we discovered (painfully) that missing values were sometimes miscoded with a large positive number (e.g., 99999). This led to chaos in the correlation checks, where the value reported depended only on whether or not missing cell values in two variables were on the same case.

b. Principal components (PC): You may retain the vast majority of the variance in the input data by replacing the original variable, $X$, with a smaller subset of PCs. These are linear transformations of $X$ designed to be mutually orthogonal and span as much of the input space as possible. For instance, you might be able to represent 90% of the space covered by cases in 150 variables using only 20 PC dimensions. Note, however, that PCs don’t consider the output variable when being discovered, so they may not be the best vocabulary for classification; note also that they still require measuring all of $X$ to be calculated.

c. Follow the choices of variable-selecting algorithms: Many algorithms, such as neural networks or nearest neighbors, don’t themselves select variables. But you can first run different algorithms that do—such as stepwise regression, decision trees, or polynomial networks (Elder and Brown, 2000)—and follow their lead. Try using only the superset of variables that they pick up. There is no guarantee this will be the best set for your algorithm, but the approach often proves useful in practice.

4. Divide and conquer. Many simple models may be more useful than a single complex one. You can remove from training any simple subset of the problem you can clearly define and focus modeling energies on the hard part. For instance, if all patients with a certain symptom are to be recommended for immediate treatment, take that out of the data as a known situation and train on the rest. Slice the data enough this way and many aspects of the problem will change, leading to the possibility of a novel discovery due to the novel perspective of what the problem really is.

5. Combine variables to create higher-order features. Don’t try to “build a critter from pond-scum”; use higher-order components. For instance, on a trajectory estimation problem, calculate where the craft will land without any complex effects, such as earth’s rotation and air resistance, and estimate the shortfall instead of the full distance.

6. Impute missing data. The easiest way to handle missing data on training is to not allow it. This means deleting either a case or variable for which a cell is empty, so this can vastly reduce your data. To get the benefit of “holey” cases, try filling in the data with different alternatives:
   a. Mean of known cases for variable
   b. Last value (if cases are in order)
   c. Estimated value from other known input values (best, but most complex)
   d. If categorical, the label “missing”

7. Explode categorical variables to allow use of estimation routines. A categorical variable can’t be used directly in an estimation algorithm like regression or neural networks. But you can “explode” a $C$-category variable into $C$ binary variables where each holds a 1 if the category is its value for that case.

8. Merge categories if there are too many. Each value of a categorical variable is usually allowed to have its own parameter by an algorithm, so overfit is very likely if there are