



Machine Learning with **SAS[®] Viya[®]**



SAS Institute Inc.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2020. *Machine Learning Using SAS® Viya®*. Cary, NC: SAS Institute Inc.

Machine Learning Using SAS® Viya®

Copyright © 2020, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-951685-39-3 (Hard cover)

ISBN 978-1-951685-30-0 (Paperback)

ISBN 978-1-951685-31-7 (PDF)

ISBN 978-1-951685-37-9 (EPUB)

ISBN 978-1-951685-38-6 (Kindle)

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

May 2020

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Contents

About this Book	vii
Acknowledgments	xi
Preface	xiii
Chapter 1: Introduction to Machine Learning.....	1
Introduction.....	1
Supervised Learning Predictions	2
Model Building and Selection	5
Introducing Model Studio.....	5
Quiz.....	23
Chapter 2: Preparing Your Data: Introduction	25
Introduction.....	25
Explore the Data.....	25
Divide the Data.....	34
Address Rare Events	35
Data Preparation Best Practices	44
Quiz.....	54
Chapter 3: Preparing Your Data: Missing and Unstructured Data	55
Introduction.....	55
Dealing with Missing Data	55
Add Unstructured Data.....	72
Quiz.....	82
Chapter 4: Preparing Your Data: Extract Features	83
Introduction.....	83
Extract Features	83
Handling Extreme or Unusual Values.....	88
Feature Selection	95
Quiz.....	112
Chapter 5: Discovery: Selecting an Algorithm	113
Introduction.....	113
Select an Algorithm	114
Classification and Regression	118
Quiz.....	127

Chapter 6: Decision Trees: Introduction	129
Introduction	129
Decision Tree Algorithm	129
Building a Decision Tree	131
Pros and Cons of Decision Trees	141
Quiz	141
Chapter 7: Decision Trees: Improving the Model	143
Introduction	143
Improving a Decision Tree Model by Changing the Tree Structure Parameters	143
Improving a Decision Tree Model by Changing the Recursive Partitioning Parameters	146
Optimizing the Complexity of the Model	155
Regularize and Tune Hyperparameters	160
Quiz	166
Chapter 8: Decision Trees: Ensembles and Forests	167
Introduction	167
Building Ensemble Models: Ensembles of Trees	168
Building Forests	171
Gradient Boosting with Decision Trees	175
Pros and Cons of Tree Ensembles	188
Quiz	188
Chapter 9: Neural Networks: Introduction and Model Architecture	189
Introduction	189
The Neural Network Model	190
Improving the Model	198
Modifying Network Architecture	198
Strengths, Weaknesses, and Parameters of Neural Networks	209
Quiz	213
Chapter 10: Neural Networks: Optimizing the Model and Learning	215
Optimizing the Model	215
Regularize and Tune Model Hyperparameters	235
Quiz	241
Chapter 11: Support Vector Machines	243
Introduction	243
Support Vector Machine Algorithm	244
Improve the Model and Optimizing Complexity	252
Model Interpretability	260
Regularize and Tune Hyperparameters of the Model	272
Quiz	274

Chapter 12: Model Assessment and Deployment.....	275
Introduction.....	275
Model Assessment.....	276
Model Deployment	302
Monitoring and Updating the Model.....	314
Quiz.....	315
Chapter 13: Additional Model Manager Tools and Open-Source Code	317
Introduction.....	317
Appendix A	335
A.1: CAS-Supported Data Types and Loading Data into CAS	335
A.2: Rank of a Matrix.....	338
A.3: Impurity Reduction Measures	339
A.4: Decision Tree Split Search.....	341
Appendix B: Solutions.....	347
Practice Solutions.....	347
Quiz Solutions	355
References	357

About This Book

What Is This Book About?

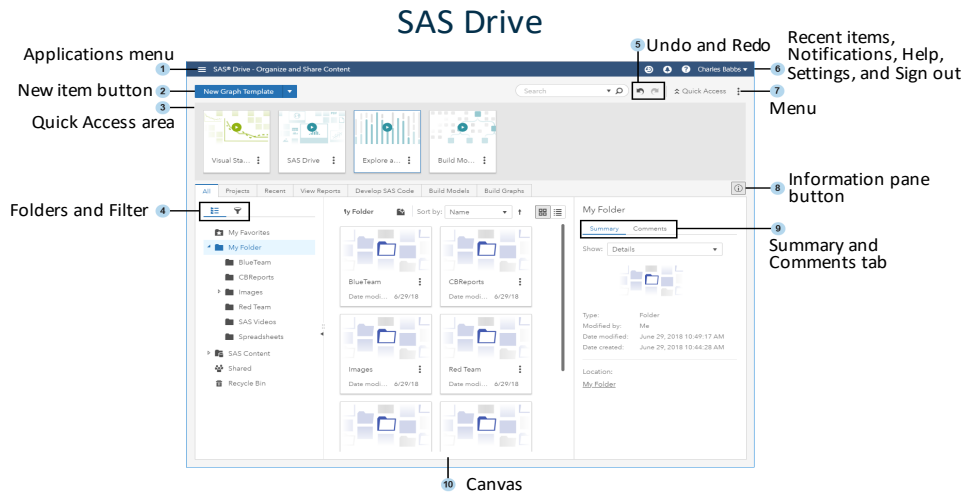
The focus of this book is to explore data using SAS® Viya®—the latest extension of the SAS Platform—to build, validate, and deploy models into production to augment business decision making. We call this the analytics life cycle. This is at the heart of the SAS Platform, and it is a series of phases: **Data**, **Discovery**, **Deployment**, with the goal to extract value from raw data.

Analytics Life Cycle



SAS Drive is a common interface for the SAS Viya applications that supports all three phases of the analytics life cycle. It enables you to view, organize, and share your content from one place.

Screen Shot of SAS Drive



SAS Drive is available from the Applications menu in the upper left. The displayed tabs depend on the products that are installed at your site. This book focuses on the Build Models action that launches Model Studio pipelines.

What Is Required to Create Good Machine Learning Systems?

In most business problems, you need to go from data to decisions as quickly as possible. Machine learning models are at the heart of critical business decisions. They can identify new opportunities and enable you to manage uncertainty and risks. To create these models, you need to wrangle your data into shape and quickly create many accurate predictive models. You also need to manage your analytical models for optimal performance throughout their lifespan. All good machine learning systems need to consider the following:

- Data preparation
- Algorithms
- Automation and iterative processes
- Scalability
- Ensemble modeling

In this book, we will illustrate each of these processes and how to do them using SAS Model Studio. We will also present just enough theory so that you can understand the techniques and algorithms used enough to be able to choose the correct model for each business problem and fine-tune the models in an efficient and insightful way.

Is This Book for You?

Building representative machine learning models that generalize well on new data requires careful consideration of both the data used for the model to train, and the assumptions about the various training algorithms. It is important to choose the right algorithm for both the data that you will be modeling and the business problem that you are trying to solve.

SAS graphical user interfaces help you build machine learning models and implement an iterative machine learning process. You don't have to be an advanced statistician. The comprehensive selection of machine learning algorithms can help you quickly get value from your big data and are included in many SAS products.

What Should You Know about the Examples?

This book includes worked demonstrations and practices for you to follow to gain hands-on experience with SAS Model Studio.

Software Used to Develop the Book's Content

Model Studio is included in SAS Viya. It is an integrated visual environment that provides a suite of analytic data mining tools that enable you to explore and build models. It is part of the **Discovery** phase of the analytic life cycle. The data mining tools provided in Model Studio enable you to deliver and distribute analytic model data mining champion models, score code, and results. Model Studio contains the following SAS solutions:

- SAS Visual Forecasting
- SAS Visual Data Mining and Machine Learning
- SAS Visual Text Analytics

The visual analytic data mining tools that appear in Model Studio are determined by your site's licensing agreement. Model Studio operates with one, two, or all three of the web-based analytic tools as components of the software.

Model Studio comes with SAS Data Preparation. SAS Data Preparation is a software offering that adds data quality transformations and other advanced features. There are several options that enable you to perform specific data preparation tasks for applications, such as SAS Environment Manager, SAS Visual Analytics, Model Studio, and SAS Decision Manager. You can perform some of the basic data preparation tasks through Model Studio, as we will describe in this book.

Example Code and Data

The data sets used in the book's demonstrations and practices are provided to download.

You can access the example code and data for this book by linking to its author page at support.sas.com/sasinstitute.

We Want to Hear from You

SAS Press books are written *by* SAS Users *for* SAS Users. We welcome your participation in their development and your feedback on SAS Press books that you are using. Please visit sas.com/books to do the following:

- Sign up to review a book
- Recommend a topic
- Request information on how to become a SAS Press author
- Provide feedback on a book

Do you have questions about a SAS Press book that you are reading? Contact the author through saspress@sas.com or https://support.sas.com/author_feedback.

SAS has many resources to help you find answers and expand your knowledge. If you need additional help, see our list of resources: sas.com/books.

Acknowledgments

This book is based on the SAS training course, *Machine Learning Using SAS® Viya®*, developed by Carlos Pinheiro, Andy Ravenna, Sharad Saxena, Jeff Thompson, Marya Ilgen-Lieth, and Cat Truxillo. Additional content and editing was made by Sian Roberts. Design, editing, and production support was provided by the SAS Press team: Robert Harris, Lauree Shepard, Suzanne Morgen, and Denise Jones.

