SAP White Paper
Feature Engineering

Feature Engineering Tips for Data Scientists

Improving Your Predictive Modeling with More Data

Table of Contents

Author: Patricia Tillotson, PhD, Senior Data Scientist, Big Data Analytics, SAP SE

Most data scientists and statisticians agree that predictive modeling is an art as well as a science. This paper describes one component of the art of modeling called feature engineering and offers a few tips to facilitate it. Using tips such as these is important because more data can increase model accuracy, which makes feature engineering an essential part of the modeling process.

The Art of Predictive Modeling

The science component of predictive modeling is clear. It embeds itself effortlessly into formulas and algorithms that calculate model results. The art of modeling, on the other hand, generally goes unnoticed through various stages of the modeling process, yet it can dramatically affect model accuracy. The art is often overlooked and not discussed in modeling arenas and, therefore, exists as a black box or hidden skill as far as new practitioners are concerned.

WHAT IS FEATURE ENGINEERING?

A predictive model is a formula or method that transforms a list of input fields or variables (x1, x2, ..., xn) into some output of interest (y). Feature engineering is simply the thoughtful creation of new input fields (z1, z2, ..., zn) from existing input data (x). Thoughtful is the key word here. The newly created inputs must have some relevance to the model output and generally come from knowledge of the domain (such as marketing, sales, climatology, and so on). The more a data scientist interacts with the domain expert, the better the feature engineering process.

Take, for example, the case of modeling the likelihood of rain given a set of daily inputs: temperature, humidity, wind speed, and percentage of cloud cover. We could create a new binary input variable called "overcast" where the value equals "no" or 0 whenever the percentage of cloud cover is less than 25% and equals "yes" or 1 otherwise. Of course, domain knowledge is required to define the appropriate cutoff percentage and is critical to the end result.

The more thoughtful inputs you have, the better the accuracy of your model. This is true whether you are building logistic, generalized linear, or machine learning models.

EXAMPLES OF FEATURE ENGINEERING

To show how feature engineering works, let's build the input fields for a marketing model. The model will predict which companies are most likely to become leads and, therefore, which companies should be targeted in an e-mail or telephone campaign. This is a complex problem.

Let's think about what input fields are appropriate here. First, there are data fields about the universe of companies you market to. The traditional firmographic inputs include data fields like:

- Number of employees
- Annual revenue
- Industry
- Customer status
- Geographic location
- Public or private company

"Feature engineering is another topic which doesn't seem to merit any review papers or books, or even chapters in books, but it is absolutely vital to [machine learning] success. . . . Much of the success of machine learning is actually success in engineering features that a learner can understand."

Scott Locklin, "Neglected Machine Learning Ideas"

As we think in more depth about the problem (domain expertise coming in here), it seems reasonable to imagine that companies who respond to marketing campaigns are more likely to become leads. There are a host of recency, frequency, and monetary (RFM) variables that deal with the response component of inputs. So for contacts within a company, we can include RFM data fields such as:

- Number of past responses to e-mail campaigns
- Number of Webinars attended
- Number of physical events attended
- Days since last marketing response
- Value of last product purchased

Then there are variables that describe the particular contact within the company. Examples are:

- Department
- Title
- Gender
- Years of experience

So our original data set and list of inputs looks something like this.

Company ID	Name	Department	Title	Gender	Years of experience	Company number of employees	Company annual revenue (millions)	Company industry	Company customer status	Geographic location	Public or private company	Number of past responses to e-mail campaigns	Number of Webinars attended	Number of physical events attended	Days since most recent marketing response	Value of last product purchased (thousands)
101	John Smith	IT	CIO	M	20	5,329	1,200	Healthcare	Customer	Midwest	Public	5	10	2	33	23
101	Carrie Jones	Finance	Analyst	F	7	5,329	1,200	Retail	Non-customer	Northeast	Private	3	4	5	27	10
101	Darrel Thomas	HR	VP	M	13	5,329	1,200	High tech	Customer	South	Private	2	8	0	6	56
102	Marty Woodrow	Sales	Director	M	10	25	10	Utilities	Customer	Northeast	Private	10	2	1	10	2
103	Phil Campella	Marketing	Manager	M	5	1,009	556	Oil and Gas	Non-customer	West	Private	8	1	1	156	35
103	Doreen Machu	IT	Database manager	F	5	1,009	556	Automotive	Customer	Midwest	Public	6	12	7	42	55

If we begin with these as inputs, we can create a host of other data fields through feature engineering:

- Number of responses by a contact within the past two weeks
- Number of responses two to four weeks ago
- Number of responses more than three months ago

The rationale is that the more recent the response, the more the company would turn into a lead.

