Microsoft: DAT210x Programming with Python for Data Science

Bookmarks

- Start Here
- 1. The Big Picture
- 2. Data And Features
- 3. Exploring Data
- 4. Transforming Data
- 5. Data Modeling
- 6. Data Modeling II
- 7. Evaluating Data

Lecture: Confusion
Quiz

Lecture: Cross Validation
Quiz

Lecture: Power Tuning
Quiz

7. Evaluating Data > Lecture: Confusion > The Right Estimator

Bookmark

Choosing the Right Estimator

In this course, the terms model, class, estimator, and predictor have been used interchangeably, but never introduced formally. A model is any formula or algorithm designed to represent the mechanics of your data. SciKit-Learn implements many machine learning models as Python language classes, so they can be instantiated and used as objects. To keep the API clean, most of these classes follow a similar paradigm and interface, as a result of inheriting from a single estimator base-class. This is why SciKit-Learn's documentation referrers to all of the machine learning methods in their library as estimators. Lastly, when dealing with data that come with attributes you want to learn to predict, such as supervised learning problems, the estimators designed to handle this are called predictors.

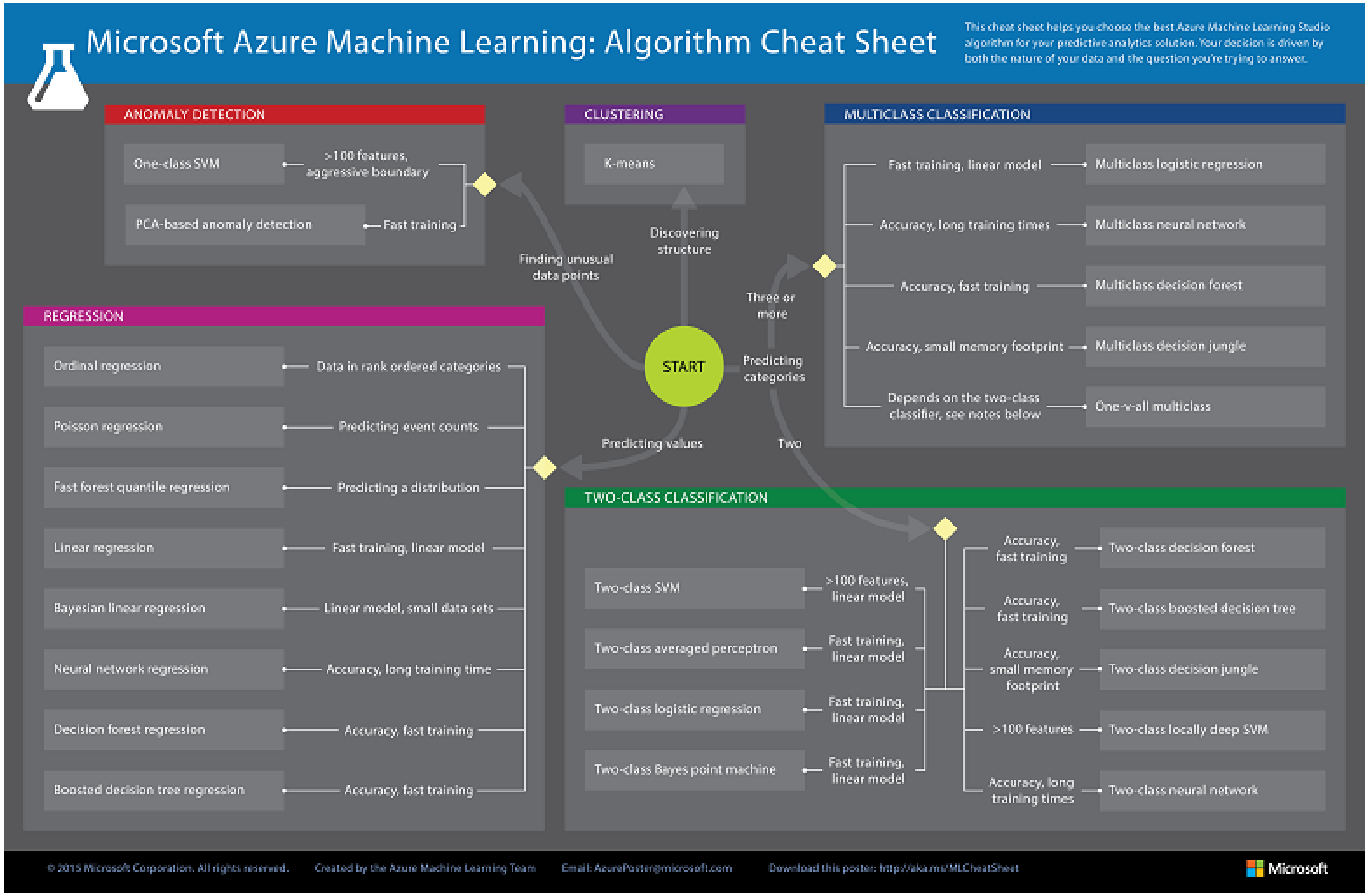
Having made it this far in the course, you now have a good foundation regarding the selection of estimators we have covered so far: isometric feature mapping, principle component analysis, k-means, k-nearest neighbors, linear regression, support vector machines, decision trees, and random forest. This is only a taste of what SciKit-Learn has to offer. With so many different categories of machine learning out there, many of which offer solutions to the same types of problems, it can get confusing trying to decide which one is most suitable for your data purposes. Developing an understanding of when to use which algorithm is an important step in your machine learning journey. Typically, algorithm choice is dictated by a balance of factors

