Tutorial on Machine Learning Environments

Thanh Le
(Dr. Vasant Honavar's Lab)

Email: txl252@ist.psu.edu

Agenda

- **Weka** - Using the Explorer GUI
- Programming basics in **Java** using **Weka**
- Programming basics in **Python** using **Scikit-learn**
- Other options

Weka

What is Weka?

WEKA
The University of Waikato

PENNSTATE 1855

- **Weka is a collection of machine learning algorithms for data mining tasks.**
- **The algorithms can either be:**
  - applied directly to a dataset
  - called from your own Java code.
- **Weka contains tools for:**
  - data pre-processing
  - classification
  - regression
  - clustering
  - association rules
  - visualization.
- **Weka is well-suited for developing new machine learning schemes**

Running Weka

PENNSTATE

- **From command line**
  - java -Xmx13g -classpath "./weka.jar" weka.gui.Main

Weka ARFF file format

PENNSTATE 1855

```
%
% 5. Number of Instances: 150 (50 in each of three classes)
%
% 6. Number of Attributes: 4 numeric, predictive attributes and the class
%
% 7. Attribute Information:
%    1. sepal length in cm
%    2. sepal width in cm
%    3. petal length in cm
%    4. petal width in cm
%    5. class:
%       -- Iris Setosa
%       -- Iris Versicolour
%       -- Iris Virginica
%
% 8. Missing Attribute Values: None
%
% Summary Statistics:
%              Min  Max   Mean    SD   Class Correlation
%    sepal length: 4.3  7.9   5.84  0.83    0.7826
%     sepal width: 2.0  4.4   3.05  0.43   -0.4194
%    petal length: 1.0  6.9   3.76  1.76    0.9490  (high!)
%     petal width: 0.1  2.5   1.20  0.76    0.9565  (high!)
%
% 9. Class Distribution: 33.3% for each of 3 classes.

@RELATION iris

@ATTRIBUTE sepallength  REAL
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  REAL
@ATTRIBUTE petalwidth   REAL
@ATTRIBUTE class    {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
```

Weka 3.6.3 - Explorer

Program Applications Tools Visualization Windows Help

Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose None Apply

Current relation

Relation: hepatitis
Instances: 155 Attributes: 20

Attributes

All | None | Invert | Pattern

No.	Name
1	AGE
2	SEX
3	STEROID
4	ANTIVIRALS
5	FATIGUE
6	MALAISE
7	ANOREXIA
8	LIVER_BIG
9	LIVER_FIRM
10	SPLEEN_PALPABLE
11	SPIDERS
12	ASCITES
13	VARICES
14	BILIRUBIN
15	ALK_PHOSPHATE
16	SGOT
17	ALBUMIN
18	PROTIME
19	HISTOLOGY
20	Class

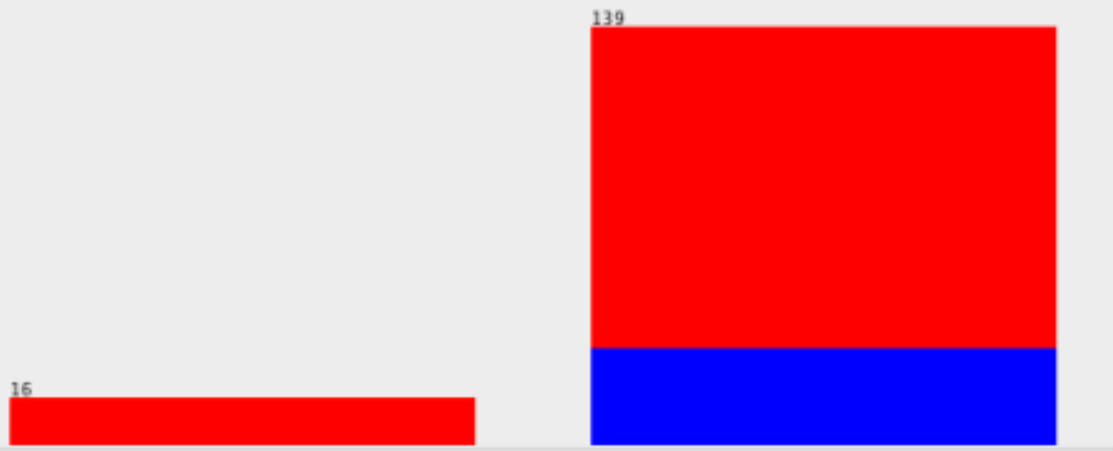
Remove

Selected attribute

Name: SEX Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count
1	male	16
2	female	139

