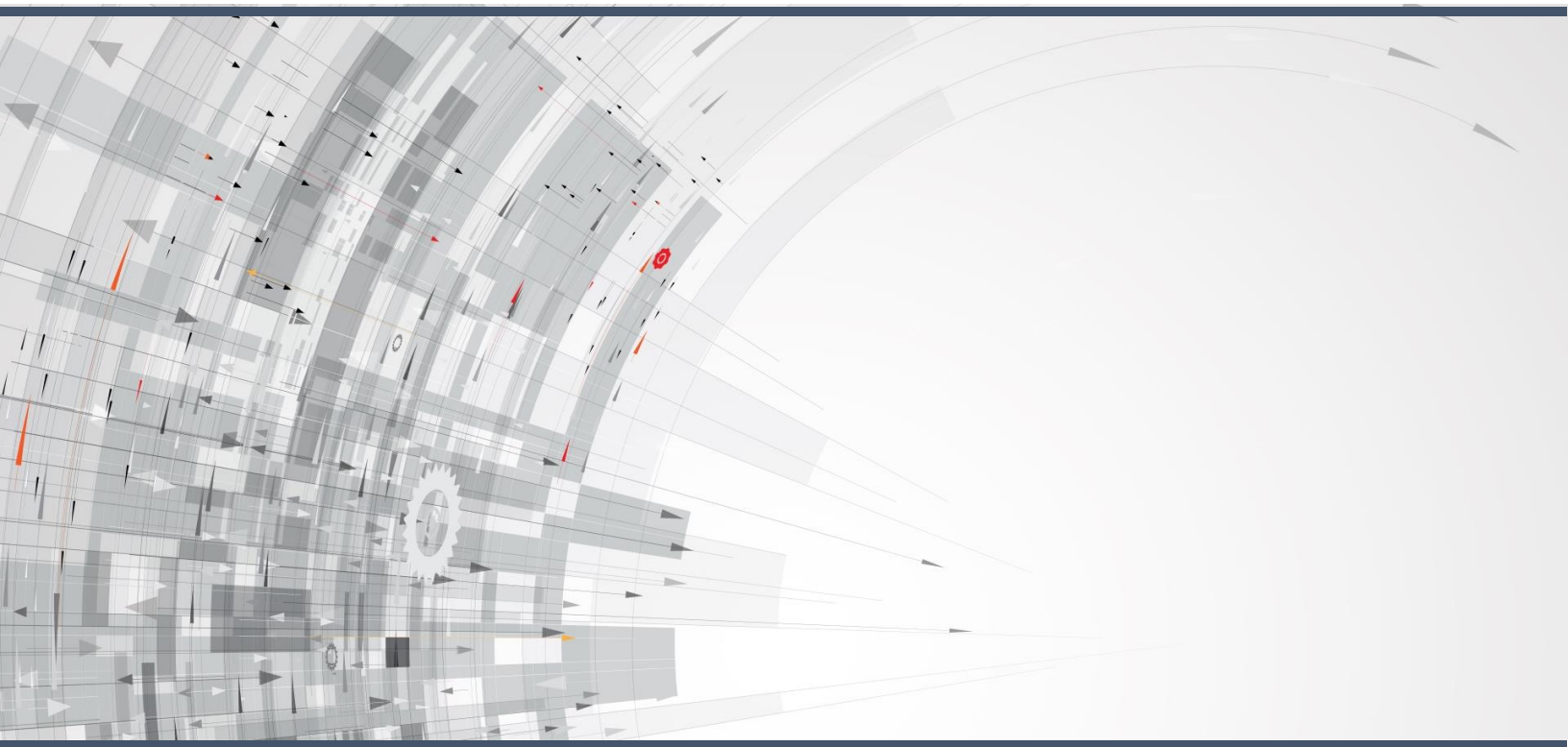
Wintellect®
Know how.



Understanding
Azure Machine Learning

Build sophisticated machine-learning models quickly and harness the power of predictive analytics to aid your research and build smarter apps

October 2016

Machine learning, which facilitates predictive analytics using large volumes of data by employing algorithms that iteratively learn from that data, is one of the fastest growing areas of computer science. Its uses range from credit-risk prediction and spam filtering to optical character recognition (OCR) and online shopping recommendations. It makes us smarter by making computers smarter. And its usefulness will only increase as more and more data becomes available and the desire to perform predictive analytics from that data – or simply find patterns in it – grows, too.

Though few outside of the data-science community are aware that it exists, machine learning touches lives every day. When your credit-card company calls to validate a transaction that just occurred, it was probably machine learning that tipped them off to possible fraud. The model was trained with information regarding millions of credit-card transactions, each classified as fraudulent or not-fraudulent. With a model thusly trained, each new transaction is run through the model and its validity is determined with an astonishing degree of accuracy.