Preface

What Is Machine Learning?

Machine learning is a branch of artificial intelligence (AI) that automates the building of models that learn from data, identify patterns, and predict future results—with minimal human intervention.

Machine learning is not all science fiction. Common examples in use today include self-driving cars, online recommenders such as movies that you might like on Netflix or products from Amazon, sentiment detection on Twitter, or real-time credit card fraud detection.

Statistical Modeling Versus Machine Learning

Just like statistical models, the goal of machine learning is to understand the structure of the data. In statistics, you fit theoretical distributions to the data that are well understood. So, with statistical models there is a theory behind the model that is mathematically proven, but this requires that data meets certain strong assumptions too. Machine learning has developed based on the ability to use computers to probe the data for structure without having a theory of what that structure looks like. The test for a machine learning model is a validation error on new data, not a theoretical test that proves a null hypothesis. Because machine learning often uses an iterative approach to learn from data, the learning can be easily automated. Passes are run through the data until a robust pattern is found.

Algorithms

Building representative machine learning models that generalize well on new data requires careful consideration of both the data used for the model to train and the assumptions about the various training algorithms. It is important to choose the right algorithm for both the data that you will be modeling and the business problem that you are trying to solve. For example, if you are building a model to detect tumors, then it would be important to choose a model with a high accuracy, as it would be more important not to miss any possible tumors. On the other hand, if you were looking to build a model to predict who best to send an offer to in a marketing campaign with a limited budget, you would want the model that is best at predicting rank, or the top 100 or so customers most likely to use the offer. In Chapter 2, we discuss different measures of model performance and when they should be used in more detail.

While many machine learning algorithms have been around for a long time, advances in computer power and parallel processing have allowed the ability to automatically apply complex mathematical calculations to big data faster and faster, making them a lot more useful.

Most industries working with large amounts of data recognize the value in machine learning technology to gain insights and automate decisioning. Common application areas include:

- Fraud
- Targeted Marketing
- Financial Risk
- Churn

Fraud

Fraud detection methods attempt to detect or impede illegal activity that involves financial transactions. Anomaly detection is one of the ways to detect fraud. You look to predict an event that occurs rarely and identify patterns in the data that do not conform to expected behavior, such as an abnormally high purchase made on a credit card.

Targeted Marketing

Targeted marketing is another common application area. Most companies rely on some form of direct marketing to acquire new customers and generate additional revenue from existing customers. Predictive modeling generally accomplishes this by helping companies answer crucial questions such as: Who should I contact? What should I offer? When should I make the offer? How should I make the offer?

Financial Risk

Financial risk management models attempt to predict monetary events such as credit default, loan prepayment, and insurance claim. Banks use multiple models to meet a variety of regulations (such as CCAR and Basel III). With increased scrutiny on model risk, bankers must establish a model risk management program for regulatory compliance and business benefits. Models are useful things to have around, and bankers have come to rely on them for certain applications, some of which expose the bank to significant risks. Predictive models fall into this category. Examples include loan approval using credit scoring and hedging models using swaps and options to manage the balance sheet while protecting liquidity and determining capital adequacy.

Churn

Customer churn is one of the main problems in many businesses. Churn or attrition is the turnover of customers of a product or users of a service. Studies have shown that attracting new customers is much more expensive than retaining existing ones. Consequently, companies focus on developing accurate and reliable predictive models to identify potential customers who will churn soon.

What Is SAS Viya?

SAS Viya is an open, cloud-enabled, analytic run-time environment with a number of supporting services, including SAS Cloud Analytic Services (CAS). CAS is the in-memory engine on the SAS Platform.

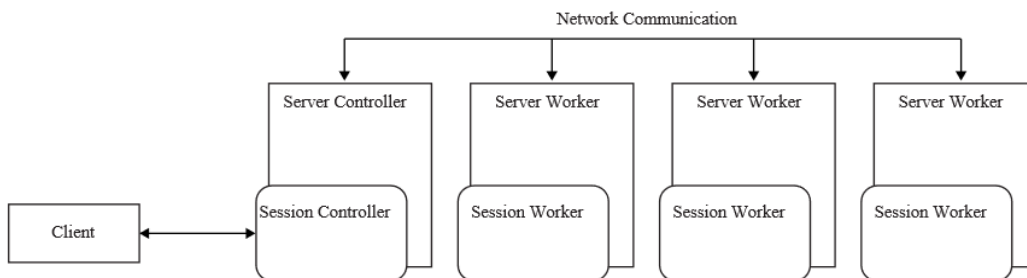
Run-time environment refers to the combination of hardware and software in which data management and analytics occur.

CAS is designed to run in a single-machine symmetric multiprocessing (SMP) or multi-machine massively parallel processing (MPP) configuration. CAS supports multiple platform and infrastructure configurations. CAS also has a communications layer that supports fault tolerance. When CAS is running in an MPP configuration, it can continue processing requests even if it loses connectivity to some nodes. This communication layer also enables you to remove or add nodes while the server is running.

Distributed Server: Massively Parallel Processing (MPP)

A distributed server uses multiple machines to perform massively parallel processing. The figure below depicts the server topology for a distributed server. Of the multiple machines used, one machine acts as the controller and other machines act as workers to process data.

Distributed Server: Massively Parallel Processing (MPP)

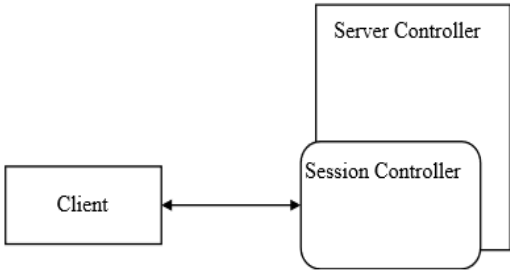


Client applications communicate with the controller, and the controller coordinates the processing that is performed by the worker nodes. One or more machines are designated as worker nodes. Each worker node performs data analysis on the rows of data that are in-memory on the node. The server scales horizontally. If processing times are unacceptably long due to large data volumes, more machines can be added as workers to distribute the workload. Distributed servers are fault tolerant. If communication with a worker node is lost, a surviving worker node uses a redundant copy of the data to complete the data analysis. Whenever possible, distributed servers load data into memory in parallel. This provides the fastest load times.

Single-Machine Server: Symmetric Multiprocessing (SMP)

The figure below depicts the server topology for a single-machine server. The single machine is designated as the controller. Because there are no worker nodes, the controller node performs data analysis on the rows of data that are in-memory. The single machine uses multiple CPUs and threads to speed up data analysis.

Single-Machine Server: Symmetric Multiprocessing (SMP)



This architecture is often referred to as symmetric multi-processing (SMP). All the in-memory analytic features of a distributed server are available to the single-machine server. Single-machine servers cannot load data into memory in parallel from any data source.

Using Cloud Analytic Services (CAS)

Leveraging the CAS server that is part of the SAS Viya release includes a whole host of tangible benefits. The main reason is represented by a simple three-word phrase: tremendous performance gains. Because processes run so much faster, you can complete your work faster. This means that you can complete more work, and even entire projects, in a significantly reduced time frame.

Processing Type	Multi-threaded, Single Machine (SAS Viya SMP)	Multi-threaded, Multiple Machines (SAS Viya MPP)
Distributed, parallel processing?	Yes	Yes
In-memory data persistence?	Yes	Yes
Common performance speed-up	10x–20x	Up to 100x*

* Increase depends on many factors including hardware allocation. Performance could be higher.

See Appendix A.1 for information about working with CAS, CAS-supported data types, and loading data into CAS.

The Mindset Shift

There are some differences that you need to be aware of when working with SAS Viya. In SAS Viya, you might have nondeterministic results or might not get reproducible results, essentially because of two reasons:

- distributed computing environment
- nondeterministic algorithms

In distributed computing, cases are divided over compute nodes, and there could be variation in the results. You might get slightly different results even in the same server when the controllers/workers are more manageable. In different servers, this is even more expectable. A CAS server represents pooled memory and runs code multi-threaded. Multi-threading tends to distribute the same instructions to other available threads for execution, creating many different queues on many different cores using separate allocations or subsets of data. Most of the time, multiple threads perform operations on isolated collections of data that are independent of one another but part of a larger table. For that reason, it is possible to have a counter (for example, $n+1$;) operating on one thread to produce a result that might be different from a counter operating on another thread because each thread is working on a different subset of the data.

Therefore, results can be different from thread to thread unless and until the individual results from multiple threads are summed together. It is not as complicated as it might sound. That is because SAS Viya automatically takes care of most collation and reassembly of processing results, with a few minor exceptions where you must further specify how to combine results from multiple threads.

A nondeterministic algorithm is an algorithm that, even for the same input, can exhibit different behaviors on different runs, as opposed to a deterministic algorithm. There are several ways an algorithm might behave differently from run to run. A concurrent algorithm can perform differently on different runs due to a race condition. A probabilistic algorithm's behaviors depend on a random number generator. The nondeterministic algorithms are often used to find an approximation to a solution when the exact solution would be too costly to obtain using a deterministic one (Wikipedia). Some SAS Visual Data Mining and Machine Learning models are created with a nondeterministic process. This means that you might experience different displayed results when you run a model, save that model, close the model, and re-open the report or print the report later.