We can "roll up" responses by contacts to the company level to come up with:

- Total number of company responses
- Maximum number of responses by any contact within the company
- Average number of responses by contacts in the past three months
- Number of contacts at the company who have responded in the past month
- Number of contacts who have attended events in the past three months

You can see that we started with 15 input fields and quickly expanded the list to a total of 23, and the list goes on and on.

So how exactly do you produce a list of new variables? What's the thought process? What are the steps? In the following section we outline some of the steps to creating new variables, focusing on tips that will help you to improve feature engineering. This is the process that SAP and other companies take in their modeling approach. In fact, many of the variables discussed here have been used to build a model that predicts which marketing responders will most likely turn into leads.

The more a data scientist interacts with the domain expert, the better the feature engineering process.

SIX TIPS FOR BETTER FEATURE ENGINEERING

Tip 1: Think about inputs you can create by **rolling up existing data fields to a higher or broader level** or category. As an example, a person's title can be categorized into strategic or tactical. Those with titles of "VP" and above can be coded as strategic. Those with titles of "Director" and below become tactical. Strategic contacts are those that make high-level budgeting and strategic decisions for a company. Tactical are those in the trenches doing day-to-day work.

Other roll-up examples include:

- Collating several industries into a higher-level industry: Collate oil and gas companies with utility companies, for instance, and call it the energy industry, or fold high-tech and telecommunications industries into a single area called "technology."
- Defining "large" companies as those that make US$1 billion or more and "small" companies as those that make less than $1 billion

Tip 2: Think about ways to **drill down into more detail in a single field.** As an example, a contact within a company may respond to marketing campaigns, and you may have information about his or her number of responses. Drilling down, we can ask how many of these responses occurred in the past two weeks, one to three months, or more than six months in the past. This creates three additional binary (yes = 1/no = 0) data fields for our model.

Other drill-down examples include:

- Cadence: Number of days between consecutive marketing responses by a contact (1–7, 8–14, 15–21, 21+)
- Multiple responses on same day flag (multiple responses = 1, otherwise = 0)

Tip 3: Split data into separate categories also called bins. For example, annual revenue for companies in your database may range from $50 million (M) to over $1 billion (B). Split the revenue into sequential bins: $50M–$200M, $201M–$500M, $501M–$1B, and $1B+. Whenever a company falls with the revenue bin, it receives a 1; otherwise the value is 0. There are now four new data fields created from the annual revenue field.

Other examples are:

- Number of marketing responses by contact (1–5, 6–10, 10+)
- Number of employees in company (1–100, 101–500, 502–1,000, 1,001–5,000, 5,000+)

Tip 4: Think about ways to **combine existing data fields into new ones.** As an example, you may want to create a flag (0/1) that identifies whether someone is a VP or higher and has more than 10 years of experience.

Other examples of combining fields include:
- Title of director or below **and** in a company with less than 500 employees
- Public company **and** located in the Midwestern United States

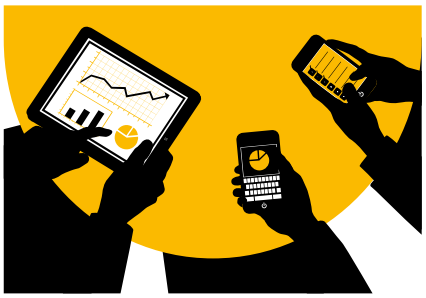
You can even multiply, add, or subtract one data field by another to create a new input.

Tip 5: Don't reinvent the wheel – **use variables that others have already fashioned.**

Tip 6: Think about the problem at hand and be creative. Don't worry about creating too many variables at first. Just let the brainstorming flow.

Feature selection methods are available to deal with a large input list; see the excellent description in Matthew Shardlow's "An Analysis of Feature Selection Techniques." Be cautious, however, of creating too many features if you have a small amount of data to fit. In that case you may overfit the data, which can lead to spurious results.

Hundreds, even thousands of new variables can be created using the simple techniques described in this paper. The key is to develop thoughtful additional variables that seem relevant to the target or dependent variable. So go ahead, be creative, have fun, and enjoy the process.



LEARN MORE
To learn more about predictive modeling, visit www.sap.com/predictive.

The Best-Run Businesses Run SAP®