Azure Machine Learning is a cloud-based predictive-analytics service that offers a streamlined experience for data scientists of all skill levels. It's accompanied by the Azure Machine Learning Studio, which is a browser-based tool for building machine-learning models using a drag-and-drop paradigm. It comes with a library of time-saving experiments and features best-in-class algorithms developed and tested in the real world by Microsoft businesses such as Bing and Hotmail. And its built-in support for the R programming language and Python means you can build custom scripts to customize your model. Once you've built and trained a model in Azure Machine Learning Studio, you can easily expose it as a Web service that is consumable from a variety of programming languages, or share it with the community by placing it in the Cortana Intelligence Gallery.

The Science of Machine Learning

Machine-learning models fall into two broad categories: supervised and unsupervised.

In supervised learning, the model is trained using existing data for the purpose of predicting outcomes from future data. For example, suppose you have a large volume of data regarding cancer patients and their age, socioeconomic backgrounds, and lifestyle habits, and you want to build a model that predicts the probability that a new patient will be diagnosed with cancer. You could achieve this by building a supervised-learning regression model. *Regression models* use regression algorithms to predict numeric outcomes – for example, the probability of contracting a disease – from a range of features such as age, gender, income level, height, and weight.

Regression models rely on algorithms that are common in statistical analysis. The simplest regression algorithm is *univariate linear regression*, also known as *simple linear regression*, which models the relationship between a single independent variable *X* and a dependent variable *Y* by best-fitting the equation of a line (hence "linear regression") to a set of inputs, as shown in Figure 1. Other commonly used regression algorithms include *multivariate linear regression*, which accepts multiple input values *X*, *neural network regression*, which "learns" by mimicking the behavior of the human brain, and *decision trees*, which use statistical methods to split the data into a tree of hierarchically arranged nodes from which an outcome can be predicted by following a path from top to bottom.

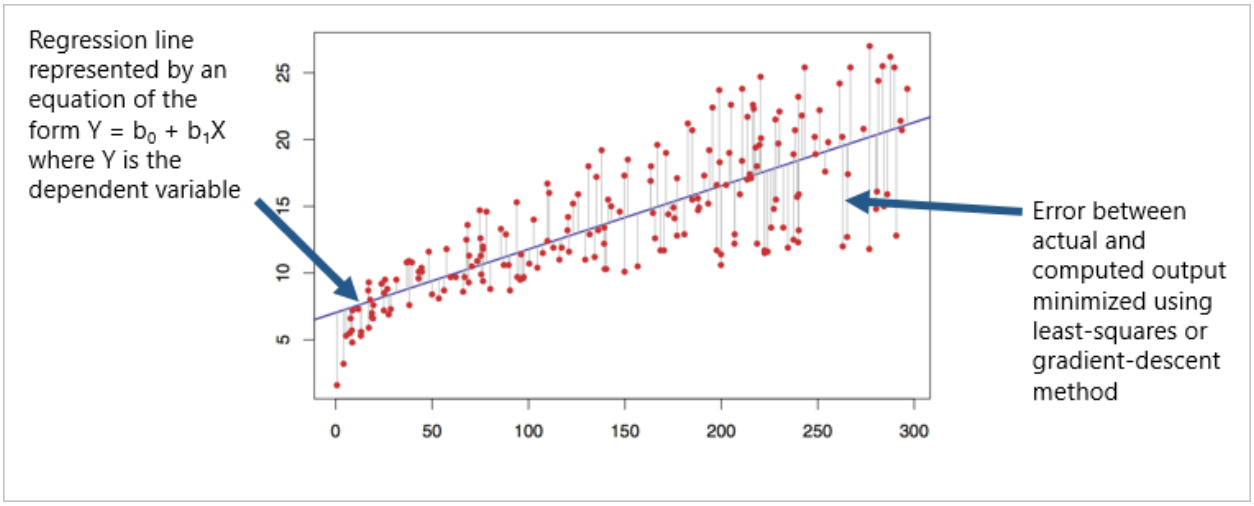


Figure 1: Univariate linear regression

Regression models are very common in supervised learning, but so are *classification models*. The purpose of a classification model is to predict an outcome from a finite range of possible outcomes – for example, classifying a credit-card transaction as fraudulent or not-fraudulent or an e-mail as spam or not-spam. Classification models employ various classification algorithms such as Support Vector Machine (SVM), Bayes Point Machine, and logistic regression (also known as "logit regression"), which uses regression to estimate the probability of each outcome in a finite set of outcomes.

Regression and classification are instances of supervised learning and are used to perform predictive analytics – given a set of inputs, predict the outcome based on patterns identified in the training data. Unsupervised learning, by contrast, seeks to find patterns in the data and use those patterns to identify groups or *clusters* of similar objects. A common application for unsupervised learning is identifying groups of customers with similar buying habits or socioeconomic backgrounds in order to develop targeted marketing strategies.

Unsupervised learning algorithms abound, but one of the most common is *k-means clustering*, which is illustrated in Figure 2. The goal of k-means is to divide a set of *n* data points into a set of *k* groups. It is typically implemented using an iterative algorithm that begins by grouping data points by their proximity to *k* centroids and computing a new centroid for each group, and then repeating until the centroids no longer move. Like all machine-learning algorithms, k-means has strengths and weaknesses. Among its weaknesses is the fact that the outcome tends to be very sensitive to the initial centroid selections.

