



"turning data into dollars"

Tom's Ten Data Tips – December 2007

Data Preparation

Data Preparation appears to be where data miners spend most of their time. Some say that 80%-90% of time in an average data mining project is spent "merely" preparing the data. And this is time well spent to end up with a good predictive model, yet data preparation and feature extraction are underrepresented in the data mining literature.

Some tools assist in automatically preparing the data, but few, very few do even a decent job at this. Barring statistically sound missing value replacement procedures, this part of the data mining process is bound to remain a cumbersome and time consuming effort for some time to come. Automated procedures for handling date variables, etc., would be useful, but are notoriously difficult to program.

1. The *Entire* Data Preparation Process Needs To Be Replicable In One Single Program

When you prepare a data set, what you do in essence is present the data to the algorithm in a "format" that makes it easiest to discern the meaningful patterns.

All the operations that are performed on the mining set, need to be replicated when applying the eventual data mining model on run-time data. Therefore, each conversion, imputation, etc. needs to be replicated by means of a computer program that can "automatically" be run on the "live" data. Otherwise, real-life data would not be presented in the same manner to the data mining algorithm, as they were in the training set. This would render the model invalid.

2. Hang On To The Missing Value Pattern

When missing values in a data set must be replaced (for instance, when working with a regression model), it can be highly informative to retain the missing value pattern as separate variables in the data set. This can be done in a simple format by Boolean indicators that display a "1" to represent the value was initially missing, and a "0" otherwise. From this missing value pattern, new (potentially valuable) variables

can be derived: for instance, the total number of missings in each record.

By using only the missing value replacement columns as predictor variables, one can quickly and easily test if the *pattern* in missing values is associated with the target variable (usually it is). Only use the missing value indicators as input variables, and the planned target variable as the output. If this missing value replacement model has significant predictive power, you have demonstrated that the *pattern* in missings is relevant. If this is the case, you need to be extra careful how to replace the missing values (see also tip #3).

3. Missing Values Should *Not* Be Replaced By The Median (Or Any Constant)

There is a tendency (and this is often the default setting – sic!) to replace missing values by either the median or modus value in a column. This is very bad practice because such a procedure is guaranteed to add bias to the dataset.

A prerequisite of *any* missing value replacement procedure is that it should not add bias to the data set through the imputed values. For one thing, the variance in the variable needs to remain the same after missing replacement. That is (one reason) why it can *never* be good to replace missing values by *any* constant, because it will always reduce the variance in the variable.

4. Balancing The Dataset Influences Cumulative Gains Curves

Depending on which algorithm you are using, the needs for “balancing” your dataset will be different. Balancing here refers to stratified sampling from output categories, to ensure that training will converge on the best possible model, and/or performance will be better. In applications like direct marketing and fraud there are usually abundant non-responders or non-fraudulent cases. Then the category of interest (response, fraudulent transactions) will be taken entirely, and from the other category you take a sample. The target category will now be overrepresented in the mining set, relative to the population.

Since most people are accustomed to using the cumulative gains chart to assess lift of a model (often incorrectly referred to as “lift curves”), it is important to note that the cumulative gains curve is not invariant to the sampling distribution. When the category of interest is (severely) underrepresented, the curve will (falsely) *suggest* a model

with more predictive power. Keep this in mind when comparing lift curves across models with different underlying distributions of the target variable.

5. Date/Time Variables Require Multiple Representations

Dates are notoriously difficult variables to prepare for modeling (or reporting for that matter). This has to do with the fact that any particular Gregorian date in itself is usually meaningless. "Gregorian dates" refers to how we think in terms of a specific date on the calendar in everyday language. A date for the purpose of a model derives its meaning from the relation to other meaningful dates. For example a product acquisition date has relevance when the number of days expired since today is calculated, to derive the product ownership tenure. Or a purchase date may be relevant as the number of days before a holiday. For instance, purchases may be related to the number of days before Christmas that the sale took place. That, for one thing, is why dates are often internally represented as Julian dates in most database applications. Because there are usually multiple reference dates to compare to, you need multiple derived Julian dates. The reference point is often here and now, or some fixed event in the past (say, origination date), or the future (typically holidays). Sometimes it can be a reference point in the customer lifecycle that would therefore be different for all customers.