Deterministic and Nondeterministic Algorithms

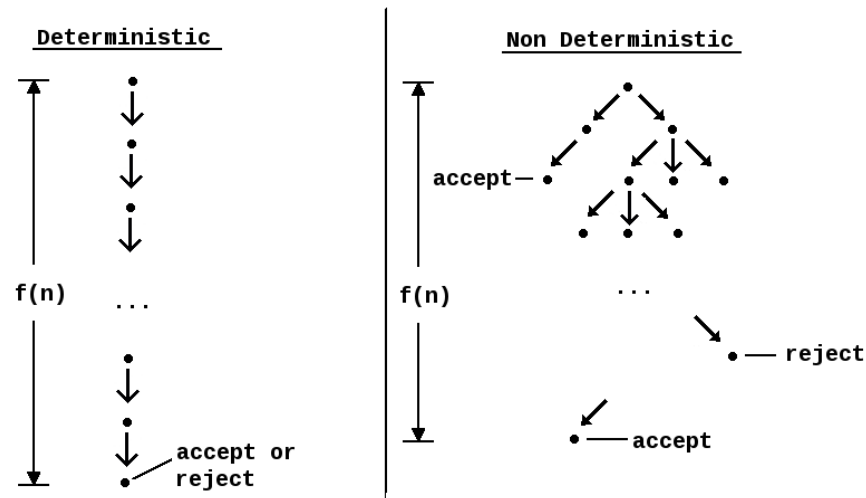


Image source: By Eleschinski2000—With a paint program, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=43528132>

A deterministic algorithm that performs $f(n)$ steps always finishes in $f(n)$ steps and always returns the same result. A nondeterministic algorithm that has $f(n)$ levels might not return the same result on different runs. A nondeterministic algorithm might never finish due to the potentially infinite size of the fixed height tree.

It is an altogether different mindset!

You are “converging” on a model or “estimating” a model, not exactly computing the parameters of the model. Bayesian models understand this when they look for convergence of parameters. They try to converge to a distribution, not a point. Maybe it would be interesting to try running the models 10 times across different samples and ensembling them to see the dominant signal. You cannot expect the results to be reproduced because some algorithms have randomness included in the process. However, the results do converge. This is a distinguished computing environment designed for big data, and this non-reproducibility is the price that we pay.

Note: “Data Science’s Reproducibility Crisis” <https://towardsdatascience.com/data-sciences-reproducibility-crisis-b87792d88513> is an interesting read.

SAS Visual Data Mining and Machine Learning:

A variety of products sit in SAS Viya. They enable users to perform their jobs as part of the analytics life cycle. In this book, you use SAS Visual Data Mining and Machine Learning.

The Model Studio interface is superset of SAS Visual Data Mining and Machine Learning, SAS Visual Forecasting, and SAS Visual Text Analytics.

SAS Visual Data Mining and Machine Learning is a product offering in SAS Viya that contains:

1. underlying CAS actions and SAS procedures for data mining and machine learning applications
2. GUI-based applications for different levels and types of users.

These applications are as follows:

- **Programming interface:** a collection of SAS procedures for direct coding or access through tasks in SAS Studio.
- **Interactive modeling interface:** a collection of tasks in SAS Visual Analytics for creating models in an interactive manner with automated assessment visualizations
- **Automated modeling interface:** a pipeline application called Model Studio that enables you to construct automated flows consisting of various nodes for preprocessing and modeling, with automated model assessment and comparison, and direct model publishing and registration.

Each of these executes the same underlying actions in the CAS execution environment. In addition, there are supplementary interfaces for preparing your data (Data Studio) and managing and deploying your models (SAS Model Manager and SAS Decision Manager) to support all phases of a machine learning application.

In this book, you primarily explore the Model Studio interface and its integration with other SAS Visual Data Mining and Machine Learning interfaces.

You use the SAS Visual Data Mining and Machine Learning web client to visually assemble, configure, build, and compare data mining models and pipelines for a wide range of analytic data mining tasks.

Chapter 1: Introduction to Machine Learning

- Introduction 1
 - Supervised Learning..... 1
 - Unsupervised Learning..... 2
 - Semisupervised Learning and Reinforcement Learning 2
- Supervised Learning Predictions..... 2
 - Decision Prediction 3
 - Ranking Prediction 3
 - Estimation Prediction..... 4
- Model Building and Selection 5
 - Model Complexity..... 5
- Introducing Model Studio..... 5
 - Demo 1.1: Creating a Project and Loading Data 6
 - Model Studio: Analysis Elements 14
 - Demo 1.2: Building a Pipeline from a Basic Template 18
- Quiz 23

Introduction

There are two main types of machine learning methods, *supervised learning* and *unsupervised learning*.

Supervised Learning

Supervised learning (also known as *predictive modeling*) starts with a training data set. The observations in a training data set are known as *training cases* (also known as *examples*, *instances*, or *records*). The variables are called *inputs* (also known as *predictors*, *features*, *explanatory variables*, or *independent variables*) and *targets* (also known as *responses*, *outcomes*, or *dependent variables*). The learning algorithm receives a set of inputs along with the corresponding correct outputs or targets, and the algorithm learns by comparing its actual output with correct outputs to find errors. It then modifies the model accordingly. Through methods like classification, regression, prediction, and gradient boosting, supervised learning uses patterns to predict the values of the label on additional unlabeled data. In other words, the purpose of the training data is to generate a predictive model. The *predictive model* is a concise representation of the association between the inputs and the target variables.

Supervised learning is commonly used in applications where historical data predicts likely future events. For example, it can anticipate when credit card transactions are likely to be fraudulent or which insurance customer is likely to file a claim.

Unsupervised Learning

Unsupervised learning is used against data that has no historical labels. In other words, the system is not told the “right answer” – there is no target data – the algorithm must figure out what is being shown. The goal is to explore the data and find some structure or pattern. Unsupervised learning works well on transactional data. For example, it can identify segments of customers with similar attributes who can then be treated similarly in marketing campaigns. Or it can find the main attributes that separate customer segments from each other. Popular techniques include self-organizing maps, nearest-neighbor mapping, k-means clustering, and singular value decomposition. These algorithms are also used to segment text topics, recommend items, and identify data outliers.

Semisupervised Learning and Reinforcement Learning

Other common methods include semisupervised learning and reinforcement learning.

Semisupervised learning is used for similar applications as supervised learning. But it uses both labeled and unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data (because unlabeled data is less expensive and takes less effort to acquire). This type of learning can be used with methods such as classification, regression, and prediction. Semisupervised learning is useful when the cost associated with labeling is too high to allow for a fully labeled training process. Early examples of this include identifying a person’s face on a web cam.

Reinforcement learning is often used for robotics, gaming, and navigation. With reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest rewards. This type of learning has three primary components: the agent (the learner or decision maker), the environment (everything the agent interacts with), and actions (what the agent can do). The objective is for the agent to choose actions that maximize the expected reward over a given amount of time. The agent will reach the goal much faster by following a good policy. So the goal in reinforcement learning is to learn the best policy.

In this book, we will be focusing on supervised learning or predictive modeling.

Supervised Learning Predictions

The outputs of the predictive model are referred to as *predictions*. Predictions represent your best guess for the target given a set of input measurements. The predictions are based on the associations learned from the training data by the predictive model.

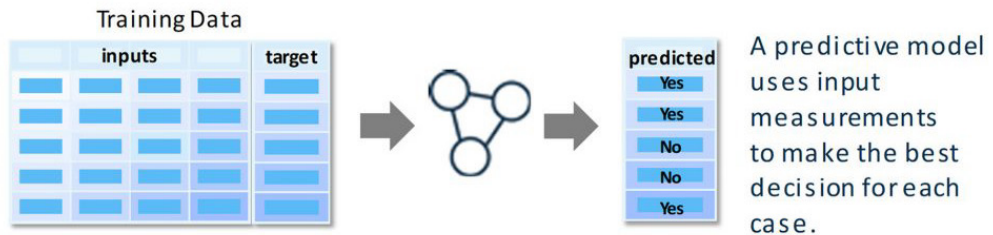
The training data are used to construct a model (rule) that relates the inputs to the target. The predictions can be categorized into three distinct types:

- decisions
- rankings
- estimates

Decision Prediction

Decision predictions are the simplest type of prediction. Decisions usually are associated with some type of action (such as classifying a case as a churn or no-churn). For this reason, decisions are also known as *classifications*. Decision prediction examples include handwriting recognition, fraud detection, and direct mail solicitation.

