorithms in open source libraries, fitting models has become routine. As such, the most challenging part of each predictive modeling project is how to prepare the one thing that is unique to the project: the data used for modeling. In this tutorial, you will discover the importance of data preparation for each machine learning project. After completing this tutorial, you will know:

- Structured data in machine learning consists of rows and columns.
- Data preparation is a required step in each machine learning project.
- The routineness of machine learning algorithms means the majority of effort on each project is spent on data preparation.

Let's get started.

2.1 Tutorial Overview

This tutorial is divided into three parts; they are:

1. What Is Data in Machine Learning
2. Raw Data Must Be Prepared
3. Predictive Modeling Is Mostly Data Preparation

2.2 What Is Data in Machine Learning

Predictive modeling projects involve learning from data. Data refers to examples or cases from the domain that characterize the problem you want to solve. In supervised learning, data is composed of examples where each example has an input element that will be provided to a model and an output or target element that the model is expected to predict.

> What we call data are observations of real-world phenomena. [...] Each piece of data provides a small window into a limited aspect of reality.
>
> — Page 1, *Feature Engineering for Machine Learning*, 2018.

Classification is an example of a supervised learning problem where the target is a label, and regression is an example of a supervised learning problem where the target is a number. The input data may have many forms, such as an image, time series, text, video, and so on. The most common type of input data is typically referred to as tabular data or structured data. This is data as you might see it in a spreadsheet, in a database, or in a comma separated variable (CSV) file. This is the type of data that we will focus on.

Think of a large table of data. In linear algebra, we refer to this table of data as a matrix. The table is composed of rows and columns. A row represents one example from the problem domain, and may be referred to as an *example*, an *instance*, or a *case*. A column represents the properties observed about the example and may be referred to as a *variable*, a *feature*, or a *attribute*.

- **Row**. A single example from the domain, often called an instance, example or sample in machine learning.
- **Column**. A single property recorded for each example, often called a variable, predictor, or feature in machine learning.

For example, the columns used for input to the model are referred to as input variables, and the column that contains the target to be predicted is referred to as the output variable. The rows used to train a model are referred to as the training dataset and the rows used to evaluate the model are referred to as the test dataset.

- **Input Variables**: Columns in the dataset provided to a model in order to make a prediction.
- **Output Variable**: Column in the dataset to be predicted by a model.

When you collect your data, you may have to transform it so it forms one large table. For example, if you have your data in a relational database, it is common to represent entities in separate tables in what is referred to as a *normal form* so that redundancy is minimized. In order to create one large table with one row per *subject* or *entity* that you want to model, you may need to reverse this process and introduce redundancy in the data in a process referred to as denormalization.

If your data is in a spreadsheet or database, it is standard practice to extract and save the data in CSV format. This is a standard representation that is portable, well understood, and ready for the predictive modeling process with no external dependencies. Now that we are familiar with structured data, let's look at why we need to prepare the data before we can use it in a model.

2.3 Raw Data Must Be Prepared

Data collected from your domain is referred to as raw data and is collected in the context of a problem you want to solve. This means you must first define what you want to predict, then gather the data that you think will help you best make the predictions. This data collection exercise often requires a domain expert and may require many iterations of collecting more data, both in terms of new rows of data once they become available and new columns once identified as likely relevant to making a prediction.

- **Raw data**: Data in the form provided from the domain.

In almost all cases, raw data will need to be changed before you can use it as the basis for modeling with machine learning.

> A feature is a numeric representation of an aspect of raw data. Features sit between data and models in the machine learning pipeline. Feature engineering is the act of extracting features from raw data and transforming them into formats that are suitable for the machine learning model.
>
> — Page vii, *Feature Engineering for Machine Learning*, 2018.

The cases with no data preparation are so rare or so trivial that it is practically a rule to prepare raw data in every machine learning project. There are three main reasons why you must prepare raw data in a machine learning project. Let's take a look at each in turn.

2.3.1 Machine Learning Algorithms Expect Numbers

Even though your data is represented in one large table of rows and columns, the variables in the table may have different data types. Some variables may be numeric, such as integers, floating-point values, ranks, rates, percentages, and so on. Other variables may be names, categories, or labels represented with characters or words, and some may be binary, represented with 0 and 1 or True and False. The problem is, machine learning algorithms at their core operate on numeric data. They take numbers as input and predict a number as output. All data is seen as vectors and matrices, using the terminology from linear algebra.

