DATA PREPARATION FOR AUTOMATED MACHINE LEARNING

BY JEN UNDERWOOD
# TABLE OF CONTENTS

- Introduction .................................................................................................................. 3
  - Art and Science of Data Preparation ........................................................................... 3
  - Automation for Faster Data Preparation ................................................................. 3
  - DataRobot for Machine Learning .............................................................................. 4
- Where to Start ................................................................................................................. 5
  - Machine Learning Lifecycle ....................................................................................... 5
  - Plan for Data Collection ............................................................................................. 5
  - Avoid Overfitting and Underfitting ............................................................................. 7
- Collect and Structure Data .............................................................................................. 8
  - Gather Data ................................................................................................................ 8
  - Beware of Bias ........................................................................................................... 11
- Explore and Profile ......................................................................................................... 11
  - Understand Your Data ............................................................................................... 12
  - Detect Leakage ......................................................................................................... 12
  - Find and Reduce Errors ............................................................................................ 13
- Improve Data Quality ..................................................................................................... 15
- Engineer Features ......................................................................................................... 18
- Conclusion ..................................................................................................................... 20
- Recommended Next Steps ............................................................................................. 20

JANUARY 2018 – WHITE PAPER COMMISSIONED BY DATAROBOT
INTRODUCTION

The beauty of the human mind in combination with automated machine learning empowers amazing predictive insights that might never be found using manual techniques. Since the quality of predictive output relies on the quality of input, proper data preparation is a critical success factor for achieving optimal machine learning results.

THE ART AND SCIENCE OF DATA PREPARATION

The iterative process of preparing data for automated machine learning is both an art and a science. The art of data preparation requires knowledge of the business – or “domain expertise” – to select the right problems to solve, identify crucial input data, and carefully transform and engineer informative features to maximize predictive model accuracy. The science of data preparation involves cleansing and normalizing collected data, selecting influencer features, and generating training, testing, and validation data sets for automated machine learning.

Although automated machine learning solutions may provide safeguards to prevent common mistakes, you’ll still want to learn how to correctly prepare, shape, and format your data to create great models. Providing the wrong data, irrelevant data, or improperly prepared data undermines model performance for generalization. Essentially you will want to design an input dataset with feature independence, explainable variance, and maximum information gain to find signals in the noise.

AUTOMATION FOR FASTER DATA PREPARATION

To identify relationships in data — “the signals”— and isolate distracting, irrelevant data — “the noise” — you can expedite the highly iterative predictive data preparation process with automated machine learning. In minutes, you can find the most relevant features and pinpoint specific areas in your dataset where prediction errors occur to help you focus efforts on the right data and reduce experimentation time.

After running basic input data and evaluating the results, you will enhance input data, add features, build another model, and review performance once again. You’ll continue this process until your model meets performance objectives.

Ideally subject matter experts that understand the business process and data source nuances will assist in the data preparation process. 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 experiment further.
DATAROBOT FOR MACHINE LEARNING

DataRobot is the world’s most advanced automated machine learning platform. DataRobot automates the machine learning process from data ingestion to deployment. It delivers immediate value and unmatched ease-of-use, and no complicated math or scripting is required.

DataRobot includes an array of data preparation features, automating feature engineering to find key insights and hidden patterns. This invaluable technology expedites analytical investigation across millions of variable combinations that would be far too time-consuming for manual human exploration.

For optimal machine learning model performance, domain knowledge and best practices used by the world’s leading data scientists have been uniquely baked into DataRobot blueprints. Users of all skill levels can safely apply machine learning with its built-in optimizations and safeguards.

DataRobot supports popular advanced machine learning techniques and open source tools such as Apache Spark, H2O, Scala, Python, R, and TensorFlow. Using drag-and-drop, point-and-click guided menu options, DataRobot users can simply and quickly create predictive models with automated machine learning. The process is simple:

- Ingest data sources
- Select a target variable to predict
- Automatically generate features, extract balanced samples, build and iterate through 100s of machine learning models
- Visually explore top performing models and key findings
- Easily deploy and operationalize models

Machine learning development steps that used to take weeks or months of effort can now be completed in hours. By embedding DataRobot automated machine learning model intelligence into your reporting or business processes, you can quickly close the loop between insight and action.

Since each data set and business objective can be unique with varied challenges, we have provided the following guidelines to help get you started. We also share essential tips and additional resources for further study.