Figure 1.1: Decision Predictions



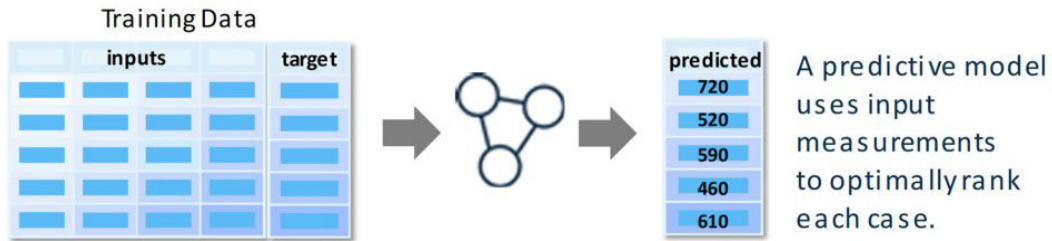
Decision predictions usually relate to a categorical target variable. For this reason, they are identified as primary, secondary, and tertiary in correspondence with the levels of the target.

Note: Model assessment in Model Studio generally assumes decision predictions when the target variable has a categorical measurement level (binary, nominal, or ordinal).

Ranking Prediction

Ranking predictions order cases based on the input variables' relationships with the target variable. Using the training data, the prediction model attempts to rank *high value* cases higher than *low value* cases. It is assumed that a similar pattern exists in the scoring data so that *high value* cases have high scores. The actual produced scores are inconsequential. Only the relative order is important. The most common example of a ranking prediction is a credit score.

Figure 1.2: Ranking Predictions

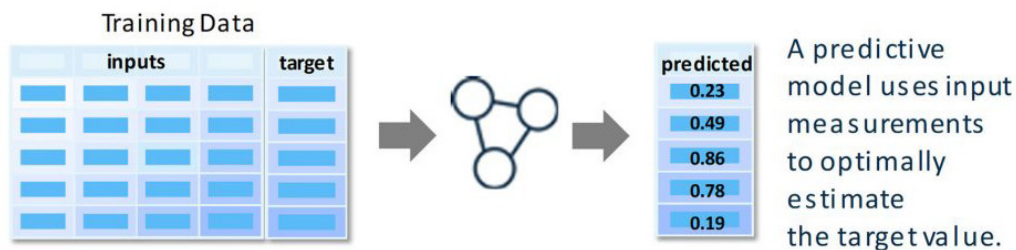


Ranking predictions can be transformed into decision predictions by taking the primary decision for cases above a certain threshold while making secondary and tertiary decisions for cases below the correspondingly lower thresholds. In credit scoring, cases with a credit score above 700 can be called good risks, those with a score between 600 and 700 can be intermediate risks, and those below 600 can be considered poor risks.

Estimation Prediction

Estimation prediction uses the inputs to estimate a *value* for the dependent variable conditioned on some unobserved values of the independent variable. For cases with numeric targets, this can be thought of as the average value of the target for all cases having the observed input measurements. For cases with categorical targets, this number might equal the probability of a target outcome.

Figure 1.3: Estimate Prediction.



Prediction estimates are most commonly used when their values are integrated into a mathematical expression. For example, two-stage modeling, where the probability of an event is combined with an estimate of profit or loss to form an estimate of unconditional expected profit or loss. Prediction estimates are also useful when you are not sure of the ultimate application of the model.

Estimate predictions can be transformed into both decision and ranking predictions. When in doubt, use this option. Most Model Studio modeling tools can be configured to produce estimate predictions.

Model Building and Selection

In order to choose the best model for the business problem and data, many models are built and compared in order to choose a *champion model*, which can then be deployed into production. We will discuss scoring and model selection in a later chapter. But before you start building models it is important to hold back some of the data to be used to help select the best model.

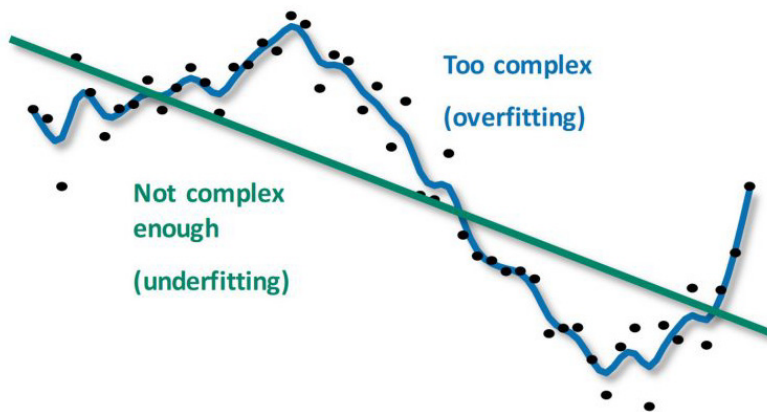
Model Complexity

Selecting model complexity is a balance between bias and variance. An insufficiently complex model might not be flexible enough, which leads to *underfitting*. An underfit model leads to biased inferences, which means that they are not the true ones in the population; for example, in the case of a decisioning model, they could predict “no” when the target should be “yes.”

An overly complex model might be too flexible, which leads to *overfitting*. An overfit model includes the random noise in the sample, which can lead to models that have higher variance when applied to the population. This model would perform almost perfectly with the training data but is likely to have poor performance with the validation data.

A model with just enough flexibility gives the best generalization.

Figure 1.4: Accuracy Versus Generalizability



Introducing Model Studio

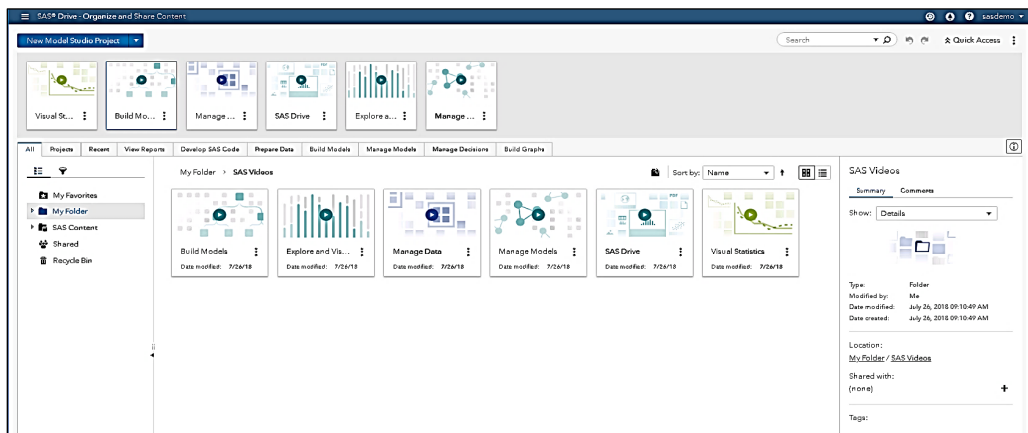
Model Studio enables you to explore ideas and discover insights by preparing data and building models. It is part of the **discovery** piece of the analytics life cycle. Model Studio is a central, web-based application that includes a suite of integrated data mining tools. The data mining tools supported in Model Studio are designed to take advantage of the SAS Viya programming and cloud processing environments to deliver and distribute analytic model data mining champion models, score code, and results.



Demo 1.1: Creating a Project and Loading Data

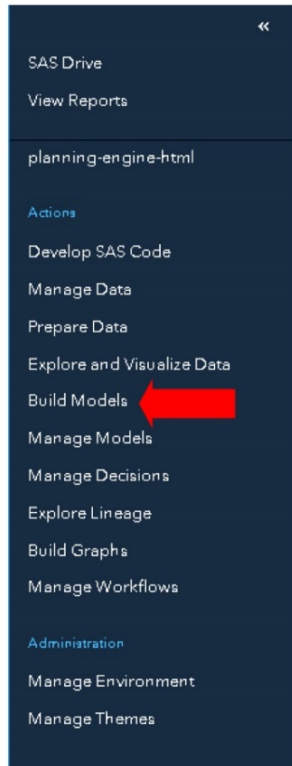
In this demonstration, you will create a new project in Model Studio based on the **commsdata** data set. A project is a top-level container for your analytic work in Model Studio. The table is imported from a local drive. The type of project is defined. This project is used to predict churn for a fictitious telecommunications company. A target variable is selected for this table.

1. First, open SAS Drive on your machine and select **SAS Viya ► SAS Drive** from the bookmarks bar or from the link on the page.
2. Next, log on using your user ID and password.
Note: Use caution when you enter the user ID and password because values can be case-sensitive.
3. Click **Sign In**.
4. Select **Yes** in the Assumable Groups window. The SAS Drive home page appears.



Note: The SAS Drive page on your computer might not have the same tiles as the image above.

5. Click the Applications menu in the upper left corner of the SAS Drive page. Select **Build Models**.



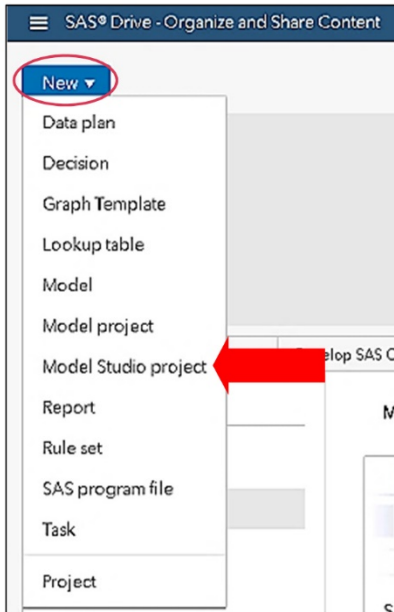
This launches Model Studio.