As such, raw data must be changed prior to training, evaluating, and using machine learning models. Sometimes the changes to the data can be managed internally by the machine learning algorithm; most commonly, this must be handled by the machine learning practitioner prior to modeling in what is commonly referred to as *data preparation* or *data pre-processing*.

2.3.2 Machine Learning Algorithms Have Requirements

Even if your raw data contains only numbers, some data preparation is likely required. There are many different machine learning algorithms to choose from for a given predictive modeling project. We cannot know which algorithm will be appropriate, let alone the most appropriate for our task. Therefore, it is a good practice to evaluate a suite of different candidate algorithms systematically and discover what works well or best on our data. The problem is, each algorithm has specific requirements or expectations with regard to the data.

> ... data preparation can make or break a model's predictive ability. Different models have different sensitivities to the type of predictors in the model; how the predictors enter the model is also important.
>
> — Page 27, *Applied Predictive Modeling*, 2013.

For example, some algorithms assume each input variable, and perhaps the target variable, to have a specific probability distribution. This is often the case for linear machine learning models that expect each numeric input variable to have a Gaussian probability distribution. This means that if you have input variables that are not Gaussian or nearly Gaussian, you might need to change them so that they are Gaussian or more Gaussian. Alternatively, it may encourage you to reconfigure the algorithm to have a different expectation on the data.

Some algorithms are known to perform worse if there are input variables that are irrelevant or redundant to the target variable. There are also algorithms that are negatively impacted if two or more input variables are highly correlated. In these cases, irrelevant or highly correlated variables may need to be identified and removed, or alternate algorithms may need to be used. There are also algorithms that have very few requirements about the probability distribution of input variables or the presence of redundancies, but in turn, may require many more examples (rows) in order to learn how to make good predictions.

> The need for data pre-processing is determined by the type of model being used. Some procedures, such as tree-based models, are notably insensitive to the characteristics of the predictor data. Others, like linear regression, are not.
>
> — Page 27, *Applied Predictive Modeling*, 2013.

As such, there is an interplay between the data and the choice of algorithms. Primarily, the algorithms impose expectations on the data, and adherence to these expectations requires the data to be appropriately prepared. Conversely, the form of the data may provide insight into those algorithms that are more likely to be effective.

2.3.3 Model Performance Depends on Data

Even if you prepare your data to meet the expectations of each model, you may not get the best performance. Often, the performance of machine learning algorithms that have strong expectations degrades gracefully to the degree that the expectation is violated. Further, it is common for an algorithm to perform well or better than other methods, even when its expectations have been ignored or completely violated. It is a common enough situation that this must be factored into the preparation and evaluation of machine learning algorithms.

> The idea that there are different ways to represent predictors in a model, and that some of these representations are better than others, leads to the idea of feature engineering — the process of creating representations of data that increase the effectiveness of a model.
>
> — Page 3, *Feature Engineering and Selection*, 2019.

The performance of a machine learning algorithm is only as good as the data used to train it. This is often summarized as **garbage in, garbage out**. Garbage is harsh, but it could mean a *weak representation* of the problem that insufficiently captures the dynamics required to learn how to map examples of inputs to outputs.

Let's take for granted that we have *sufficient* data to capture the relationship between input and output variables. It's a slippery and domain-specific principle, and in practice, we have the data that we have, and our job is to do the best we can with that data. A dataset may be a *weak representation* of the problem we are trying to solve for many reasons, although there are two main classes of reason. It may be because complex nonlinear relationships are compressed in the raw data that can be unpacked using data preparation techniques. It may also be because the data is not perfect, ranging from mild random fluctuations in the observations, referred to as a statistical noise, to errors that result in out-of-range values and conflicting data.

- **Complex Data**: Raw data contains compressed complex nonlinear relationships that may need to be exposed
- **Messy Data**: Raw data contains statistical noise, errors, missing values, and conflicting examples.