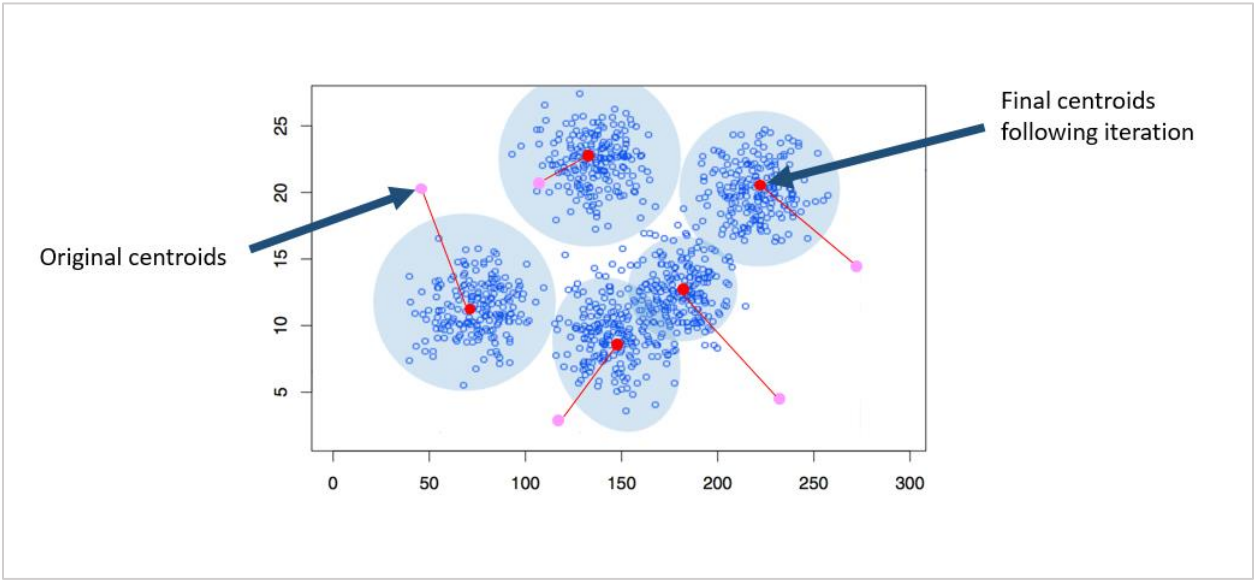


Figure 2: k-means clustering

Even for a trained data scientist, picking the right algorithm for a robust and accurate machine-learning model can be challenging. For those not versed in data science, it can be daunting. To help, Microsoft offers the Machine Learning Algorithm Cheat Sheet (Figure 3), which helps identify candidate algorithms based on the model's intended goal. The latest version is available at http://aka.ms/MLCheatSheet. In addition, the article How to choose algorithms for Microsoft Azure Machine Learning offers valuable insights into choosing an algorithm.

