The Art & Science of

Data Preparation for Predictive Analytics

July 27, 2016

Author: Jen Underwood
Technical Reviewers: Arijit Sengupta, Chris Ames
Introduction
Introduction to BeyondCore

BeyondCore analyzes millions of data combinations in minutes, for unbiased answers, explanations and recommendations. Unlike manual data analysis, BeyondCore smart data discovery automatically finds and explains statistically significant key metric drivers that truly matter. It explains what happened, why it happened, what will happen and how you can improve it. In a rapidly-paced digital business era, BeyondCore can be an invaluable asset in your analytics arsenal.

Finding the Signal in the Noise

To identify relationships in data “the signals” and isolate distracting, irrelevant data “the noise”, BeyondCore uses powerful machine learning algorithms (prediction models) for estimating relationships among variables. To get started with BeyondCore, select a few columns of data including one outcome measure (revenue, units, days) and several predictor, independent variables (product category, region, date of sale, customer type) to find what insights are immediately identified. Then you can continue to further experiment with more columns of data or by organizing your data set in a machine learning-friendly format.

BeyondCore does not require data preparation to get started. You can connect or upload a few columns of data to enjoy rapid insights in minutes. It is best to start with a narrow scope of columns and then add more data using the following techniques to optimize results.

Ideally subject matter experts that understand the business process and the data source nuances would assist in collecting, cleansing and shaping data for BeyondCore analytics automation. Depending on your project, data preparation might be a one-time activity or a periodic one. As new insights are revealed, it is common to further experiment by adding or changing aspects of the BeyondCore input data.

Data preparation for predictive analytics is both an art and a science. Since each data set and business objective can be unique with varied data preparation challenges, we have provided the following guidelines to help get you started. In practice, you will iteratively add your own creative approaches and data preparation techniques to further enhance your BeyondCore project results.
Preparing Data
Preparing Data for Optimal Insights

The data preparation process involves choosing an outcome measure to evaluate, potential influencer variables, cleansing the data, creating features and generating data sets to provide to BeyondCore for automated analysis.

Throughout this document, we will walk through a simple Sales example. The source file for the Sales example is SalesOpportunities.csv. You can download it from the Resources section on BeyondCore.com or Support.

The Basics

BeyondCore requires data input as a table, view, or comma separated (.csv) flat file of rows and columns. If you have data stored in a dimensional data warehouse or transactional database format, you will need to use record identifiers or primary keys to join fields from multiple tables to create a single unified, flattened “view”. Your view will contain an outcome metric along with input predictor variables collected at a level of analytical granularity that you can make actionable decisions upon.