Note: Some of the top features in Model Studio in SAS Visual Data Mining and Machine Learning are presented in a paper titled “Playing Favorites: Our Top 10 Model Studio Features in SAS® Visual Data Mining and Machine Learning” at

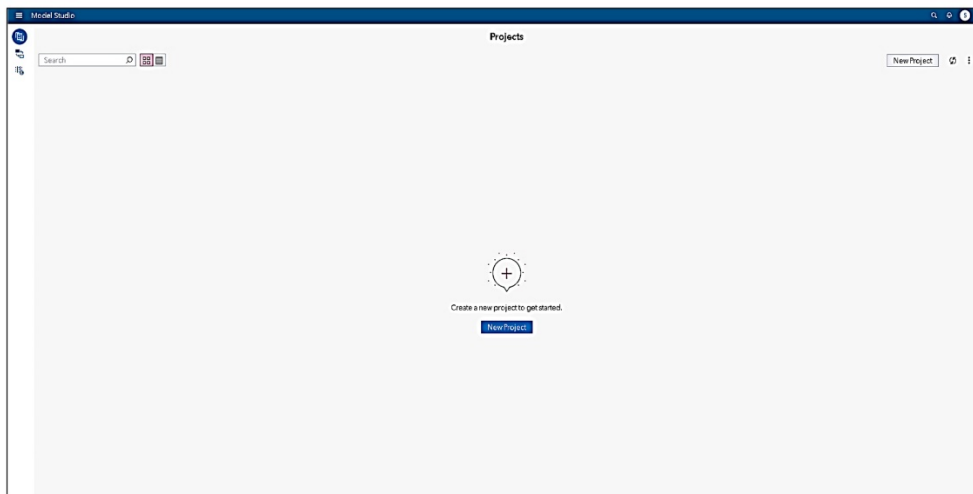
<https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3236-2019.pdf>.

Alternatively, click **New** in the upper left corner to reveal a menu to create a new item. Select **Model Studio project** from the menu.

Note: When this alternative process is used to go to Model Studio, it bypasses the Model Studio Projects page and immediately opens the window to create a new project as shown below in step 7 of this demonstration.



The Model Studio Projects page is now displayed.



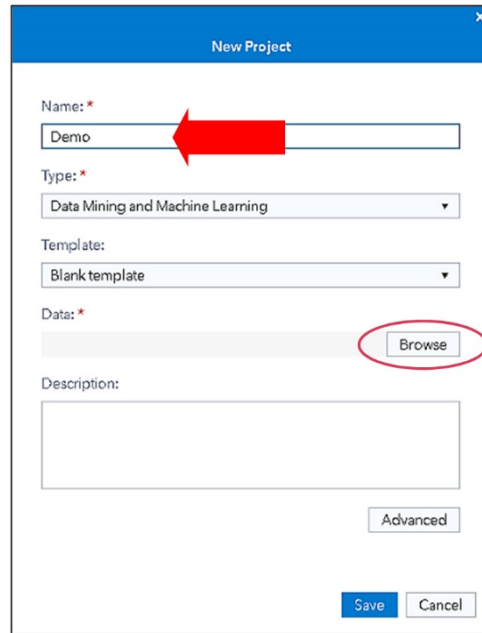
Note: On your computer, the Projects page might differ from the image above. There might be pre-existing projects on your computer.

From the Model Studio Projects page, you can view existing projects, create new projects, access the Exchange, and access Global Metadata. Model Studio projects can be one of three types (depending on the SAS licensing for your site): Forecasting projects, Data Mining and Machine Learning projects, and Text Analytics projects.

Note: The Exchange organizes your favorite settings and enables you to collaborate with others in one place. Find a recommended node template or create your own

template for a streamlined workflow for your team. The Exchange is accessed later in this chapter.

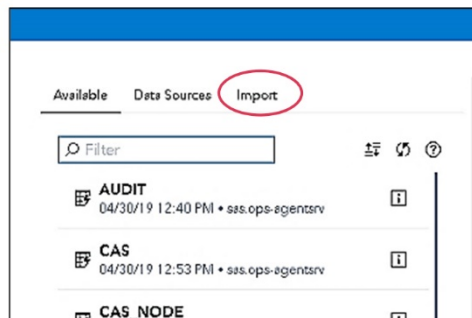
6. Select **New Project** in the upper right corner of the Projects page.
7. Enter **Demo** as the name in the New Project window. Leave the default type of **Data Mining and Machine Learning**. Click **Browse** in the Data field.



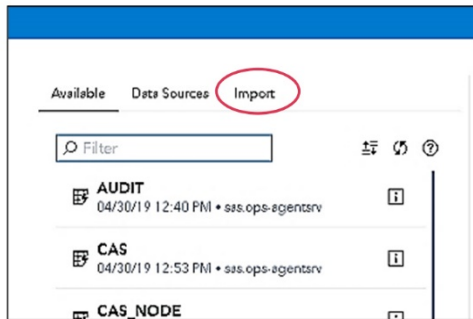
The screenshot shows the 'New Project' dialog box. The 'Name' field is filled with 'Demo'. The 'Type' dropdown menu is set to 'Data Mining and Machine Learning'. The 'Template' dropdown menu is set to 'Blank template'. The 'Data' field is empty, and the 'Browse' button next to it is circled in red. There is also a 'Description' text area and 'Save' and 'Cancel' buttons at the bottom.

Note: You can specify a pipeline template at project creation. Continue with a blank template. Pipeline templates are discussed soon.

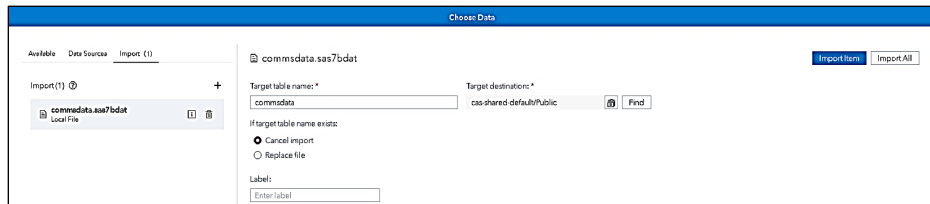
8. Import a SAS data set into CAS.
 - a. In the Choose Data window, click **Import**.



- b. Under Import, select **Local File**.

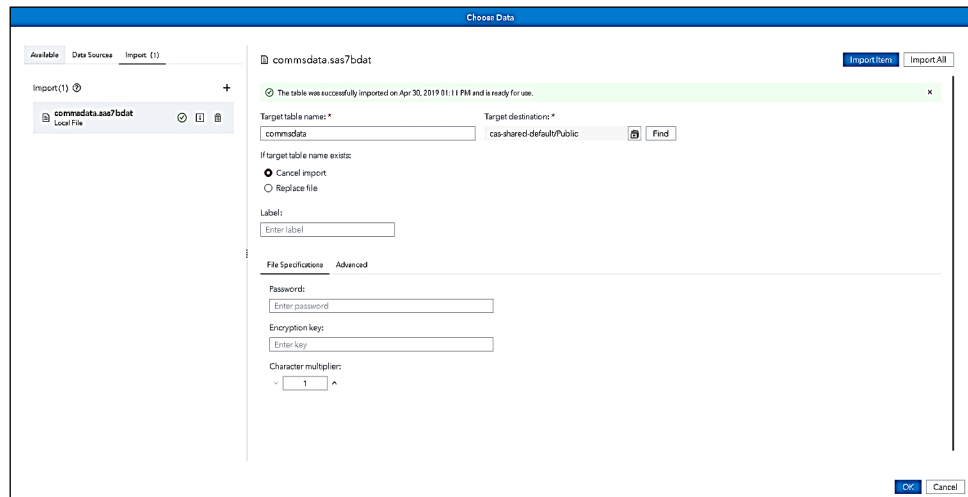


- c. Navigate to the data folder.
d. Select the **commsdata.sas7bdat** table. Click **Open**.
e. Select **Import Item**. Model Studio parses the data set and pre-populates the window with data set configurations.



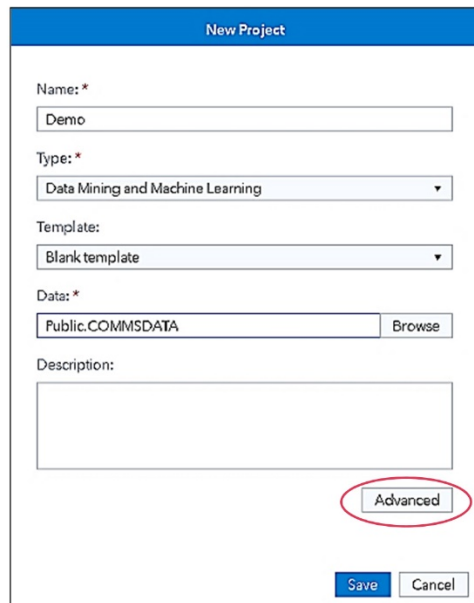
Note: When the data is in memory, it is available for other projects through the Available tab.

- f. Click **OK** after the table is imported.