Class: Class (Nom) Visualize All



Status

OK

Log x 0

Weka 3.6.3 - Explorer

Program Applications Tools Visualization Windows Help

Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose

Test options

Use trainin

Supplied t

Cross-vali

Percentage

M

(Nom) Class

Start

Result list (righ

weka.gui.GenericObjectEditor

weka.classifiers.bayes.NaiveBayes

About

Class for a Naive Bayes classifier using estimator classes.

More

Capabilities

debug False

displayModelInOldFormat False

useKernelEstimator False

useSupervisedDiscretization False

Open... Save... OK Cancel

Information

NAME
weka.classifiers.bayes.NaiveBayes

SYNOPSIS
Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data. For this reason, the classifier is not an UpdateableClassifier (which in typical usage are initialized with zero training instances) -- if you need the UpdateableClassifier functionality, use the NaiveBayesUpdateable classifier. The NaiveBayesUpdateable classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.

For more information on Naive Bayes classifiers, see

George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San

Status

OK

Log

x 0

Weka 3.6.3 - Explorer

Program Applications Tools Visualization Windows Help

Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose **NaiveBayes**

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) Class

Start | Stop

Result list (right-click for options)

19:32:56 - bayes.NaiveBayes

Classifier output

```
  precision          7.6386   7.6386

ALBUMIN
  mean               3.1454   3.9791
  std. dev.          0.5797   0.5547
  weight sum             27      112
  precision          0.1536   0.1536

PROTIME
  mean              43.2817   66.711
  std. dev.         16.4286  21.8107
  weight sum             18       70
  precision          2.3256   2.3256

HISTOLOGY
  no                    8.0     79.0
  yes                  26.0     46.0
  [total]              34.0    125.0



Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         131               84.5161 %
Incorrectly Classified Instances        24               15.4839 %
Kappa statistic                          0.5483
Mean absolute error                      0.1661
Root mean squared error                  0.3638
Relative absolute error                 50.3116 %
Root relative squared error             89.8459 %
Total Number of Instances              155

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.688     0.114       0.611     0.688     0.647      0.86     DIE
                 0.886     0.313       0.916     0.886     0.901      0.86     LIVE
Weighted Avg.    0.845     0.271       0.853     0.845     0.848      0.86

=== Confusion Matrix ===

   a   b   <-- classified as
  22  10 |   a = DIE
  14 109 |   b = LIVE
```

Status

OK

Log x 0

Weka 3.6.3 - Explorer

Program Applications Tools Visualization Windows Help

Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose NaiveBayes

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) Class

Start | Stop

Result list (right-click for options)

19:32:56 - bayes.NaiveBayes

- View in main window
- View in separate window
- Save result buffer
- Delete result buffer
- Load model
- Save model
- Re-evaluate model on current test set
- Visualize classifier errors
- Visualize tree
- Visualize margin curve
- Visualize threshold curve ▶ DIE / LIVE
- Cost/Benefit analysis ▶
- Visualize cost curve ▶

Classifier output

```
  precision          7.6386   7.6386

ALBUMIN
  mean               3.1454   3.9791
  std. dev.          0.5797   0.5547
  weight sum             27      112
  precision          0.1536   0.1536

PROTIME
  mean              43.2817   66.711
  std. dev.         16.4286  21.8107
  weight sum             18       70
  precision          2.3256   2.3256

HISTOLOGY
  no                    8.0     79.0
  yes                  26.0     46.0
  [total]              34.0    125.0



Time taken to build model: 0 seconds

n ===

                       131               84.5161 %
es                      24               15.4839 %
                        0.5483
                        0.1661
                        0.3638
                       50.3116 %
                       89.8459 %
                      155

 ===

te   Precision   Recall  F-Measure   ROC Area  Class
14       0.611    0.688      0.647       0.86   DIE
13       0.916    0.886      0.901       0.86   LIVE
53       0.853    0.845      0.848       0.86

14 109 |   b = LIVE
```