<table>
<thead>
<tr>
<th>Opportunity ID</th>
<th>Days Between Lead to Last Contact</th>
<th>Lead Source</th>
<th>Product</th>
<th>Region</th>
<th>Vertical</th>
<th>Close Date</th>
<th>Discount</th>
<th>Amount</th>
<th>Competition</th>
<th>Promotion</th>
<th>Won</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1D8100</td>
<td>82</td>
<td>Channel Part</td>
<td>Product 1</td>
<td>Germany</td>
<td>Utilities</td>
<td>0</td>
<td>8195.02</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>O1D5447</td>
<td>157</td>
<td>Article</td>
<td>Product 1</td>
<td>Germany</td>
<td>Insurance</td>
<td>0</td>
<td>74127.35</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D5452</td>
<td>88</td>
<td>Article</td>
<td>Product 1</td>
<td>Germany</td>
<td>Insurance</td>
<td>0</td>
<td>75373.2</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D1200</td>
<td>257</td>
<td>Call In</td>
<td>Product 1</td>
<td>Germany</td>
<td>Insurance</td>
<td>0</td>
<td>70647.65</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D1205</td>
<td>257</td>
<td>Email Campai</td>
<td>Product 1</td>
<td>Germany</td>
<td>Public Sector</td>
<td>0</td>
<td>56048.85</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D16372</td>
<td>118</td>
<td>Web Site</td>
<td>Product 1</td>
<td>India</td>
<td>Oil &amp; Gas</td>
<td>0</td>
<td>75373.2</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D18094</td>
<td>201</td>
<td>Cold Call</td>
<td>Product 1</td>
<td>Africa</td>
<td>Oil &amp; Gas</td>
<td>0</td>
<td>72302.26</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D19543</td>
<td>107</td>
<td>Cold Call</td>
<td>Product 1</td>
<td>Africa</td>
<td>Middle East Retail</td>
<td>0</td>
<td>51836.56</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D20931</td>
<td>193</td>
<td>Call In</td>
<td>Product 1</td>
<td>Africa</td>
<td>Banking</td>
<td>0</td>
<td>57779.15</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D22012</td>
<td>34</td>
<td>Web Site</td>
<td>Product 1</td>
<td>Brazil</td>
<td>Media</td>
<td>0</td>
<td>91096.87</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D22104</td>
<td>155</td>
<td>Article</td>
<td>Product 1</td>
<td>Brazil</td>
<td>Banking</td>
<td>0</td>
<td>94551.17</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D22215</td>
<td>127</td>
<td>White Paper</td>
<td>Product 1</td>
<td>India</td>
<td>Media</td>
<td>0</td>
<td>59771.41</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>O1D22511</td>
<td>65</td>
<td>Web Site</td>
<td>Product 1</td>
<td>Latin America</td>
<td>Consumer Products</td>
<td>0</td>
<td>80433.3</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>O1D26385</td>
<td>3</td>
<td>Existing Acco,</td>
<td>Product 1</td>
<td>India</td>
<td>Chemicals</td>
<td>0</td>
<td>71895.78</td>
<td>Analytix</td>
<td>QuickStart</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>O1D26490</td>
<td>107</td>
<td>Existing Acco,</td>
<td>Product 1</td>
<td>Germany</td>
<td>High Tech</td>
<td>0</td>
<td>93880.56</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D31446</td>
<td>72</td>
<td>Email Campai</td>
<td>Product 1</td>
<td>Italy</td>
<td>Passenger Travel &amp; Leisure</td>
<td>0</td>
<td>97686.63</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D45579</td>
<td>25</td>
<td>Cold Call</td>
<td>Product 1</td>
<td>India</td>
<td>Professional Services</td>
<td>0</td>
<td>65974.07</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>O1D61277</td>
<td>141</td>
<td>White Paper</td>
<td>Product 1</td>
<td>Western</td>
<td>U. Chemicals</td>
<td>0</td>
<td>72204.06</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>O1D65067</td>
<td>217</td>
<td>Web Site</td>
<td>Product 1</td>
<td>South East</td>
<td>Media</td>
<td>0</td>
<td>63513.49</td>
<td>Analytix</td>
<td>Other Promotion</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 1 Flattened analytical view*
For many outcome metrics, data is captured at various business process steps in multiple data sources. A sales process might have data in a CRM, email marketing program, Excel spreadsheet and accounting system. If that is the case, you will want to identify the fields in those systems that can relate, join or blend the different data sources together.

If you trying to capture changes in data over a time period, check if your data source is only keeping the current state values of a record. Most data warehouses are designed to save different values of a record over time and do not overwrite historical data values with current data values. Transactional application data sources, Salesforce for instance, only contain current state value for a record. If you want to get a prior value, you would need to have a snapshot of the historical data stored or keep the prior value data in custom fields on the current record.

After identifying outcome metric business process data sources and fields, you will want to select any fields (predictor variables) that may directly affect the outcome. In doing so, ensure that the variable data is clean and consistent. The order and meaning of input predictor variables should remain the same from record to record. Inconsistent data formats, “dirty data” and outliers can undermine the quality of analytical findings.

If you do have data issues, you can still get started with BeyondCore. During the data load process, BeyondCore asks if you want to remove outliers and identifies a variety of other common issues along with suggested potential actions you can have BeyondCore automatically apply for you to save time.