Note: Tables are imported to the CAS server and are available to use with SAS Visual Analytics. When the import is complete, you are returned to Model Studio. For more information about data types supported in CAS and how to load data into CAS, see the details section at the end of this demo.

9. Click **Advanced** in the New Project window.



10. The Advanced project settings appear. There are four groups of Advanced project settings: Advisor Options, Partition Data, Event-based Sampling, and Node Configuration.

Under the Advisor Options group, there are three options:

Maximum class levels specifies the threshold for rejecting categorical variables. If a categorical input has more levels than the specified maximum number, it is rejected.

Interval cutoff determines whether a numeric input is designated as interval or nominal. If a numeric input has more distinct values/levels than the interval cutoff value, it is declared interval. Otherwise, it is declared nominal.

Maximum percent missing specifies the threshold for rejecting inputs with missing values. If an input has a higher percentage of missing values than the specified maximum percent, it is rejected. This option can be turned on or off. It is on by default.

Note: This is the only place where these Advisor Options are seen and can be changed.

The screenshot shows the 'New Project Settings' dialog box with the 'Advisor Options' tab selected. On the left, there is a list of settings: 'Advisor Options' (selected), 'Partition Data', 'Event-Based Sampling', and 'Node Configuration'. The main area displays the 'Advisor Options' configuration. It includes three input fields: 'Maximum class level' with the value 20, 'Interval cutoff' with the value 20, and 'Maximum percent missing' with the value 60. A checkbox labeled 'Apply the "maximum percent missing" limit' is checked.

The Advanced project settings options for Partition Data and Event-Based Sampling are covered in the next chapter. And, along with Node Configuration, they are discussed in the next demo. You can access the Partition Data, Event-Based Sampling, and Node Configuration options here, and you can also access them after the project is created.

Click **Cancel** to return to the New Project window.

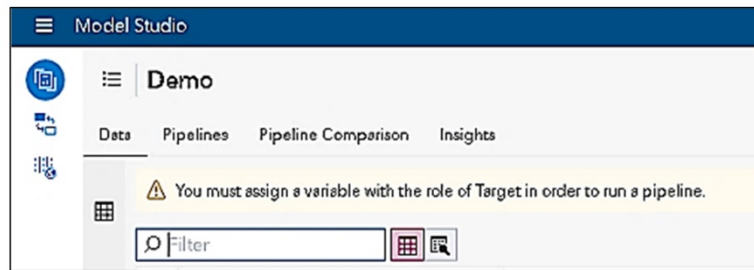
11. Click **Save**.

The screenshot shows the 'New Project' dialog box. It contains several fields: 'Name' with the value 'Demo', 'Type' set to 'Data Mining and Machine Learning', 'Template' set to 'Blank template', and 'Data' set to 'Public.COMMSDATA' with a 'Browse' button next to it. There is an empty 'Description' text area. At the bottom right, there are two buttons: 'Advanced' and 'Save'. The 'Save' button is highlighted with a red circle.

Note: After you create your new project, Model Studio takes you to the Data tab of your new project. Here, you can adjust data source variable role and level assignments and define certain metadata rules (for example, methods of imputation and transformation). You can also retrain a model with new data, if the target variable in the new data set is the same as the original data set.

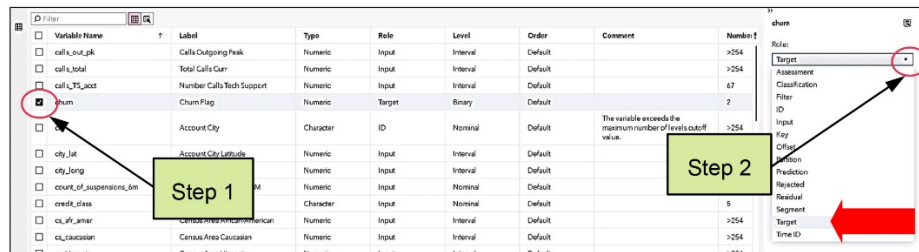
Note: In Model Studio, *metadata* is defined as the set of variable roles, measurement levels, and other configurations that apply to your data set. When you need to create multiple projects using similar data sets (or when using a single data set), you might find it useful to store the metadata configurations for usage across projects. Model Studio enables you to do this by collecting the variables in a repository named Global Metadata. By storing your metadata configurations as global metadata, the configurations will apply to new data sets that contain variables with the same names.

12. When the project is created, you need to assign a target variable to run a pipeline. In Model Studio, you can create analytic process flow in the form of a pipeline.



You can also have target variable roles already defined in your data. Model Studio provides several options for managing and modifying data. The Data tab enables you to modify variable assignments and manage global metadata.

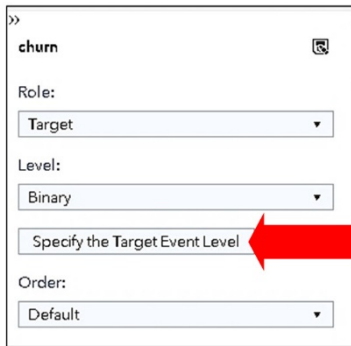
13. In the variables window, select **churn** (Step 1). Then in the right pane, select **Target** under the Role property (Step 2). (You might need to scroll down in the variable list to see **churn**.)



The right pane enables you to specify several properties of the variables, including Role, Level, Order, Transform, Impute, Lower Limit, and Upper Limit.

For the Transform, Impute, Lower Limit, and Upper Limit properties, altering these values on the Data tab does not directly modify the variable. Instead, this sets metadata values for these properties. The Data Mining Preprocessing nodes that use metadata values (Transformations, Impute, Filter, and Replacement) might use these parameters if the corresponding action is requested. You see this in the next few demonstrations.

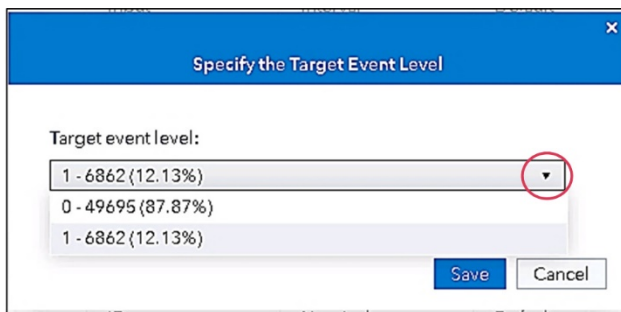
14. Click **Specify the Target Event Level**. You can specify the target event level here that needs to be modeled.



The screenshot shows a configuration window for the variable 'churn'. It has a 'Role' dropdown set to 'Target', a 'Level' dropdown set to 'Binary', and a button labeled 'Specify the Target Event Level' which is highlighted by a red arrow. Below this is an 'Order' dropdown set to 'Default'.

15. Click the drop-down arrow.

Note that the churn rate is around 12%. By default, Model Studio considers alphanumerically the last category as the event, and therefore no change is required.



The screenshot shows a dialog box titled 'Specify the Target Event Level'. It contains a 'Target event level:' label and a dropdown menu. The dropdown menu is open, showing three options: '1 - 6862 (12.13%)', '0 - 49695 (87.87%)', and '1 - 6862 (12.13%)'. The first option is selected, and the dropdown arrow is circled in red. At the bottom right, there are 'Save' and 'Cancel' buttons.

16. Close the Specify the Target Event Level window.

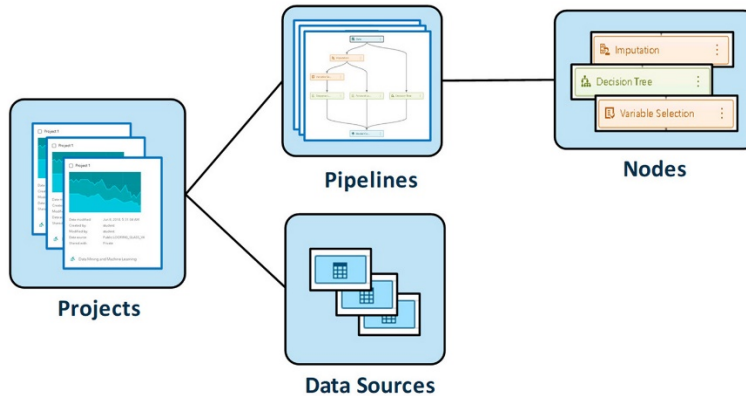
End of Demonstration

Model Studio: Analysis Elements

A **project** is a top-level container for your analytic work in Model Studio. A Model Studio project contains the **data source**, the **pipelines** that you create, and related **project metadata** (such as project type, project creator, share list, and last update history). If you create more than one pipeline in your project, analytic results that compare the performance of multiple pipelines are also stored in the project.

Model Studio: Analysis Elements

Figure 1.5: Analysis Events in Model Studio



You can add nodes to the pipeline to create your modeling process flow. You can save results from SAS Visual Data Mining and Machine Learning nodes by inserting a SAS Code node after the node (and sometimes a Manage Variable node before the SAS Code Node). This enables you to write some specific DATA step or other Base SAS or CASL statements to save your desired outputs and data sets to a permanent library for further use.

