Cheat sheet: How to choose a MicrosoftML algorithm

2017/09/25 • 5 分钟阅读时长 • 作者

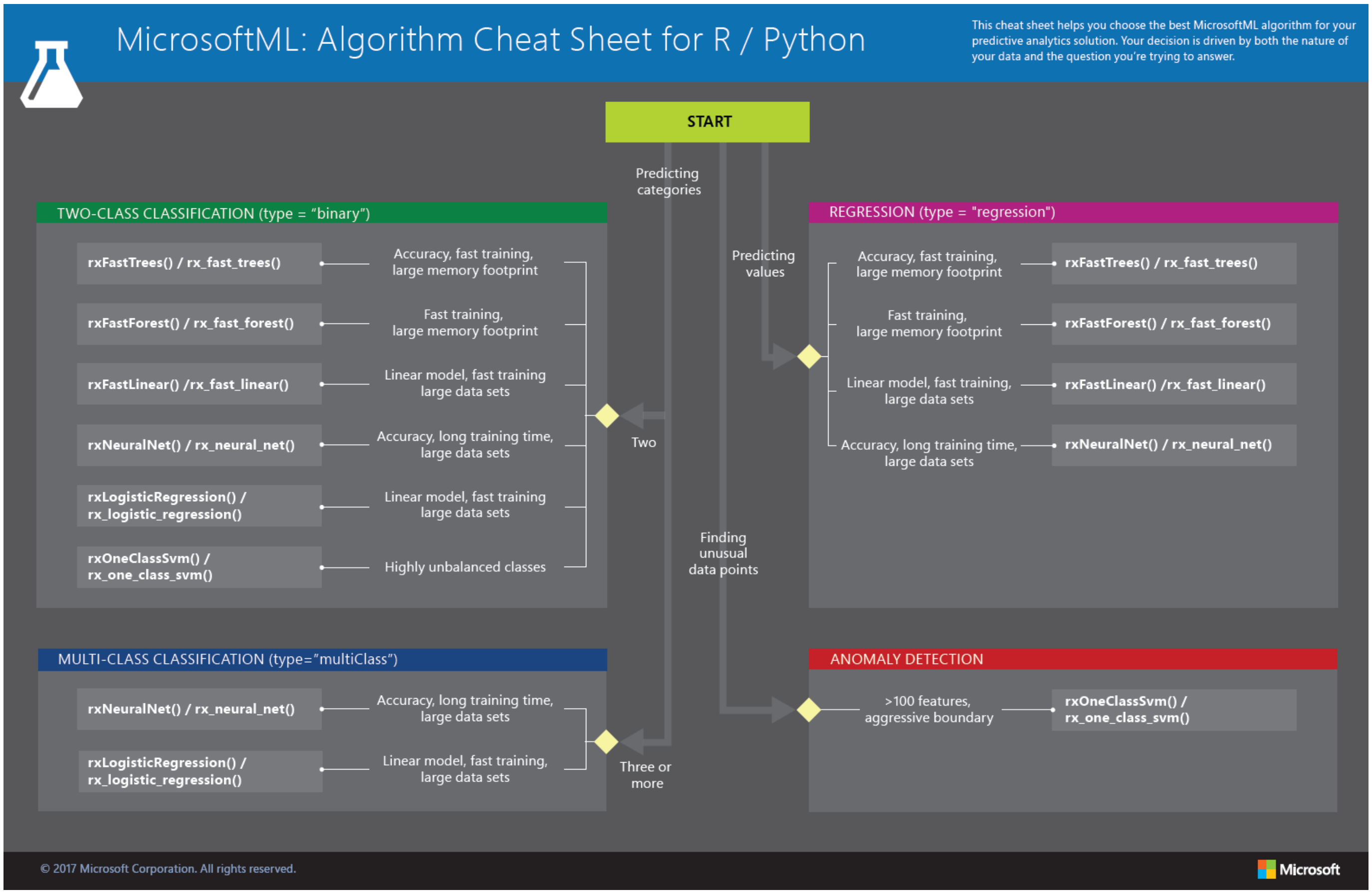
本文内容

The **MicrosoftML: Algorithm Cheat Sheet** helps you choose the right machine learning algorithm for a predictive analytics model when using Machine Learning Server. The algorithms are available in R or Python.

MicrosoftML provides a library of algorithms from the ***regression***, ***classification (two-class and multi-class)***, and ***anomaly detection*** families. Each is designed to address a different type of machine learning problem.

Download the MicrosoftML Algorithm Cheat Sheet

Download the cheat sheet here: MicrosoftML Package: Algorithm Cheat Sheet v2 (11x17 in.)



MicrosoftML: Algorithm Cheat Sheet for R / Python

This cheat sheet helps you choose the best MicrosoftML algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.

START

Predicting categories

Predicting values

Two

Finding unusual data points

Three or more

TWO-CLASS CLASSIFICATION (type = "binary")

- rxFastTrees() / rx_fast_trees() — Accuracy, fast training, large memory footprint
- rxFastForest() / rx_fast_forest() — Fast training, large memory footprint
- rxFastLinear() /rx_fast_linear() — Linear model, fast training large data sets
- rxNeuralNet() / rx_neural_net() — Accuracy, long training time, large data sets
- rxLogisticRegression() / rx_logistic_regression() — Linear model, fast training large data sets
- rxOneClassSvm() / rx_one_class_svm() — Highly unbalanced classes

REGRESSION (type = "regression")

- Accuracy, fast training, large memory footprint — rxFastTrees() / rx_fast_trees()
- Fast training, large memory footprint — rxFastForest() / rx_fast_forest()
- Linear model, fast training, large data sets — rxFastLinear() /rx_fast_linear()
- Accuracy, long training time, large data sets — rxNeuralNet() / rx_neural_net()

MULTI-CLASS CLASSIFICATION (type="multiClass")

- rxNeuralNet() / rx_neural_net() — Accuracy, long training time, large data sets
- rxLogisticRegression() / rx_logistic_regression() — Linear model, fast training, large data sets

ANOMALY DETECTION

- >100 features, aggressive boundary — rxOneClassSvm() / rx_one_class_svm()

© 2017 Microsoft Corporation. All rights reserved.