Then you will shape the data into analytical features with derived variables that you feel might describe or influence the outcome metric. If you ever saw the movie or read the book “Moneyball: The Art of Winning an Unfair Game” by Michael Lewis that discusses the revolution of baseball analysis with new performance metrics On-Base Percentage (OBP) and Slugging Percentage (SLG), essentially you will be using a similar approach. Shaping data involves subject matter expert thought to creatively select, create and transform variables for maximum influence.
Keep in mind that analytical feature variables do need to be prepared at the correct level of granularity for decision making. You don’t want to overly aggregate and you may not need lower levels of detail. Choose a granularity that is actionable, understandable, and useful in the event you incorporated the results into your existing business process or application.

A common mistake is to overly aggregate data. Keep the BeyondCore desired outcome in mind and use data collected in rows at that level of granularity. Data analyzed in Excel might be at a different level than you will want to analyze it in BeyondCore. For example, if want to understand the effects of day of week. Provide BeyondCore data at the day level. You cannot predict a day level outcome from an aggregated monthly level dataset.

Also analytical rows of features are wide when compared to reporting or transactional application rows that are optimized for other purposes. Statistical analysis data sets can summarize a lifetime of values in just one single row with many columns that describe different points in time. On that note, don’t collect a lifetime of features if a window of time, “right horizon”, will more accurately predict the outcome metric than all time. Usually events closest to the outcome are stronger predictors than events that happened a long time ago.

If you do want to use BeyondCore for predictions, then your variables need to be at the point in time that the prediction is based on. For example, if your objective is to decrease defaults on loans by not pre-approving loans that are likely to default. You will need to capture variables such as a credit score at the time of loan application and prior; if the person was late on two payments after loan origination it would not be used in pre-approval analysis since they have already been approved.

Figure 2 Input Variables transformed into a Feature
Getting Started
Getting Started

Step 1: Select an Outcome Metric

The first thing you need to decide is what metric you want to better understand and at what granularity. A metric could be a revenue, discount or cost measure, a duration, or any other business process number. You can also use categories with two values as a numeric outcome such as Win (100)/Loss (0) as we did in the Sales example. Notably two value outcomes are less accurate than continuous value outcomes.

BeyondCore algorithms assume that each record is independent and are not related to other records. If relationships exist between records, you will want to create a new variable called a feature within the row of data to capture that behavior. For instance, if the same Opportunity had multiple Competitors you would not prepare multiple rows of data with the same Opportunity ID. You would create additional fields on one Opportunity ID called Competitor 1, Competitor 2 and so on.

![Figure 3 Independent record design](image)

Granularity refers to a unit of analysis. That unit might be an opportunity, customer or a transaction with variables/columns containing descriptive data. Granularity is determined by the business objectives and how the model will be used operationally. Are decisions made from the model scores based on a transaction? Are they made based on the behavior of a single customer? Are they made based on a single visit, or based on aggregate behavior of several transactions or visits over a time period?

---

1 Applied Predictive Analytics: Principles and Techniques for the Professional Data by Dean Abbott
Step 2: Select Predictor Variables

Next you will want to think about what variables might relate, describe or influence the numeric outcome. In our sales example, potential influencer variables include Discount, Days between Lead Received and Last Contacted, Lead Source, Region, Vertical, Competitor, and Promotion. When selecting predictor variables, keep in mind that you want to gather a maximum amount of information from a minimum number of variables.

Take note that dates will likely be rolled up to a duration when used as input in BeyondCore. If there are multiple key dates in a business process, create multiple variables in the BeyondCore prepared input view to store numeric durations. For example, Days between Lead to Last Contact and Days between Demo to Trial. Common date variable rollups include the earliest date and the most recent date. Time durations can also represented in either absolute or relative form.

For proof-of-concept projects, try to keep your input data between 10 and 25 variables. It is much faster to learn and iteratively improve your data preparation skills with BeyondCore on less complex models.

After you have mastered data preparation, in production you can use as many variables as you want. BeyondCore will automatically figure out which variables and combinations of variables best explain the behavior of your chosen metric without overfitting or underfitting your analysis.