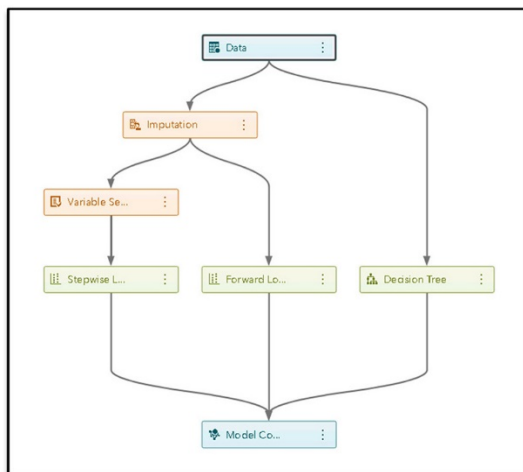
If you want to capture the data being exported out of a node, you can simply attach a Save Data node anywhere on your pipeline and specify details of the table that you want to save. In addition, there is an output tab in the results of nodes that enables you to view the scored output table. From here, you can specify a sample if desired, as well as request to save the table. You can also save data other than the scored output tables. Further, you can download and save data from the tables and plots shown in the results of nodes. This is done by clicking the **Download Data** shortcut button in the upper right corner, which appears right next to the **Expand** shortcut button.

/opt/sas/viya/config/data/cas/default/projects				
Name	Size	Changed	Rights	Owner
..		12/20/2017 1:15:20 PM	rwxt-xr-x	cas
datamining-8eb9539b-95c3-46bc-807f-588213f7566c		3/9/2018 10:07:29 AM	rwxt-xr-x	cas
datamining-63e94ff6-07c7-4b64-9d79-87f77e66bc9		6/1/2018 3:01:21 AM	rwxt-xr-x	cas
datamining-a7976f72-a721-407c-af1c-a304067b1e69		5/29/2018 3:13:48 AM	rwxt-xr-x	cas
datamining-de12ceab-326f-49b9-aaf0-c82561a068f4		5/29/2018 6:18:34 AM	rwxt-xr-x	cas
datamining-f5428069-a643-4b2a-a567-d0da70c16ebe		6/8/2018 5:30:53 AM	rwxt-xr-x	cas
datamining-fad7c7d7-b67e-4d66-91b7-14b01e6939e6		5/28/2018 7:20:40 AM	rwxt-xr-x	cas
forecasting-50657c82-e4f2-41eb-8dcc-6501448dc47d		3/9/2018 10:40:13 AM	rwxt-xr-x	cas

Model Studio: Analysis Elements

You can view the list of projects (like above) by navigating to the location where it has been saved. Shown above is the path for a Linux OS using WinSCP (a File Transfer Protocol application on your client machine). Generally, the path is **/opt/sas/viya/config/data/cas/default/projects/**. You might have a different path if it is a Windows installation.

Figure 1.6: Pipeline



- Pipelines are structured flows of analytic actions.
- Pipelines contain the nodes that process data and create models.
- Custom pipelines can be saved to **the Exchange** for others to use.

A *pipeline* is an analytic process flow. After creating a new pipeline, you can create visual data mining functionality by adding nodes to the pipeline. Nodes can be added separately, or, to save time, templates can add several nodes at once. To create a pipeline from a template, specify the template in the New Pipeline window. You can add nodes to a pipeline in the following two ways:

1. Drag and drop from an expanded Nodes pane.
2. Right-click and select either **Add child node** or **Add parent node**.

Pipelines are grouped together in a top-level container (that is, in a project that also includes the data set that you want to model and a pipeline comparison tool). A project can contain multiple pipelines. You can create a new pipeline and modify an existing pipeline.

Pipelines can be saved to the *Exchange* where they become accessible to other users. All available nodes, along with descriptions, and all available pipeline templates, including pre-built and user-created, can be found here.

Model Studio: Analysis Elements

Templates

Model Studio supports templates as a method for creating statistical models quickly. A *template* is a special type of pipeline that is pre-populated with configurations that can be used to create a model. A template might consist of multiple nodes or a single node. Model Studio includes a set of templates that represent frequent use cases, but you can also create models themselves and save them as templates in the Exchange.

There are three levels of templates available, both for a class target as well as for an interval target. An intermediate template for class target was shown in Figure 1.6. You can create a new template from an existing pipeline, create a new template in the Exchange, and modify an existing template.

The advanced templates are also available with *autotuning* functionality. A large portion of the model-building process is taken up by experiments to identify the optimal set of parameters for the model algorithm. As algorithms get more complex (neural networks to deep neural networks, decision trees to forests and gradient boosting), the amount of time required to identify these parameters grows. There are several ways to support you in this cumbersome work of tuning machine learning model parameters. These approaches are called *hyperparameter optimization* and are discussed later in the book. The following pipeline templates are included with Model Studio:

Table 1.1: Pipeline Templates

Pipeline Template Name	Pipeline Template Description
Blank template	A data mining pipeline that contains only a Data node
Basic template for class target	A simple linear flow: Data, Imputation, Logistic Regression, Model Comparison
Basic template for interval target	A simple linear flow: Data, Imputation, Linear Regression, Model Comparison
Intermediate template for class target	Extends the basic template with a stepwise logistic regression model and a decision tree

Model Studio: Analysis Elements

Pipeline Template Name	Pipeline Template Description
Intermediate template for interval target	Extends the basic template with a stepwise linear regression model and a decision tree
Advanced template for class target	Extends the intermediate template for class target with neural network, forest, and gradient boosting models, as well as an ensemble
Advanced template for class target with autotuning	Advanced template for class target with autotuned tree, forest, neural network, and gradient boosting models

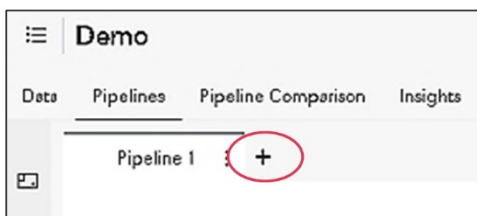
The next demo shows how to build a new pipeline from a basic template for a class target. This template is a simple linear flow and includes a logistic regression node as the predictive model. Chapter 5 includes a refresher on logistic regression for those who are not familiar with this technique.



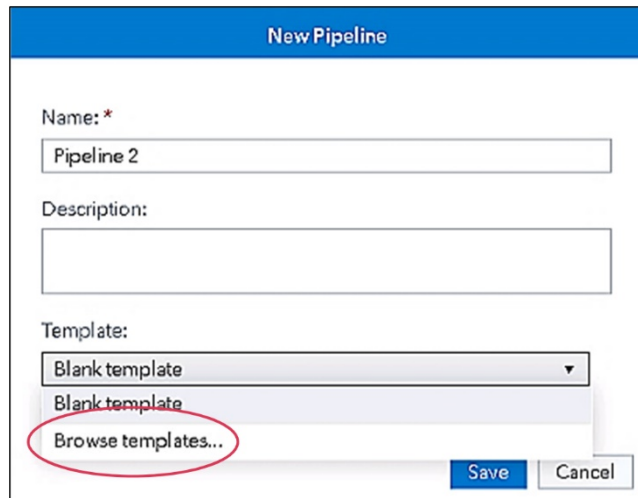
Demo 1.2: Building a Pipeline from a Basic Template

Although it is nice to be able to build up your own pipelines from scratch, it is often convenient to start from a template that represents best practices in building predictive models. The application comes with a nice set of templates available for creating new pipelines. In this demonstration, to start simple, you build a new pipeline from a basic template for class target.

1. Click + next to the current pipeline tab in the upper left corner of the canvas.



2. In the New Pipeline window, select **Browse templates** in the **Template** field.



New Pipeline

Name: *

Pipeline 2

Description:

Template:

Blank template

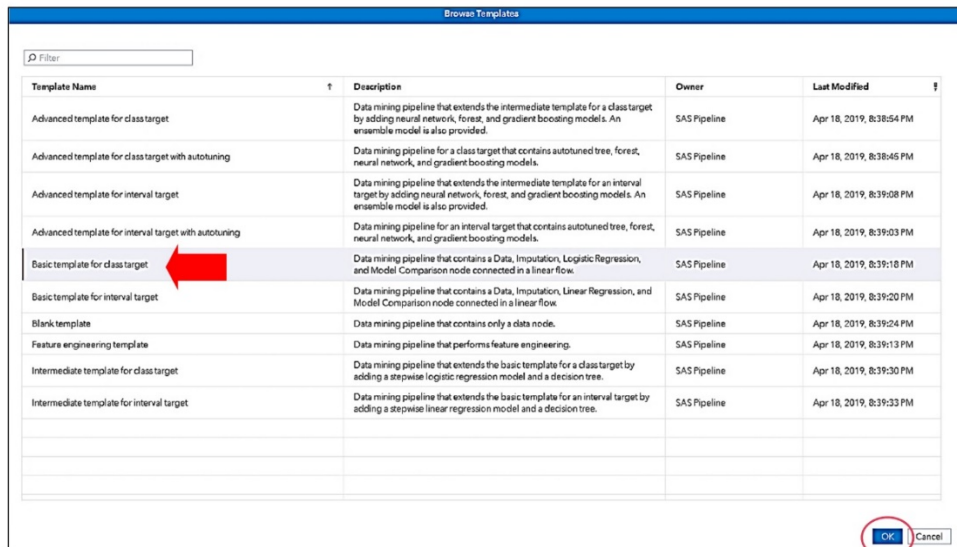
Blank template

Browse templates...