- The dimensionality of your data
- The geometric nature of your data

- The types of features used to represent your data
- The number of training samples you have at your disposal
- The required training and prediction speeds needed for your purposes
- The predictive accuracy level desired
- How configurable you need your model to be
- And much more

As you continue to learn the ropes, a few groups have put together visualizations that will guide you to ensuring you've made an informed choice. The first one we'd like to draw your attention to is Microsoft's Azure Machine Learning Algorithm Cheat Sheet documentation page. Read through the entire article, from the definitions to the comments. This course has been structured around SciKit-Learn, but there are hundreds of other machine learning packages out there. Having exposure to a few of them, and gaining familiarity with the jargon and verbiage they use is beneficial to your future as a data scientist. Moreover, closer to the bottom of the Microsoft Azure page, they also have succinct notes regarding the details of some of the algorithms we've covered and some we haven't, such as logistic regression, boosted trees, neural networks, naive bayes, and other specialized algorithms.

The page also has a cheat-sheet available for download, a flow-chart that describes the circumstances under which you should use the different machine learning algorithms provided on the Azure platform:



Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio 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

ANOMALY DETECTION (Finding unusual data points)
- One-class SVM — >100 features, aggressive boundary
- PCA-based anomaly detection — Fast training

CLUSTERING (Discovering structure)
- K-means

MULTICLASS CLASSIFICATION (Predicting categories — Three or more)
- Multiclass logistic regression — Fast training, linear model
- Multiclass neural network — Accuracy, long training times
- Multiclass decision forest — Accuracy, fast training
- Multiclass decision jungle — Accuracy, small memory footprint
- One-v-all multiclass — Depends on the two-class classifier, see notes below