Overfitting is a common beginner predictive modeling mistake that happens when too many variable fields are used in a predictive model. Overfitting captures the noise in your data with an overly complex, unreliable predictive model. Essentially what happens is the model memorizes unnecessary details. When new data comes in, the model fails. For example, if you wanted to predict sales amount you might want to include customer demographic variables such as gender and age. You should exclude variables that are too detailed such as the customer shoelace color that would cause overfitting in your model.

Underfitting is often a result of an excessively simple model. Underfitting occurs when a statistical algorithm cannot capture the underlying patterns in the data. Thus there is a delicate balance needed between being too specific with too many variables and too vague with not enough selected variables for an outcome.
Step 3: Determine How Much Data to Model

Then you will decide how much data you want to explore. Keep in mind that BeyondCore’s statistical-based engine does require a minimum of 10,000 rows of data for accuracy. If you are using a free trial of BeyondCore, you will be limited to one million rows of data. For proof-of-concept engagements, we recommend using between 10,000 and one million rows of data. In production environments, there is not a hard limit on the number of rows that can be analyzed. You can load as much data as your BeyondCore server can handle.

If you are using a free trial of BeyondCore, there is a limit of 12 columns and one million rows of input data.
Step 4: Cleanse and Prepare Data

Now you will assess the condition of your source data. As you collect the data into variables, you will want to profile the values. Specifically look for extremes, outliers, missing, values, incorrect values, skew, and high cardinality.

BeyondCore does provide capabilities in the data loading process to identify common data preparation issues. You can optionally repair them in BeyondCore or in your data preparation process.

We recommend that you address data quality issues as early as possible. If you are seeing errors from source applications that should get fixed, a best practice is to try and resolve the issue at the source system versus in a data preparation process. Here are a few approaches for handling common data issues for predictive data preparation.

Handling Extreme Values and Outliers
Outliers are values that exceed three standard deviations from the mean. BeyondCore algorithms are sensitive to outliers since those values affect averages (means) and standard deviations in statistical significance calculations. If you come across unusual values or outliers, confirm if these data points are relevant and real. Often odd values are errors. If the extreme data points are accurate, predictable and something you can count on happening again, do not remove them unless those points are unimportant. You can reduce outlier influence by using transformations or converting the numeric variable to a categorical value with binning.

Dealing with Missing Values
The most common repair for missing values is imputing a likely or expected value for it using a mean or computed value from a distribution. If you use mean, you may be reducing your standard deviation thus the distribution imputation approach is more reliable.
Another approach is to remove any record with missing values. Don’t get too ambitious with filtering out missing values. Sometimes the pattern is in the missing data. Also if you delete too many records, you will undermine the real-world aspects in your analysis.

As you address missing values, do not lose the initial missing value context. A common data preparation approach is to add a column to the row to flag data was missing coded with a 1 or 0.

**Correcting Incorrect Values**

Predictive algorithms assume the input information is correct. Treat incorrect values as missing if there are only a few. If there are a lot of inaccurate values, try to determine what happened to repair them.

<table>
<thead>
<tr>
<th>CENTRAL TENDENCY</th>
<th>DISPERSION</th>
<th>VISUALIZATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Range</td>
<td>Histogram</td>
</tr>
<tr>
<td>Median</td>
<td>Quartile</td>
<td>Box Plot</td>
</tr>
<tr>
<td>Mode</td>
<td>IQR</td>
<td>Bar Chart</td>
</tr>
<tr>
<td>Min</td>
<td>Variance</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>Standard Deviation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Skewness and Kurtosis</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 5 Techniques to evaluate variable skewness*

**Reducing Skew**

For continuous variables, review the distributions, central tendency and spread of the variable. These are measured using various statistical metrics visualization methods. Continuous variables should be normally distributed. If not, try to reduce skewness for optimal prediction.

For categorical variables, use a frequency table to understand distributions of each category along with a bar chart.

If variable values are skewed, BeyondCore might produce biased models. When a skewed distribution needs to be corrected, the variable is typically transformed by a function that has a disproportionate effect on the tails of the distribution. Log transform (log(x), logn(x), log10(x)), the multiplicative inverse (1/x), square root transform sqrt(x) or power (x^n) are the most often used corrective functions.
In the table to the left, several issues and formulas to minimize skew are shown. The before chart illustrates how a skewed variable distribution might look before any preparation is performed. After applying one of the fixes, a normal distribution for the variable is achieved. The newly prepared, transformed variable will perform much better in BeyondCore automation for predictive modeling purposes.