Save Cancel

Note: Some of the options on the Template menu might be different on your computer from what is shown above.

3. In the Browse Templates window, select **Basic template for class target**. Click **OK**.



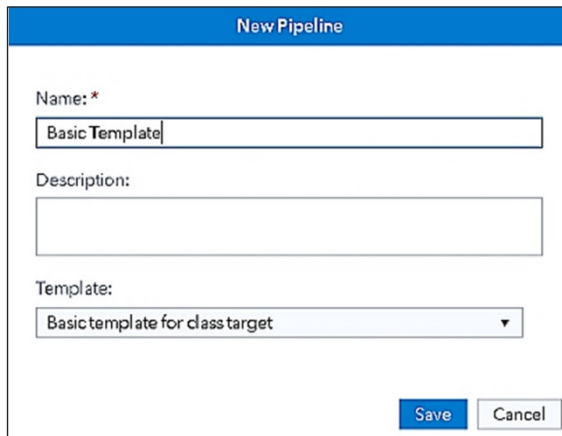
Browse Templates

Filter

Template Name	Description	Owner	Last Modified
Advanced template for class target	Data mining pipeline that extends the intermediate template for a class target by adding neural network, forest, and gradient boosting models. An ensemble model is also provided.	SAS Pipeline	Apr 18, 2019, 8:38:54 PM
Advanced template for class target with autotuning	Data mining pipeline for a class target that contains autotuned tree, forest, neural network, and gradient boosting models.	SAS Pipeline	Apr 18, 2019, 8:38:45 PM
Advanced template for interval target	Data mining pipeline that extends the intermediate template for an interval target by adding neural network, forest, and gradient boosting models. An ensemble model is also provided.	SAS Pipeline	Apr 18, 2019, 8:39:08 PM
Advanced template for interval target with autotuning	Data mining pipeline for an interval target that contains autotuned tree, forest, neural network, and gradient boosting models.	SAS Pipeline	Apr 18, 2019, 8:39:03 PM
Basic template for class target	Data mining pipeline that contains a Data, Imputation, Logistic Regression, and Model Comparison node connected in a linear flow.	SAS Pipeline	Apr 18, 2019, 8:39:10 PM
Basic template for interval target	Data mining pipeline that contains a Data, Imputation, Linear Regression, and Model Comparison node connected in a linear flow.	SAS Pipeline	Apr 18, 2019, 8:39:20 PM
Blank template	Data mining pipeline that contains only a data node.	SAS Pipeline	Apr 18, 2019, 8:39:24 PM
Feature engineering template	Data mining pipeline that performs feature engineering.	SAS Pipeline	Apr 18, 2019, 8:39:13 PM
Intermediate template for class target	Data mining pipeline that extends the basic template for a class target by adding a stepwise logistic regression model and a decision tree.	SAS Pipeline	Apr 18, 2019, 8:39:30 PM
Intermediate template for interval target	Data mining pipeline that extends the basic template for an interval target by adding a stepwise linear regression model and a decision tree.	SAS Pipeline	Apr 18, 2019, 8:39:33 PM

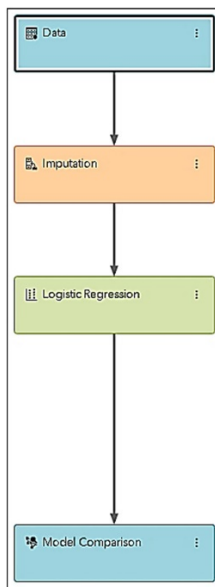
OK Cancel

4. In the New Pipeline window, name the pipeline **Basic Template**.



The 'New Pipeline' dialog box has a blue title bar. It contains three input fields: 'Name: *' with the text 'Basic Template', 'Description:' which is empty, and 'Template:' with a dropdown menu showing 'Basic template for class target'. At the bottom right are 'Save' and 'Cancel' buttons.

5. Click **Save**.



The *basic template for class target* is a simple linear flow and includes the following nodes: Data, Imputation, Logistic Regression, and Model Comparison. You can add nodes by right-clicking the existing nodes (or dragging and dropping from the Nodes pane.)

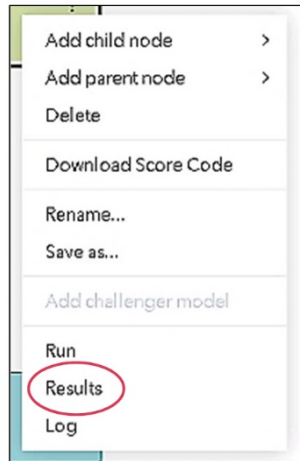
Different colors of nodes represent their respective groups in the Model Studio.

Note: Because a predicted response might be different for cases with a missing input value, a binary imputation indicator variable is often added to the training data. Adding this variable enables a model to adjust its predictions in the situation where “missingness” itself is correlated with the target.

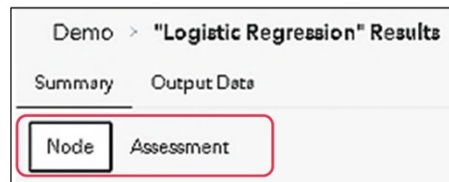
6. Click **Run Pipeline** in the upper right corner.



7. After the pipeline has successfully run, right-click the **Logistic Regression** node and select **Results**.



The Results window contains two important tabs at the top: one for Node results and one for Assessment results.

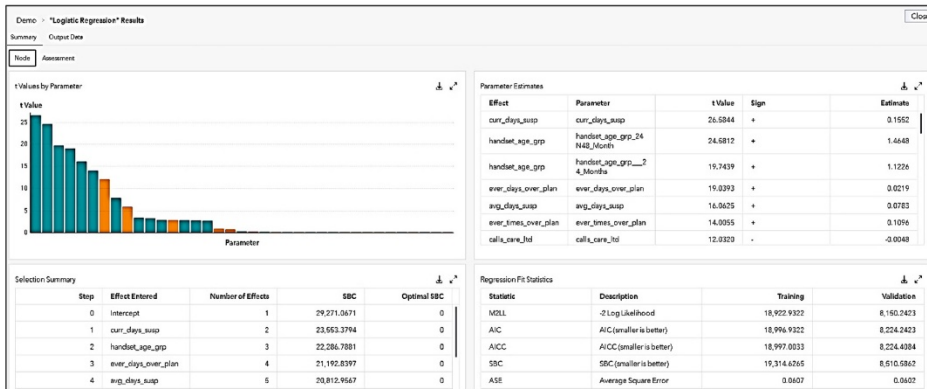


Here are some of the windows included under the Node tab in the results from the Logistic Regression node:

- t-values by Parameter plot
- Parameter Estimates table
- Selection Summary table
- Output

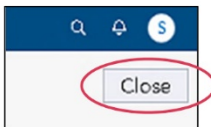
Here are some of the windows included under the Assessment tab in the results from the Logistic Regression node:

- Lift Reports plots
- ROC Reports plots
- Fit Statistics table

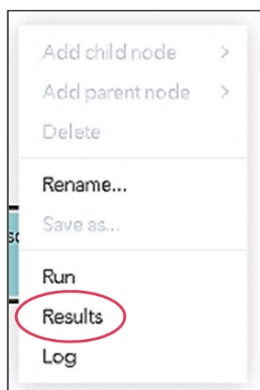


Explore the results as you see fit.

8. Close the Results window by clicking **Close** in the upper right corner of the window.



9. Right-click the **Model Comparison** node and select **Results**.




10. Click to expand the **Model Comparison** table. Unless specified, the default fit statistic (KS) is used for selecting a champion model with a class target.

Note: To change the default fit statistic for just this comparison, change the class selection statistic of the Model Comparison properties in the right-hand pane when the node is selected in the pipeline. To change the default fit statistic for all projects,

change the class selection statistic on the Project Settings menu. The default is the Kolmogorov-Smirnov statistic (KS).

A subset of the Model Comparison table is shown below.

Model Comparison									
Champl...	Name	Algorit...	KS (You...	Misclas...	Misclas...	Root AV...	Averag...	Sum of...	Multi-Cl...
	Logistic Regression	Logistic Regression	0.5672	0.0660	0.0660	0.2454	0.0602	16,967	0.2402

Note: The Model Comparison node is always added by default when any model is contained in the pipeline. If the pipeline contains only a single model, the Model Comparison node summarizes performance of this one model.

- Exit the maximized view by clicking **X** in the upper right corner of the window.

End of Demonstration

Quiz

- After you create your new project, Model Studio takes you to the Data tab. What can you do in the Data tab? (Select all that apply.)
 - Modify variable roles and measurement levels.
 - Manage global metadata.
 - Modify variable names and labels.
 - Manage columns to display the Variables table.

Ready to take your SAS® and JMP® skills up a notch?



Be among the first to know about new books,
special events, and exclusive discounts.

support.sas.com/newbooks

Share your expertise. Write a book with SAS.

support.sas.com/publish



sas.com/books
for additional books and resources.

sas
THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.
Other brand and product names are trademarks of their respective companies. © 2017 SAS Institute Inc. All rights reserved. M1588358 US.0217