“Domain knowledge and best practices used by the world’s leading data scientists have been uniquely baked into DataRobot blueprints.”
WHERE TO START

The machine learning process begins with Business Understanding. This initial step focuses on defining the right problem to solve and recognizing the business objectives and requirements. After selecting a problem, you will collect and assess data. During the Data Understanding step, you will get familiar with available data sources, identify data quality problems, and perform exploratory analysis. Then, in the Data Preparation step, you will cleanse the data, shaping and transforming it into a flattened format for loading into the automated machine learning platform.

MACHINE LEARNING LIFECYCLE

For the purposes of this white paper, we will concentrate on collecting data and preparing it properly. We will not cover the entire machine learning lifecycle.

Before you begin the data collection and data preparation process, it is assumed that you already have selected, defined, and isolated a business problem to solve that is a viable candidate for machine learning. You should also have chosen at least one metric that you want to better understand. If you need more information on those steps in the machine learning process, please refer to our previous white paper, Moving from BI to Machine Learning.

PLAN FOR DATA COLLECTION

As you develop requirements for machine learning model data collection, contemplate the business process and review it from different perspectives. Consider what happens at each step, what data is captured, where it is stored, if data history or changes are retained, and if that data truly reflects the real world for resulting predictions. Often data is collected in line-of-business applications or data warehouses for other reasons.
data may be missing situational context such as location, environmental conditions, and other relevant variables for predicting an outcome. Document known issues and preferred data that could be added in the future.

**TO IMPROVE INPUT DATA COLLECTION, DIAGRAM THE BUSINESS PROCESS FLOW.** IDENTIFY THE STEPS, ENVIRONMENTAL CONDITIONS, SCENARIOS, PEOPLE, SYSTEMS, AVAILABLE DATA, AND MISSING DATA.

As you continue planning to gather data for your machine learning modeling project, you’ll need to confirm decision-level metric granularity. Granularity refers to a unit of analysis. A unit might be an opportunity, customer, or transaction. Granularity is determined by the business objectives and how your model will be used operationally. Ask stakeholders how decisions will be made from the predictive models. Are they based on a single customer, transaction, or event, or are they based on aggregate data over time?

To illustrate these concepts, we will be referring to a publicly available Bank Marketing Data Set from UCI’s machine learning repository. The sample data set contains partially prepared data to predict client term deposits collected during the bank’s telemarketing campaigns.

In the Bank Marketing Data Set, the desired outcome to predict is client term deposits. This is a binary yes or no outcome in the sample, but it could have alternatively been a total amount figure to maximize deposits. Don’t always limit yourself to collecting one outcome variable while assembling data. Think about other questions that might be asked and data that would make sense to include.

Potential influencer features for the example client term deposit outcome include client demographics such as age, job, marital status, and education. Past credit and loan repayment information is also important to know. Other features chosen included campaign contacts, previous marketing campaign outcomes, and several external social and economic environmental attributes such as employment rate.

**NOTE: WHEN SELECTING FEATURES, YOU WILL WANT TO EXTRACT THE MAXIMUM INFORMATION FROM THE MINIMUM NUMBER OF INPUT VARIABLES.**

---

1 UCI machine learning repository data set [https://archive.ics.uci.edu/ml/datasets/bank+marketing](https://archive.ics.uci.edu/ml/datasets/bank+marketing)
AVOID OVERFITTING AND UNDERFITTING

Overfitting and underfitting are common mistakes for beginners who are preparing machine learning modeling data. 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.

If you don’t have enough features, your model might be oversimplified and suffer from underfitting issues. Underfitting occurs when a statistical algorithm cannot capture the underlying patterns in the data.

Why are overfitting and underfitting problematic? The machine learning model will have too many prediction errors to be useful for decision-making.

For categorical data, overfitting can occur if a high number of categories are observed with a small number of observations per category. These types of variables hold less information for predictive value. For time series data, overly complex mathematical functions that describe the relationship between the input variable and the target variable can also lead to overfitting. In the most extreme form of overfitting, individual identifiers are inadvertently used as machine learning inputs. Individual identifiers can perfectly model existing data, but would only by chance reliably model and predict outcomes for other data.

Thus, there is a delicate balance between being too specific with too many features and too vague with not enough features. Designing machine learning model features with just the right amount of predictive information gain and precision is a key skill in the art of data preparation.
COLLECT AND STRUCTURE DATA

After reviewing the business process and planning the machine learning input data requirements, you’ll delve into the data collection and shaping process.

GATHER DATA

Machine learning 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 in a column within the row of data to capture that behavior. Unlike third-normal form transactional or dimensional patterns used in business intelligence, machine learning requires data to be input as a “flattened” table, view, or comma separated (.csv) flat file of rows and columns. Your view will need to contain an outcome metric and target variable, along with input predictor variables. This data representation for machine learning is called the feature matrix.