### Eliminating High Cardinality Fields

High-cardinality fields are categorical attributes that contain a very large number of distinct values. Examples include names, ZIP codes or account numbers. Although these variables could be highly informative, high-cardinality attributes are rarely used in predictive modeling. The main reason is that including these attributes will vastly increase the dimensionality of the data set making it difficult or even impossible for most algorithms to build accurate prediction models.

**BeyondCore provides features in the data loading process to automatically identify high cardinality fields. This enables you to easily filter them out.**

### Handling Ordinal Variables

**Ordinal** variables are usually problematic for predictive models. Ordinal data consists of numerical scores on an arbitrary scale that is designed to show ranking in a set of data points. For example, Low, Medium and High are ordinal. Predictive algorithms will assume the variable is an interval or ratio variable and may be misled or confused by the scale. Most often ordinal variables are treated as categorical. If you have do ordinal values, transform them into a continuous or categorical one.
Avoid Duplicate, Redundant or Highly Correlated Variables

Duplicate, redundant or other highly correlated variables that carry the same information should be minimized. BeyondCore algorithms perform better without those kinds of collinear variables. Collinearity occurs when two or more predictor variables are highly correlated, meaning that one can be linearly predicted from the others with a substantial degree of accuracy.

To avoid the collinearity issue, do not include multiple variables that are highly correlated or data that is from the same reporting hierarchy. Often those fields end provide obvious insights. For example, customers who live in the city of Tampa also happen to live in the state of Florida. If you have two variables that are almost identical and you do want to retain the difference between them, consider creating a ratio variable as a feature. Another approach is to use Principal Component Analysis (PCA) output as input variables.

To identify high correlation between two continuous variables, review scatter plots. The pattern of a scatter plot indicates the relationship between variables. The relationship can be linear or non-linear. To find the strength of the relationship, compute correlation. Correlation varies between -1 and +1.

The following formula can be used to compute variable correlation.

\[
\text{Correlation} = \frac{\text{Covariance}(X,Y)}{\sqrt{\text{Var}(X) \cdot \text{Var}(Y)}}
\]

Many analysis tools contain functions to identify correlation between variables. In Excel, CORREL() can be used to return the correlation between two variables. You can also manually compute it.
Step 5: Create Analytical Features

Analytical feature creation is the art of extracting more information from existing data. You are making the data you already have more useful. Strong features that precisely describe the process being predicted can improve pattern detection and enable more actionable insights to be found.

Creating features from several combined variables and ratios usually provides more improvement and model accuracy than any single-variable transformation because of the information gain associated with these interactions.

Aggregations

Some commonly computed aggregate features including the mean (average), most recent, minimum, maximum, sum, multiplying two variables together and ratios made by dividing one variable by another.

Ratios

Interestingly, ratios can be excellent feature variables. Ratios can communicate more complex concepts such as price-to-earnings ratio where neither price nor earnings alone can deliver this insight.

Features that add or subtract two variables are not necessary for you to create. BeyondCore automatically develops these kinds of features in your model.

Transformations

Transformation refers to the replacement of a variable by a function. For instance, replacing a variable x by the square / cube root or logarithm x is a transformation. You transform variables when you want to change the scale of a variable or standardize the values of a variable for better understanding. Variable transformation can also be done using categories or bins to create new variables. An example transformation might be binning continuous Lead Age into Lead Age Groups or Price into Price Categories such as Discount, Retail, and OEM.
Step 6: Generate Data Sets

The most common strategy is to split data into training, testing and validation data sets. These data sets are usually created through a random selection of records without replacement meaning each record belongs to one and only one subset. All three data sets must contain at least 10,000 rows of data and should reflect your real world scenario.