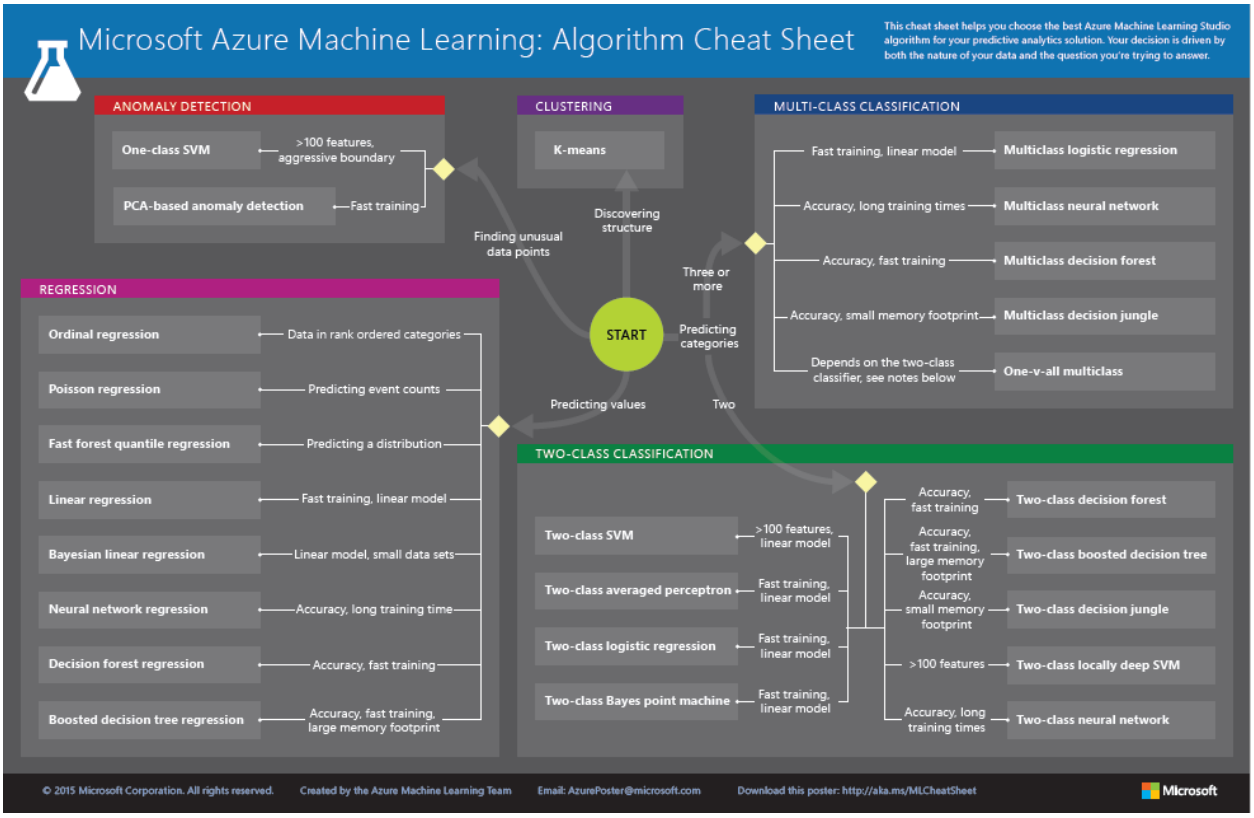


Figure 3: Microsoft's Machine Learning Algorithm Cheat Sheet

A second challenge in implementing a machine-learning model is coding the algorithm. Azure Machine Learning helps out in this regard by providing canned implementations of 25 of the most commonly used algorithms in machine learning. The goal is to make machine learning more approachable by not requiring a data scientist to be a programmer as well as a machine-learning expert. You can always code algorithms of your own in R or Python. But the whole point of Azure Machine Learning is that you rarely have to, hence the reason it is sometimes described as machine learning for the masses.

Azure Machine Learning

Azure Machine Learning combines Software-as-a-Service (SaaS) with a sleek visual editor (Azure Machine Learning Studio) to simplify the task of building and deploying machine-learning models. The general process for building such a model is illustrated in Figure 4, which is taken from the whitepaper Introducing Azure Machine Learning by David Chappell.

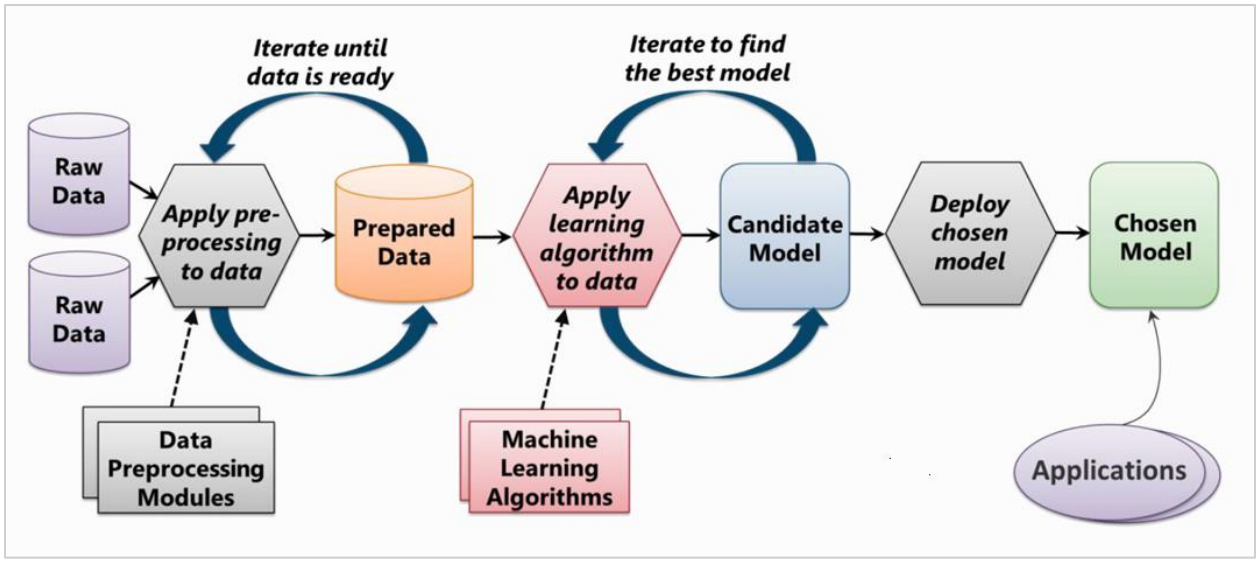


Figure 4: The machine learning process

Workflow begins with the data itself, which can be uploaded to ML Studio or imported from a variety of data sources, including REST endpoints, Hive queries, and Azure Storage. ML Studio supports a variety of data formats, including CSV, TSV, OData, RData, and even zip files.

> "I spent most of last semester building an ML model in Python, and I just did the same thing with Azure ML in 10 minutes." – *Grad student at the University of Massachusetts*

Because real-world datasets typically require preparation in order to be useful in machine learning, the data is then preprocessed or "cleaned." Cleaning typically involves filtering out rows or columns containing data that has little or no influence on the outcome, removing (or replacing) missing values, and removing duplicate rows and outliers. Preprocessed data is then input to the model, which iterates over the data until it finds a best fit for the selected machine-learning algorithm in a process known as "training." Once training is complete, the model is "scored" to evaluate its accuracy. The finished model can take input structured in the same manner as the training data and predict what the output will be.