If you have data stored in several tables in a data warehouse or relational database format, you will need to use record identifiers to join fields from multiple tables to create a single unified, flattened “view.” For many target variables, input 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/or 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.

Prepared data should be collected at a level of analytical granularity upon which you can make decisions. Choose a granularity that is actionable, understandable, and useful in the
event you incorporate the results into your existing business process or application. For example, if you want to make daily sales forecasts, you need to input data at a day level rather than week, month, or year.

If you are trying to capture changes in data over a certain 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 such as Salesforce only contain the current state value for a record. If you want to get a prior value, you need to have a snapshot of the historical data stored or keep the prior value data in custom fields on the current record.

While structuring input data, ensure that it is clean and consistent. The order and meaning of input variables should remain the same from record to record. Inconsistent data formats, “dirty data,” and outliers can undermine the quality of analytical findings.

**HOW MUCH DATA TO COLLECT**

The actual number of records is not always easy to determine and depends on patterns 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 4: Sample Size Estimation*

To determine minimum data set sizes, consider the dimensionality of your data and pattern complexity.²

- For small 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).

**TRAINING, TESTING, AND VALIDATION DATA SETS**

The most common strategy is to split data into training, testing, and validation data sets.

---

² Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst by Dean Abbott
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 should reflect your real-world scenario.

**Cross Validation**

Keep in mind that DataRobot includes industry standard k-fold cross validation features that divide 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 combined to form a training set. DataRobot’s k-fold feature enables you to independently choose how much data you want to use in testing.

**Time Series Considerations**

Data that changes over time should be reflected in your input data set. When time sequences (Contact Made > Quote Provided > Deal Closed) are important in predictions, proportionally collecting data from those different time periods is important as well. The key principle is to provide data that reflects what actually happens at the right level of outcome metric granularity.

![Sampling Data](image)

When collecting data, think about the balance of your values in your raw data. In our example, how many prospects do you have at different ages and stages of life? Do they have loans? What are the balances on outstanding loans? How many defaulted on a loan? How recent was the default? What is the prospect’s income history, credit ratings, debt-to-income ratio, and so on?

**Think Proportionally**

When extracting a subset of data, be sure to include approximately the same proportion of variables in your DataRobot input data set that you see in the real-world data. If you provide more records of one variable value, you can accidentally bias the machine learning model’s predictions, thereby diminishing performance. If you have data sets with millions or billions of rows, it is much less likely that you will encounter accidental bias in data preparation.
SAMPLING

The choice of the optimal sampling method\textsuperscript{3} for a given problem especially depends on the character of the dataset and the desired proportion of the subsets. Each method has its advantages but also its limitations.

For simple, nearly uniformly distributed datasets, the method of simple random sampling may be sufficient. For naturally well-ordered time series data, highly efficient deterministic approaches such as convenience and systematic sampling can achieve reliable results. When dealing with complex, high-dimensional data, more sophisticated and stratified sampling techniques can reduce the bias and variance of model error.

Unbalanced Two-Class Problems

Data sets seldom come with evenly distributed samples. Unbalanced data is a common issue to remediate. Fraud or failure rate data are examples of unbalanced two-class problems. Analyzing unbalanced data creates useless results with exceptionally high error rates.

To build predictive models on unbalanced data, you need to apply sampling techniques that increase the minority class proportion with downsampling or upsampling to create a balanced data set. After training a machine learning model with a balanced training set, you will validate performance of the model with the real-world, unbalanced, unseen test set.

BEWARE OF BIAS

While you accumulate data, consider potential biases.\textsuperscript{4} Human nature is consciously and unconsciously biased. Cognitive biases are tendencies to think in certain ways that can lead to irrational judgment. Outcome, omission, and many other bias types can easily creep into the data collection process.

If unknown bias exists, it is basically an unjustified assumption that your input data reflects reality. Any model built on such assumptions reflects only the distorted reality and will perform poorly. To reduce potential bias, test hypotheses, poke holes in your own ideas, welcome challenges, and conduct peer reviews of your data collection and sampling thought processes. Machine learning projects should be group projects and not done in isolation.

EXPLORE AND PROFILE

Now you will assess the condition of your source data. DataRobot automates several aspects of initial data examination. DataRobot also automates sampling to avoid conventional sampling and overfitting issues. During this step, you’ll visually look for

\textsuperscript{3} Common Types of Data Sampling Methods https://en.wikipedia.org/wiki/Sampling

\textsuperscript{4} The Cognitive Bias Codex https://upload.wikimedia.org/wikipedia/commons/a/a4/The_Cognitive_Bias_Codex_-_180%28_biases%29C_designed_by_John_Manooian_III_%28jm3%29.png
trends, extreme values, outliers, exceptions, skewed data, incorrect values, and inconsistent and missing data.