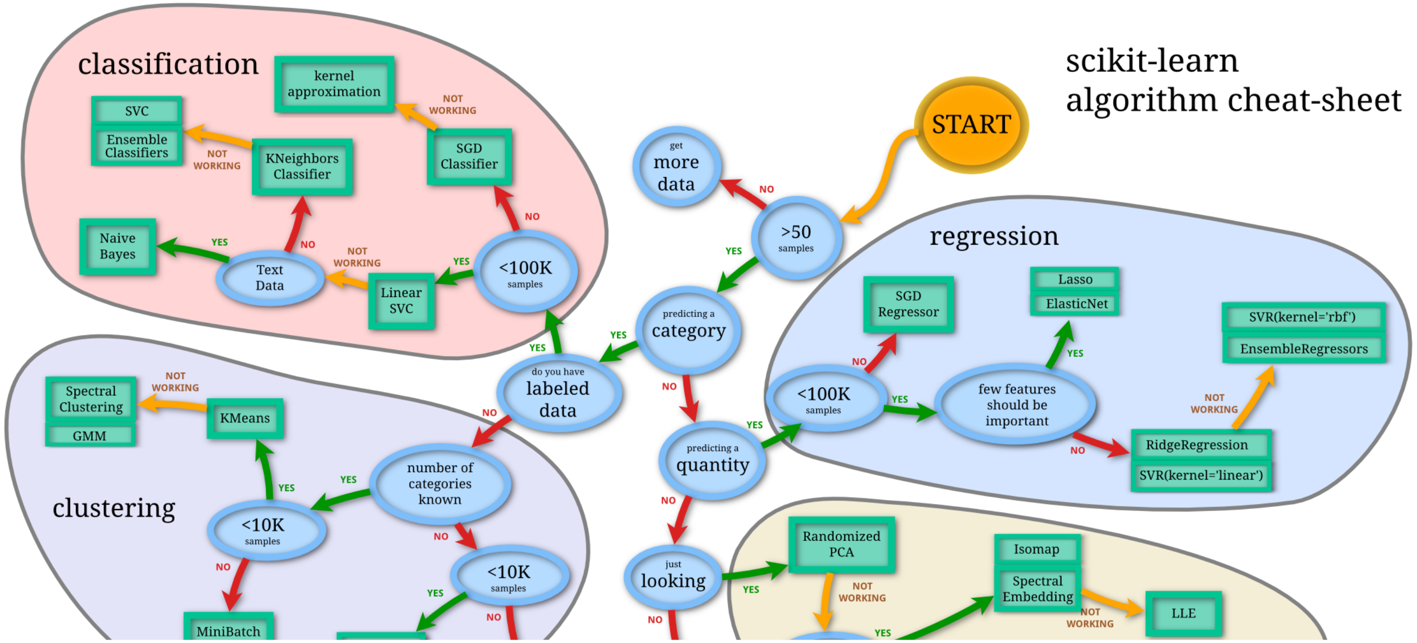
REGRESSION (Predicting values)
- Ordinal regression — Data in rank ordered categories
- Poisson regression — Predicting event counts
- Fast forest quantile regression — Predicting a distribution
- Linear regression — Fast training, linear model
- Bayesian linear regression — Linear model, small data sets
- Neural network regression — Accuracy, long training time
- Decision forest regression — Accuracy, fast training
- Boosted decision tree regression — Accuracy, fast training

TWO-CLASS CLASSIFICATION (Predicting categories — Two)
- Two-class SVM — >100 features, linear model
- Two-class averaged perceptron — Fast training, linear model
- Two-class logistic regression — Fast training, linear model
- Two-class Bayes point machine — Fast training, linear model
- Two-class decision forest — Accuracy, fast training
- Two-class boosted decision tree — Accuracy, fast training
- Two-class decision jungle — Accuracy, small memory footprint
- Two-class locally deep SVM — >100 features
- Two-class neural network — Accuracy, long training times

© 2015 Microsoft Corporation. All rights reserved. Created by the Azure Machine Learning Team Email: AzurePoster@microsoft.com Download this poster: http://aka.ms/MLCheatSheet

Microsoft

More immediately applicable to this course is a similar cheat-sheet created by Andreas Mueller, the release manager of, and serial contributor to SciKit-Learn. Rumor has it that this diagram was initially created for laughs before gaining viral acceptance and usage. Since your machine learning experience in this course until now has been based on SciKit-Learn rather than of AzureML, you might find this chart more immediately helpful:


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
get more data
NO
>50 samples
YES
predicting a category
YES
YES
do you have labeled data
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
YES
<100K samples
YES
few features should be important
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
NO
YES
predicting a quantity
NO
Spectral Clustering
GMM
NOT WORKING
KMeans
clustering
YES
YES
number of categories known
<10K samples
NO
NO
<10K samples
YES
NO
MiniBatch
just looking
YES
NO
Randomized PCA
NOT WORKING
YES
Isomap
Spectral Embedding
NOT WORKING
LLE