How Much Data to Get

As mentioned previously, you do will need to provide BeyondCore with at least 10,000 records that resemble real world distributions of variables to build reliable predictive models. The actual number of records is not always easy to determine and depends on patterns found in your data. If you have more noise in your data, you will need more data to overcome it. Noise in this context means unobserved relationships in the data that are not captured by the input predictor variables.

<table>
<thead>
<tr>
<th>DATA SETS</th>
<th>BASED ON PERCENTAGE OF RECORDS</th>
<th>SAMPLE SIZE PERCENTAGE OF RECORDS</th>
<th>DATA SETS NUMBER OF RECORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>50%</td>
<td>500,000</td>
<td>100,000</td>
</tr>
<tr>
<td>Testing</td>
<td>25%</td>
<td>250,000</td>
<td>100,000</td>
</tr>
<tr>
<td>Validation</td>
<td>25%</td>
<td>250,000</td>
<td>800,000</td>
</tr>
</tbody>
</table>

*Figure 8 Data set size guidelines*

To determine data set size consider the dimensionality of your data and pattern complexity.

- For smaller models with a few variables, ~10 to 20 records per variable value may be sufficient.
- For more complex models, ~100 records per variable value may be needed to capture patterns.
- For complex models with ~100 input variables, you will need a minimum of 10,000 records in the data for each subset (training, testing, and validation).
Cross Validation

BeyondCore has an optional k-fold cross validation feature that divides the data into k subsets with the holdout repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. The BeyondCore K-fold feature enables you to independently choose how much data you want to use in testing.

BeyondCore automatically uses industry standard k-fold cross validation for predictive model evaluation when it is selected during Story development.

Time Series Consideration with Data Sets

Data that changes over time should be reflected in your BeyondCore model features and also in your input data set. When time sequences (Lead Received > Quote Provided > Deal Closed) are important in predictions, proportionally collect data from those different time periods. The key principle is to provide data that reflects what actually happens in the real world at the right level of outcome metric granularity.

Thinking Proportionally

When collecting data, think about the balance of your variables in your raw data. For example, how many vertical industry records are there by time period? When extracting a subset of data, be sure to include approximately the same proportion of variables in your BeyondCore input data set. If you provide more records of one variable, vertical in our example, you can unintentionally bias your analysis.

If you have data sets with millions of rows, it is unlikely that you will encounter accidental bias. Essentially you want to provide BeyondCore with enough data that is similar to the data in the real world environment.
Summary

The beauty of the expert human mind in combination with BeyondCore analytics automation empowers amazing predictive insights that might never be found using manual analysis techniques. Since the quality of BeyondCore output does rely on high quality input, data preparation can be a critical success factor for achieving optimal results. BeyondCore does include an array of data preparation features when you initially load data. There are also a plethora of third-party tools and utilities that can further expedite data cleansing and wrangling tasks.

Predictive analytics is an iterative process that can continue after your solution has been deployed. In BeyondCore, you can schedule analysis to continually add new data to your model incrementally. It is also common to periodically update your model variables and features with new information or more focused business questions.

In this white paper, we have shared high level guidelines to help jump start your predictive data preparation foundation. If you want more guidance for your specific project, please reach out to your BeyondCore Customer Success team. We have also provided a list of additional resources for continued learning on this topic.
Additional Resources

*Applied Predictive Analytics: Principles and Techniques for the Professional Data* by Dean Abbott

*Data Preparation for Data Mining* by Dorian Pyle

*Moneyball: The Art of Winning an Unfair Game* by Michael Lewis

*Handbook of Statistical Analysis and Data Mining* by Robert Nisbet, Gary Miner and John Elder


The Probability Cheatsheet [http://www.wzchen.com/probability-cheatsheet](http://www.wzchen.com/probability-cheatsheet)
Glossary

**Binning**

Dividing a numeric continuous variable into categories such as high, medium and low. Data binning or bucketing is a data pre-processing technique used to reduce the effects of minor observation errors.

**Central Tendency**

In statistics, a central tendency is a central or typical value for a distribution. Measures of central tendency are often called averages. The most common measures of central tendency are the mean, median and mode.

**Collinear**

Collinearity occurs when two or more predictor variables are highly correlated, meaning that one can be linearly predicted from the others with a substantial degree of accuracy.