**UNDERSTAND YOUR DATA**

As you begin to explore and understand your data, DataRobot provides data profiles for every feature that include how many values are unique or missing, as well as the statistical mean, standard deviation, median, minimum, and maximum value. You can also review the distributions of each feature using a histogram with optional bin settings and apply transformations to normalize your data.

Informative Features immediately ranks variables that provide the most information gain for building optimal machine learning models. Knowing which areas of your data most influence the outcome is invaluable on its own. This information can guide the business to focus limited time and resources on the activities that matter most.

Another one of DataRobot’s data preparation strengths is the array of different visualization tools that identify influential features, rank them, and uncover errors. The Feature Ranking report measures how much each feature by itself contributes to the accuracy of a machine learning model.

![Feature Impact](image)

*Figure 6: Feature impact*

**DETECT LEAKAGE**

Leakage is the accidental inclusion of outcome information that would not be legitimately available for predictions. If you have inadvertent leakage, you’ll likely notice it in the feature ranking report when a feature has an exceptionally high impact.
In our Bank Marketing Data Set example, duration was identified in the DataRobot Feature Impact report as a leaked feature with 100% impact. The next best performing feature carries a little more than 10% impact. If you see an exceptionally high impact, verify process flow timing for that feature.

TO AVOID LEAKAGE, YOU NEED TO CONSIDER THE TIMING AND ORDER OF EVENTS TO ENSURE THAT OUTCOME DATA IS NOT USED AS INPUT DATA.

FIND AND REDUCE ERRORS

DataRobot Model X-Ray and Reason Codes capabilities can also provide deep insights for enhancing data preparation. Model X-Ray enables interactive, visual exploration of machine learning model performance. You can easily see where a model makes mistakes by selecting input features. It shines a light on issues that might not get detected using other tools. Model X-Ray allows machine learning model designers to concentrate on where the most model performance improvements can be made in the data preparation process.

Figure 7: Leakage Example
Reason Codes unveil what values within a feature drive the model’s results. This tool is crucial for machine learning model development processes that are subject to regulatory compliance or legal scrutiny. With Reason Codes, you can discover which combinations of feature values trigger a specific machine learning outcome. This information can also be useful throughout the iterative data preparation process to incrementally improve results.
IMPROVE DATA QUALITY

We recommend that you address data quality issues as early as possible. Here are a few tips for handling common data issues in the data preparation process.

Correcting Incorrect Values

Machine learning models assume the input data is correct. 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.

Treat incorrect values as missing if there is a minimal amount and you can’t easily determine correct values. If there are a lot of inaccurate values, try to determine what happened to repair them. If you do make changes to data, document your reasoning. Also capture initial context and changed values with a flag to identify changes. The pattern in your data might be hidden in those incorrect values.

Skewed Variables

For continuous variables, review the distributions, central tendency, and variable spread in DataRobot. These are measured using various statistical metrics and visualization methods such as histograms. Continuous variables should be normally distributed. If not, reduce skewness with transformations or by experimenting with bin sizes for optimal prediction.

When a skewed distribution needs to be corrected, the variable is transformed by a function that has a disproportionate effect on the tails of the distribution. Log transforms like log(x), logn(x), log10(x); the multiplicative inverse (1/x); square root transform sqrt(x); or power (xn) are the most frequently used corrective functions.

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>POTENTIAL FIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive skew</td>
<td>log10(1+x), 1/x, sqrt(x)</td>
</tr>
<tr>
<td>Negative skew</td>
<td>x^n -log10(1+abs(x))</td>
</tr>
<tr>
<td>Long tails</td>
<td>sgn(x) x log10(1+abs(x))</td>
</tr>
</tbody>
</table>

In the table to the left, several issues and formulas to minimize skew are shown. The before and after charts illustrate how different skewed variable transformations can be used to normalize feature distributions.

Figure 10: Transformations for Skewness

© 2017 Impact Analytix, LLC - All rights reserved.
High-Cardinality

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

Redundant, Highly Correlated Variables

Duplicate, redundant, or other highly correlated variables that carry the same information should be minimized. DataRobot algorithms will perform better without 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 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.

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 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 provide obvious insights. For example, customers who live in the city of Tampa also happen to live in the state of Florida.

Figure 11: Scatter plots for correlation detection
Missing values

The most common repair for missing values is imputing a likely or expected value using a mean or computed value from a distribution. If you use the 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. 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.