6. Date Variables Are Essentially Cyclical

Date/time variables have an intrinsically cyclical nature to them. A deliberate effort in preparation is required to surface this quality in the way variables are represented. For example, a week number 52 should really be represented in a form that demonstrates its proximity to week 1. By the same token, 23:55 is very close to 0:05 o'clock. In order to represent this temporal proximity to, for instance, a neural network tool, one needs to take special steps.

Representing time appropriately can be done in one of two ways. Either by making time linear, or by representing time in a cyclical form that "makes" 23:55 close to 0:05. For the linear representation, one can simply calculate the number of seconds/minutes/hours/days since or from a reference point (as in tip #5). For the circular representation of time, something else needs to be done. What this requires is using two cyclical variables in conjunction, a sine and cosine, that are $\frac{1}{2}\pi$ apart. That way, these two variables can represent the X and Y coordinates of a cycle that represents time.

7. Binning Of Continuous Variables Should Be Model Specific

When you use continuous variables as predictors, it is often useful, and sometimes even necessary to pull together ranges of the variable (nominal, ordinal, or interval scales) in bins. Best practice is to create these bins through some computational procedure that maximizes the contribution of the input variable for predicting the output variable.

Such computational procedures can be extremely time consuming. As a result, they are sometimes only performed once (when the generic mining table is created). But because the optimal bin boundaries are specific to a particular output variable you are predicting, this leads to sub optimal predictions. Even worse is the practice of clunking input variables in fixed ranges (of the predictor), or in groups of records of equal size (although not as bad, in practice).

8. Derived Variables From Transient Behavior Tend To Be The Strongest Predictors

When building models on customer data, one can imagine a continuum running from highly volatile and transient behaviors to more stable and “fixed” characteristics. An example of transient behavior would be the number of interactions with the contact center in the past month. An example of a “fixed” characteristic might be gender, or attributes tied to the zip code of residence.

In predictive models, the transient or behavior related variables are nearly always the strongest predictors. There's a downside to using them however, because their predictive power also tends to decay faster over time when the model gets reused repeatedly.

9. Derived Variables Are The Hallmark Of Data Research & Development

A mining set contains one row per entity you are modeling (customer, shipment, transaction). The ER or Star schema from which these input variables are derived can be highly complex. Exploratory data mining can be aimed at finding data representations that are meaningful descriptors and powerful predictors. This, in essence, is data research & development. It can bear fruits for predictive modeling, as well as a raft of other applications. An example is derived scales we discussed in a previous newsletter on affinity analysis. Data are an abstraction of reality. Because derived variables imbue reality with new meaning,

they form a “lens” through which we view the world. A perspective on the market or customer behavior is created by means of derived variables. As an example, handset type invariably is an important predictor in churn models for telcos. The initial nominal variable that defines the handset type each customer owns, though, is pretty much useless. The impact any specific handset has on probability to churn needs to be derived by creating a variable that indicates something like “trendiness” of handset. Some handset’s trendiness decays much faster over time than others, and this needs to be reflected in the derived variable(s). The Nokia 3310, for instance, never seems to go out of style, or maybe it never was “hip” to begin with.

10. The Quest For Better Variables Never Ends

Ongoing research on how to derive better predictive variables from underlying information sources ultimately leads to a richer data environment. Examples are deriving aggregates. In one project we used the slowly changing customer dimension to derive information on how often people in different zip code areas changed address (on average). The derived average can then be tied to any customer’s zip code, and be used as a predictor. Another example might be to aggregate purchasing transactions into categories, and append the number of purchases in each category to the customer record in the mining table.

As models are developed on an ongoing basis, evidence accumulates which variables often had a strong influence in models, and which didn’t. Then the next step is to do research to find out if additional variables in relevant domains might be extracted from existing (or even new) data sources. Less successful variables that were never included in final models can eventually be dropped. This leads to a development cycle of potential predictors that are added and dropped from the candidate input variable list.