**Correlation**

Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together. A positive correlation indicates the extent to which those variables increase or decrease in parallel; a negative correlation indicates the extent to which one variable increases as the other decreases. The formula for Correlation is Covariance(X,Y) / SQRT( Var(X)* Var(Y)).

**Dependent Variable (Outcome Measure)**

The dependent variable is simply that, a variable that is dependent on an independent variable(s).
Dimensional Data Warehouse

A Dimensional Data Warehouse that is designed for analytical purposes and includes current along with prior states of a record. It is organized by subject (products, customers, patients, date, type) and is time-variant for decision-making processes.

Distribution

A distribution is an arrangement of values of a variable showing their observed or theoretical frequency of occurrence.

Feature

Analytical feature creation is the art of extracting more information from existing data by creating new columns or derived variables. Examples include “Is order committed? Yes/No”, as well as more complex features like a “Credit Score”.

Frequency Table

In statistics, a frequency table is a table that displays the count of various values in a data set.

Granularity

Granularity refers to a unit of analysis. That unit might be an opportunity, customer or a transaction with variables/columns containing descriptive data. The level of detail considered in a model or decision making process.

High Cardinality

High-cardinality fields are categorical attributes that contain a very large number of distinct values. Examples include names, ZIP codes or account numbers. Although these variables could be highly informative, high-cardinality attributes are rarely used in predictive modeling.

Independent Variable (Predictor, Influencer)

An independent variable, sometimes called an experimental or predictor variable, is a variable that is being manipulated in an experiment in order to observe the effect on a dependent variable, sometimes called an outcome variable.
K-fold Cross Validation

Cross-validation is a technique to evaluate predictive models by partitioning the original data set into a various numbers of training sets to train the model, and a test set to evaluate it. In k-fold cross-validation, the data is randomly partitioned into k equal size data sets.

Machine Learning

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.

Mean

The mean or average is the sum of the values divided by the number of items.

Median

In statistics, a number that separates the lowest- and highest-value halves.

Mode

The most frequently occurring value.

Ordinal

Ordinal data consists of numerical scores on an arbitrary scale that is designed to show ranking in a set of data points. For example, Low, Medium and High are ordinal.

Outcome Measure (Independent Variable)

The dependent variable is simply that, a variable that is dependent on an independent variable(s).

Outliers

Outliers are values that exceed three standard deviations from the mean. Regression algorithms are sensitive to outliers since those values affect averages (means) and standard deviations in statistical significance calculations.

Overfitting
Overfitting is a common beginner predictive modeling mistake that happens when too many variable fields are used in a predictive model. Overfitting captures the noise in your data with an overly complex, unreliable predictive model.

**Principal Component Analysis (PCA)**

Principal component analysis is a statistical technique that is used to analyze the interrelationships among a large number of variables and to explain these variables in terms of a smaller number of variables, called principal components, with a minimum loss of information.

**Primary Key**

A primary key is a special relational database table column or combination of columns designated to uniquely identify all table records. It must contain a unique value for each row of data. It cannot contain null or empty values.

**Skewness**

In probability theory and statistics, skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive or negative, or even undefined.

**Standard Deviation**

Standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of data values.

**Statistically Significant**

Statistical significance refers to whether any differences observed between groups being studied are "real" or whether they are simply due to chance.

**Tail**

The tail basically refers to the part of the distribution that is far away from the mean.

**Transactional Database**

A transactional database is defined for day-to-day operations like insert, delete and updates from an application. It is different from a Dimensional Data Warehouse that is designed for analytical purposes and includes current along with prior states of a record.
Unbiased

Fair and impartial rather than biased or prejudiced, in statistics, with an expected value that is equal to the parameter being estimated, fair in the way that you describe or treat a situation.

Underfitting

Underfitting is often a result of an excessively simple model. Underfitting occurs when a statistical algorithm cannot capture the underlying patterns in the data. Thus there is a delicate balance needed between being too specific with too many variables and too vague with not enough selected variables for an outcome.

Variable

An element, feature, or factor that is liable to vary or change.