Microsoft

Download and print the **MicrosoftML: Algorithm Cheat Sheet** in tabloid size to keep it handy for guidance when choosing a machine learning algorithm.

MicrosoftML machine learning algorithms

This section contains descriptions of the machine learning algorithms contained in the Algorithm Cheat Sheet. The algorithms are available in R or Python. The R And Python names are provided in the format: `**R name/Python name**`.

Fast Linear model (SDCA)

The `rxFastTrees() / rx_fast_trees()` algorithm is based on the Stochastic Dual Coordinate Ascent (SDCA) method, a state-of-the-art optimization technique for convex objective functions. The algorithm can be scaled for use on large out-of-memory data sets due to a semi-asynchronized implementation that supports multithreaded processing. Several choices of loss functions are also provided and elastic net regularization is supported. The SDCA method combines several of the best properties and capabilities of logistic regression and SVM algorithms.

Tasks supported: binary classification, linear regression

OneClass SVM

The `rxOneClassSvm() / rx_one_class_svm()` algorithm is used for one-class anomaly detection. This is a type of unsupervised learning as its training set contains only examples from the target class and not any anomalous instances. It infers what properties are normal for the objects in the target class and from these properties predicts which examples are unlike these normal examples. This is useful as typically there are very few examples of network intrusion, fraud, or other types of anomalous behavior in training data sets.

Tasks supported: anomaly detection

Fast Tree

The `rxFastTrees() / rx_fast_trees()` algorithm is a high performing, state of the art scalable boosted decision tree that implements FastRank, an efficient implementation of the MART gradient boosting algorithm. MART learns an ensemble of regression trees, which is a decision tree with scalar values in its leaves. For binary classification, the output is converted to a probability by using some form of calibration.

Tasks supported: binary classification, regression

Fast Forest

The `rxFastForest() / rx_fast_forest()` algorithm is a random forest that provides a learning method for classification that constructs an ensemble of decision trees at training time, outputting the class that is the mode of the classes of the individual trees. Random decision forests can correct for the overfitting to training data sets to which decision trees are prone.

Tasks supported: binary classification, regression

Neural Network

The `rxNeuralNet() / rx_neural_net()` algorithm supports a user-defined multilayer network topology with GPU acceleration. A neural network is a class of prediction models inspired by the human brain. It can be represented as a weighted directed graph. Each node in the graph is called a neuron. The neural network algorithm tries to learn the optimal weights on the edges based on the training data. Any class of statistical models can be considered a neural network if they use adaptive weights and can approximate non-linear functions of their inputs. Neural network regression is especially suited to problems where a more traditional regression model cannot fit a solution.

Tasks supported: binary and multiclass classification, regression

Logistic regression

The `rxLogisticRegression() / rx_logistic_regression()` algorithm is used to predict the value of a categorical dependent variable from its relationship to one or more independent variables assumed to have a logistic distribution. If the dependent variable has only two possible values (success/failure), then the logistic regression is binary. If the dependent variable has more than two possible values (blood type given diagnostic test results), then the logistic regression is multinomial.

Tasks supported: binary and multiclass classification

Ensemble methods

The `rxEnsemble() / rx_emsemble()` algorithm uses a combination of learning algorithms to provide better predictive performance that the algorithms could individually. The approach is used primarily in the Hadoop/Spark environment for training across a multi-node cluster. But it can also be used in a single-node/local context.

Tasks supported: binary and multiclass classification, regression

More help with algorithms

For a list by category of all the machine learning algorithms available in the MicrosoftML package, see:

- MicrosoftML R functions
- MicrosoftML Python functions

Notes and terminology definitions for the machine learning algorithm cheat sheet

- The suggestions offered in this algorithm cheat sheet are approximate rules-of-thumb. Some can be bent and some can be flagrantly violated. This sheet is only intended to suggest a starting point. Don’t be afraid to run a head-to-head competition between several algorithms on your data. There is simply no substitute for understanding the principles of each algorithm and understanding the system that generated your data.

- Every machine learning algorithm has its own style or *inductive bias*. For a specific problem, several algorithms may be appropriate and one algorithm may be a better fit than others. But anticipating which will be the best fit beforehand is not always possible. In cases like these, several algorithms are listed together in the cheat sheet. An appropriate strategy would be to try one algorithm, and if the results are not yet satisfactory, try the others.

- Two categories of machine learning are supported by MicrosoftML: **supervised learning** and **unsupervised learning**.

  - In **supervised learning**, each data point is labeled or associated with a category or value of interest. The goal of supervised learning is to study many labeled examples like these, and then to be able to make predictions about future data points. All of the algorithms in MicrosoftML are supervised learners except `rxOneClassSvm()` used for anomaly detection.

  - In **unsupervised learning**, data points have no labels associated with them. Instead, the goal of an unsupervised learning algorithm is to organize the data in some way or to describe its structure. Only the `rxOneClassSvm()` algorithm used for anomaly detection is an unsupervised learner.

What's next?

Quickstarts for MicrosoftML shows how to use pre-trained models for sentiment analysis and image featurization.