Extreme Values and Outliers

Outliers are values that exceed three standard deviations from the mean. Many machine learning 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 that 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 log transformations or converting the numeric variable to a categorical value with binning.
ENGINEER FEATURES

Feature creation is the art of extracting more information from existing data to improve the predictive power of machine learning algorithms. 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 higher model accuracy than any single-variable transformation because of the information gain associated with data interactions. If you ever saw the movie or read the book “Moneyball: The Art of Winning an Unfair Game” by Michael Lewis, you’ll know how baseball analysis was revolutionized with new performance metrics like On-Base Percentage (OBP) and Slugging Percentage (SLG). With feature engineering, you will be using a fundamentally similar approach.

HUMAN INGENUITY AND CREATIVITY REQUIRED

Feature engineering is challenging because it depends on leveraging human intuition to interpret implicit signals in data sets that machine learning algorithms use. Consequently, feature engineering is often the determining factor in whether a machine learning modeling project is successful or not. This step in the process is experimental and usually the bottleneck in automated machine learning processes.

Although collected raw data fields can be used as-is in DataRobot without transformations, supplemental data, or calculations to train machine learning models, you’ll almost always want to add more perspective to your data set by designing features. Engineered features provide better context to differentiate patterns in the data.

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. Note DataRobot automatically generates date and time aggregation features.

Ratios

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.

Transformations

Transformation refers to the replacement of a variable by a function. For instance, replacing a variable x by the square or 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 Loan Amount into Loan Amount Categories.

FEATURE ENGINEERING TECHNIQUES
Here are a few popular feature engineering ideas shared on Data Science Central⁵ that can be used to extract more information from your input data:

1. Single variable transformations
2. Ratio or frequency of categorical variables
3. Combine important variables
4. Compute variable interactions
5. Change data types
6. Compute relative differences
7. Cartesian transformations
8. Bin transformations
9. Window time series data
10. Reframe continuous variables
11. One hot encoding for sequence problems
12. Sparse value coding

The possibilities for feature engineering are limited only by your own human ingenuity and creativity. Feature engineering truly is the human art of data preparation for automated machine learning.

⁵ Data Science Central https://www.datasciencecentral.com/profiles/blogs/feature-engineering-data-scientist-s-secret-sauce-1
CONCLUSION

In this white paper, we briefly introduced basic data preparation for machine learning concepts. We discussed how to plan your project and collect, organize, structure, and shape data in a machine learning-friendly format. We also bestowed vital tips for feature engineering to help you master the art of data preparation for automated machine learning models.

As you progress in your automated machine learning model journey, each area of the data preparation process should be further researched. For additional reading, the following books are highly recommended:

- **Data Preparation for Data Mining** by Dorian Pyle
- **Data Preprocessing in Data Mining** by Salvador García and Julián Luengo
- **Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst** by Dean Abbott
- **Feature Engineering for Machine Learning Models: Principles and Techniques for Data Scientists** by Alice Zheng and Amanda Casari

Note that DataRobot also provides classes that cover these and other related topics.

RECOMMENDED NEXT STEPS

For additional information on automated machine learning, please contact an expert at DataRobot. It's easy to get started.

- **DataRobot**
  - [www.datarobot.com](http://www.datarobot.com)
- **Data Preparation Essentials for Automated Machine Learning** webinar recording
- **DataRobot AI Acceleration Packages**
- **DataRobot Courses**
  - [www.datarobot.com/education/all-courses/](http://www.datarobot.com/education/all-courses/)
About DataRobot
DataRobot offers an enterprise machine learning platform that empowers users of all skill levels to make better predictions faster. Incorporating a library of hundreds of the most powerful open source machine learning algorithms, the DataRobot platform automates, trains and evaluates predictive models in parallel, delivering more accurate predictions at scale. DataRobot provides the fastest path to data science success for organizations of all sizes. For more information, visit www.datarobot.com.

About the Author
Jen Underwood, founder of Impact Analytix, LLC, is an analytics industry expert with a unique blend of product management, design and over 20 years of “hands-on” advanced analytics development. In addition to keeping a pulse on industry trends, she enjoys digging into oceans of data. Jen is honored to be an IBM Analytics Insider, SAS contributor, and former Tableau Zen Master. She also writes for InformationWeek, O’Reilly Media, and other tech industry publications.

Jen has a Bachelor of Business Administration – Marketing, Cum Laude from the University of Wisconsin, Milwaukee and a post-graduate certificate in Computer Science – Data Mining from the University of California, San Diego. For more information, visit www.jenunderwood.com.