Building an ML model with other toolsets frequently involves writing lots of code: code to clean the data, code to train the model, code to implement the machine-learning algorithm, and code to score the model. Models built with Azure Machine Learning often require no code at all. The reason why is Azure Machine Learning Studio.

Azure Machine Learning Studio

Azure Machine Learning Studio ("ML Studio") is the browser-based tool that puts a face on Azure Machine Learning. Its purpose is to provide an easy-to-use graphical interface for importing and cleaning datasets, building, training, and scoring machine-learning models, quickly testing various algorithms and comparing results, and deploying finished models to the cloud as REST services so they can be used to build smarter software – for example, apps for mobile devices that incorporate machine-learning logic.

> "Thanks to Text Analytics by Azure Machine Learning, we are able to incorporate guest sentiment into our actionable guest feedback platform that delivers a comprehensive view of guest satisfaction and server performance." — *Al Pappa, Head of Business Intelligence, Ziosk*

ML Studio, pictured in Figure 5, provides an interactive machine-learning environment. On the left side of the screen is the *modules palette*, which provides access to more than 100 predefined modules for importing, cleaning, and transforming data; applying machine-learning algorithms; training, scoring, and evaluating models; inserting executable code written in R and Python; and much more. Models are built by dragging modules from the modules palette to the *canvas* in the center of the page and connecting them to define a workflow. On the far right is the *properties pane*, which exposes properties of the module that is currently selected so that module can be configured for the task at hand. For the **Split Data** module, for example, the user can specify the proportions for the split (for example, 50-50 or 80-20) as well as whether the split should be randomized, which is useful when splitting ordered datasets into two parts so a single dataset can be used for training and scoring.

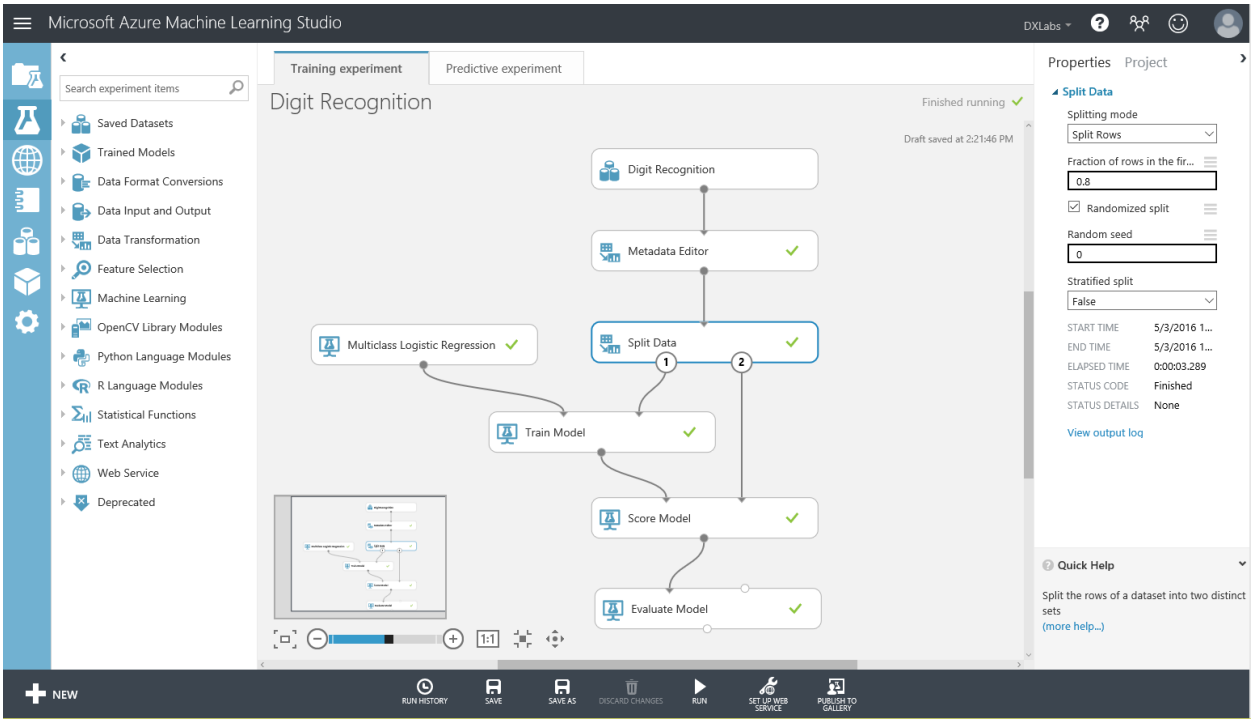


Figure 5: Azure Machine Learning Studio

Modules are the building blocks for machine-learning models in ML Studio. A complete list of modules can be found online. Here are just a few of the modules that are available:

- **Import Data** – Loads data from various external sources, including HTTP URLs, Hive queries, Azure SQL Database, and Azure Storage
- **Clean Missing Data** – Cleans data by removing missing values or replacing them with means, medians, or values computed using algorithms such as Multivariate Imputation using Chained Equations (MICE)
- **Moving Average Filter** – Smooths input data using a moving-average filter
- **Edit Metadata** – Used to label and identify feature columns, identify classes and categories, specify data types, and perform other metadata operations
- **Split Data** – Splits a dataset into two parts
- **Execute R Script** – Executes an R script at the designated point in the workflow
- **Execute Python Script** – Executes a Python script at the designated point in the workflow
- **Two-Class Support Vector Machine** – Implements a binary classification model using the Support Vector Machine (SVM) algorithm
- **Train Model** – Trains a classification or regression model using a training dataset
- **Score Model** – Scores predictions for a trained classification or regression model

- **Evaluate Model** – Evaluates a scored classification or regression model and optionally compares two models

Two-Class Support Vector Machine is one of 25 modules that represent commonly used machine-learning algorithms. A sampling of other algorithms available in ML Studio includes:

- **Multiclass Decision Forest** – Predicts categorical outputs using a hierarchical decision tree
- **Multiclass Logistic Regression** – Uses regression to predict categorical values by assigning a probability to each possible category
- **Multiclass Neural Network** – Uses a neural network to predict categorical values
- **Two-Class Bayes Point Machine** – Uses an enhanced Bayesian approach to linear classification to predict a binary output (for example, "spam" or "not spam") from provided inputs
- **Fast Forest Quantile Regression** – Uses regression to predict values for a specified number of quantiles, useful for predicting distributions
- **Linear Regression** – Uses univariate or multivariate linear regression to predict numeric values
- **Poisson Regression** – Uses regression to predict numeric values, typically counts, that conform to a Poisson distribution
- **K-Means Clustering** – Uses k-means clustering to identify groups in the input data

Figure 6 shows an actual workflow, taken directly from ML Studio, of a classification model that uses multiclass logistic regression to perform OCR. The dataset ("Digit Recognition") containing thousands of pixel patterns generated by digitizing handwritten digits is first uploaded to ML Studio. (In this example, the data was cleaned externally before being uploaded to ML Studio.) An **Edit Metadata** module defines one column of the dataset – the digit 0-9 represented by the values in the other columns – as the categorical target value, while **Split Data** is used to split the dataset so 80% can be used for training and 20% for scoring. The training output from **Split Data** goes into a **Train Model** module, which uses **Multiclass Logistic Regression** to classify a set of digitized inputs, while the scoring output goes to a **Score Model** module and then to **Evaluate Model**.

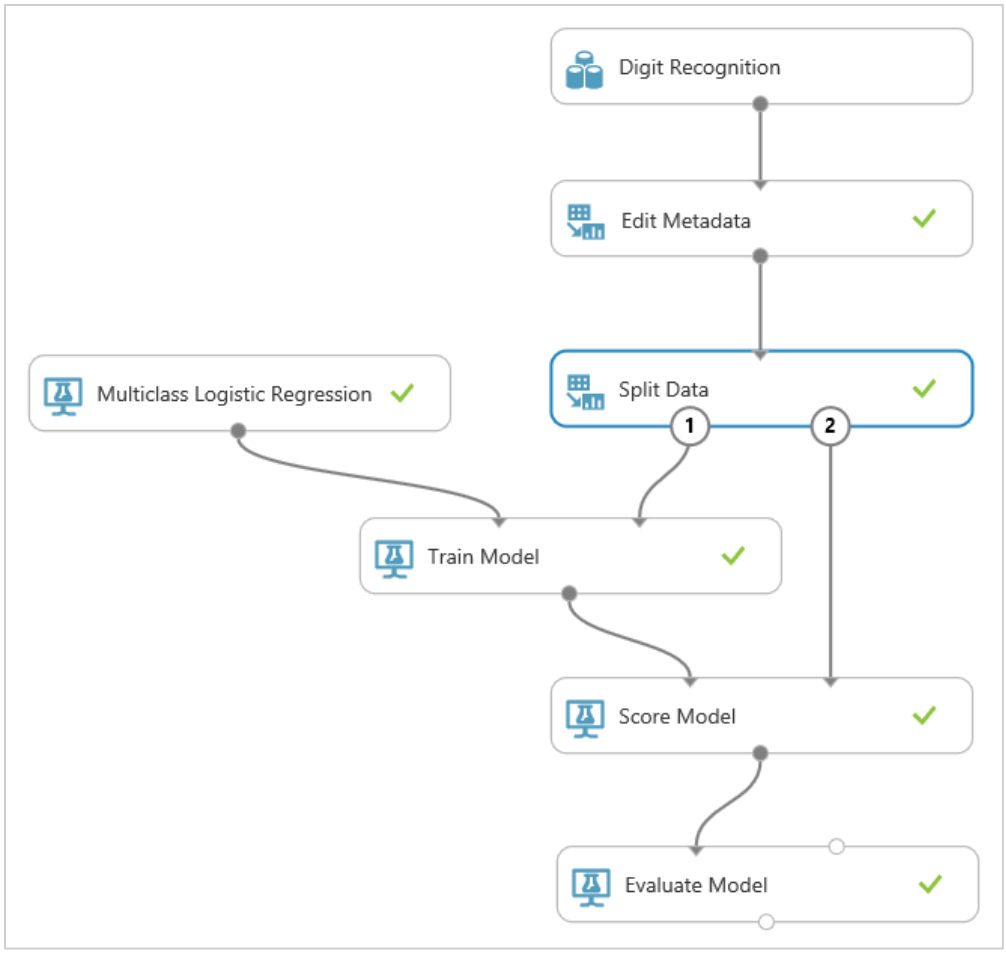


Figure 6: Machine-learning model for classifying digitized bit patterns as the digits 0-9