We can think about getting the most out of our predictive modeling project in two ways: focus on the model and focus on the data. We could minimally prepare the raw data and begin modeling. This puts full onus on the model to tease out the relationships in the data and learn the mapping function from inputs to outputs as best it can. This may be a reasonable path through a project and may require a large dataset and a flexible and powerful machine learning algorithm with few expectations, such as random forest or gradient boosting.

Alternately, we could push the onus back onto the data and the data preparation process. This requires that each row of data best expresses the information content of the data for modeling. Just like denormalization of data in a relational database to rows and columns, data preparation can denormalize the complex structure inherent in each single observation. This is also a reasonable path. It may require more knowledge of the data than is available but allows good or even best modeling performance to be achieved almost irrespective of the machine learning algorithm used.

Often a balance between these approaches is pursued on any given project. That is both exploring powerful and flexible machine learning algorithms and using data preparation to best expose the structure of the data to the learning algorithms. This is all to say, data preprocessing is a path to better data, and in turn, better model performance.

2.4 Predictive Modeling Is Mostly Data Preparation

Modeling data with machine learning algorithms has become routine. The vast majority of the common, popular, and widely used machine learning algorithms are decades old. Linear regression is more than 100 years old. That is to say, most algorithms are well understood and well parameterized and there are standard definitions and implementations available in open source software, like the scikit-learn machine learning library in Python.

Although the algorithms are well understood operationally, most don't have satisfiable theories about why they work or how to map algorithms to problems. This is why each

predictive modeling project is empirical rather than theoretical, requiring a process of systematic experimentation of algorithms on data. Given that machine learning algorithms are routine for the most part, the one thing that changes from project to project is the specific data used in the modeling.

> Data quality is one of the most important problems in data management, since dirty data often leads to inaccurate data analytics results and incorrect business decisions.
>
> — Page xiii, *Data Cleaning*, 2019.

If you have collected data for a classification or regression predictive modeling problem, it may be the first time ever, in all of history, that the problem has been modeled. You are breaking new ground. That is not to say that the class of problems has not been tackled before; it probably has and you can learn from what was found if results were published. But it is today that your specific collection of observations makes your predictive modeling problem unique. As such, the majority of your project will be spent on the data. Gathering data, verifying data, cleaning data, visualizing data, transforming data, and so on.

> ... it has been stated that up to 80% of data analysis is spent on the process of cleaning and preparing data. However, being a prerequisite to the rest of the data analysis workflow (visualization, modeling, reporting), it's essential that you become fluent and efficient in data wrangling techniques.
>
> — Page v, *Data Wrangling with R*, 2016.

Your job is to discover how to best expose the learning algorithms to the unknown underlying structure of your prediction problem. The path to get there is through data preparation. In order for you to be an effective machine learning practitioner, you must know:

- The different types of data preparation to consider on a project.
- The top few algorithms for each class of data preparation technique.
- When to use and how to configure top data preparation techniques.

This is often hard-earned knowledge, as there are few resources dedicated to the topic. Instead, you often must scour literature for papers to get an idea of what's available and how to use it.

> Practitioners agree that the vast majority of time in building a machine learning pipeline is spent on feature engineering and data cleaning. Yet, despite its importance, the topic is rarely discussed on its own.
>
> — Page vii, *Feature Engineering for Machine Learning*, 2018.

2.5 Further Reading

This section provides more resources on the topic if you are looking to go deeper.

2.5.1 Books

- *Feature Engineering and Selection*, 2019.
  https://amzn.to/3aydNGf
- *Feature Engineering for Machine Learning*, 2018.
  https://amzn.to/2XZJNR2

2.5.2 Articles

- Data preparation, Wikipedia.
  https://en.wikipedia.org/wiki/Data_preparation
- Data cleansing, Wikipedia.
  https://en.wikipedia.org/wiki/Data_cleansing
- Data pre-processing, Wikipedia.
  https://en.wikipedia.org/wiki/Data_pre-processing

2.6 Summary

In this tutorial, you discovered the importance of data preparation for each machine learning project. Specifically, you learned:

- Structured data in machine learning consists of rows and columns.
- Data preparation is a required step in each machine learning project.
- The routineness of machine learning algorithms means the majority of effort on each project is spent on data preparation.

2.6.1 Next

In the next section, we will take a tour of the different types of data preparation techniques and how they may be grouped together.