The **Evaluate Model** module reveals the accuracy of the model. Clicking the output port (the circle at the bottom of the module) and selecting *Visualize* from the ensuing menu produces the report shown in Figure 7, which indicates that based on the scoring data, the model can accurately predict, on average, more than 99% of the time which digit is represented by a two-dimensional array of pixels.

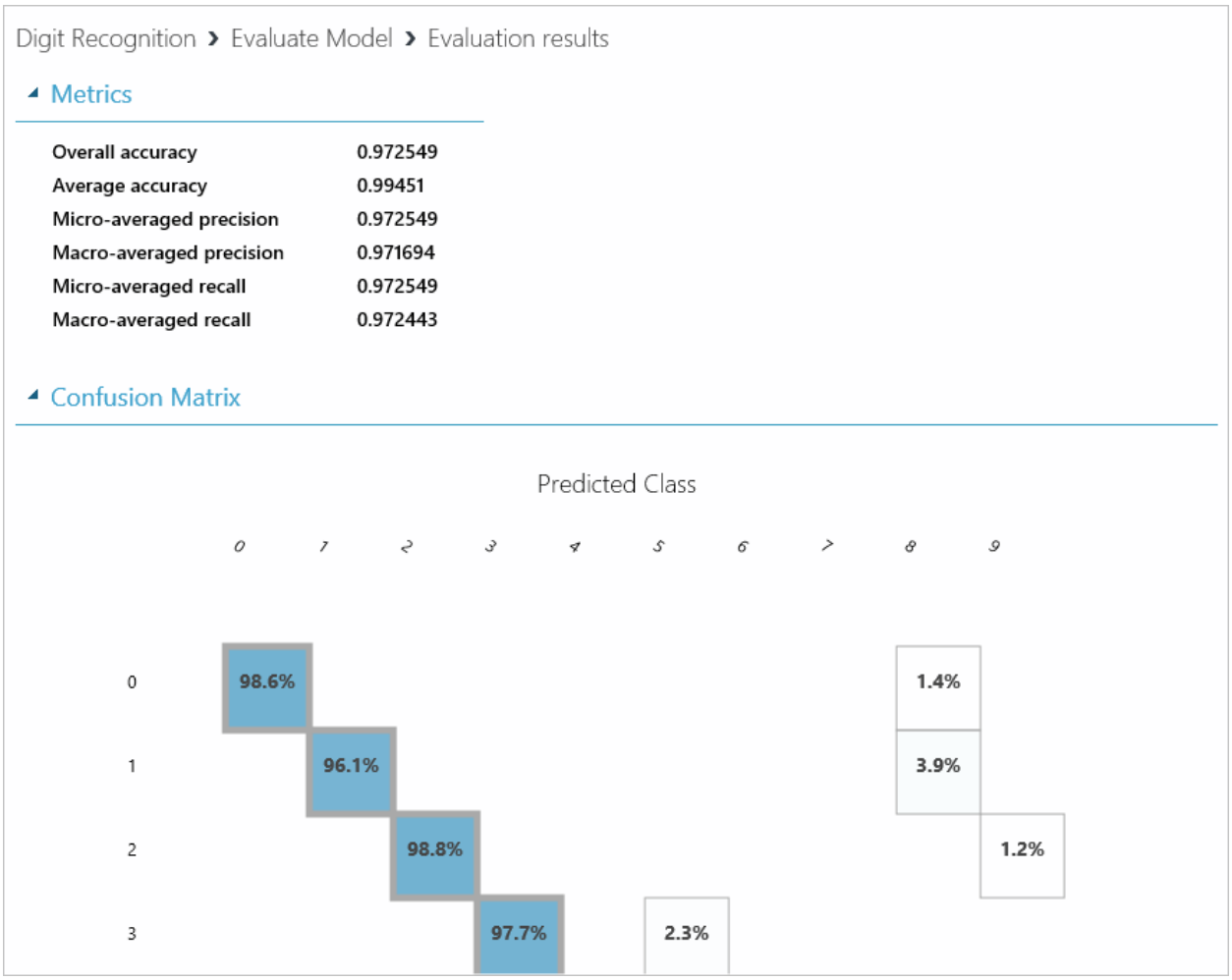


Figure 7: Results of evaluating the trained model

Once the model is trained and evaluated and the author is comfortable with its accuracy, it can be used to perform predictive analytics using an interactive test harness generated by ML Studio. But in real life, researchers and developers alike might need more. They might, for example, wish to employ the model in a mobile app or a Web site. For that, ML Studio makes it trivially easy to deploy the model as a Web service that can be reached via REST endpoints using virtually any modern programming language.

Deploying as a Web Service

Deploying a trained model as a Web service hosted in the Azure cloud requires little more than a couple of button clicks in ML Studio. Once deployed as a Web service, the model can be called over HTTPS. ML Studio even provides sample code tailored to each model in C#, Python, and R, as shown in Figure 8.

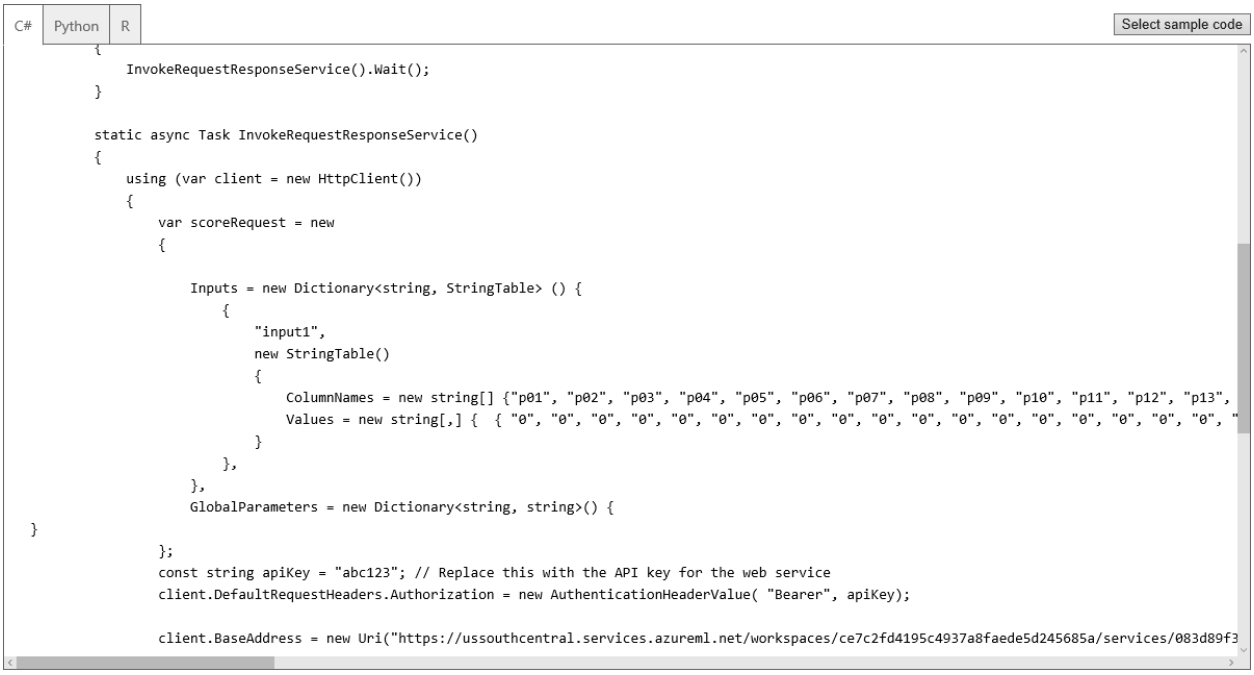


Figure 8: Sample code demonstrating how to call an Azure Machine Learning Web service

Data passed to and from the Web service is serialized using JavaScript Object Notation (JSON), which can be generated and consumed in a variety of programming languages. (Not all languages feature native JSON support, but virtually all have free, open-source JSON libraries available.) Use of the Web service is restricted to callers who transmit a secret API key that is unique to each model in the Authorization header of each request.

The fact that ML Studio natively supports ML models as Web services means you don't have to be a programmer with knowledge of HTTP, REST, and other technologies to make the model accessible over the Internet. As an added benefit, ML Studio also provides a downloadable Excel spreadsheet that can be used to invoke the Web service with input that you provide.

Summary

Azure Machine Learning, together with Azure ML Studio, represents a new way of thinking when it comes to building sophisticated machine-learning models for performing predictive analytics. Traditional tools that require considerable coding and expertise are still viable (and still widely used) in the research community, but there is now an alternative that makes machine learning available to a larger population of researchers and developers.

Machine learning touches lives every day, and its use will only increase as its capabilities and benefits become more widely understood. Azure Machine Learning lowers the bar for entry to

machine learning, allowing researchers and developers alike to focus on the intent of the model rather than writing code to bring it to life. The future of machine learning is bright. The time to learn about Azure Machine Learning is now.

Need Help with Azure Machine Learning?

Wintellect is an Azure Gold Partner with years of experience building cloud-enabled software and training others to do the same. We employ multiple Azure MVPs and we practice what we preach, having migrated our own internal infrastructure to Azure while realizing a cost savings of more than 70%. We also develop extensive Azure training content for Microsoft and deliver it to customers all over the world. Want to work with the Azure experts who Microsoft trusts to know Azure inside and out? Send us an email at consulting@wintellect.com or call 1-865-966-5528 for more info.