Status

OK

Log x 0

Weka Classifier Visualize: ThresholdCurve. (Class value DIE)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

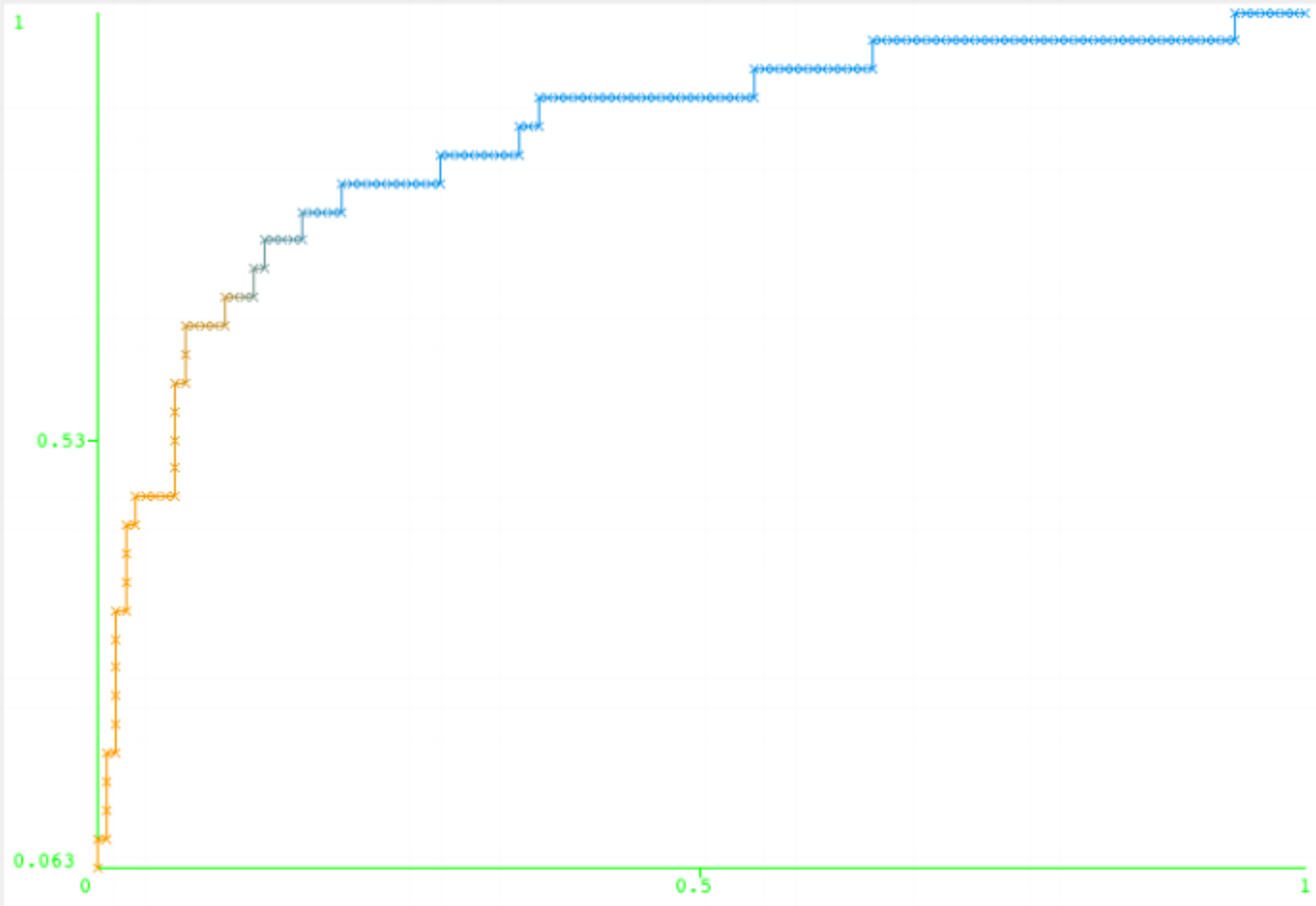
Colour: Threshold (Num)

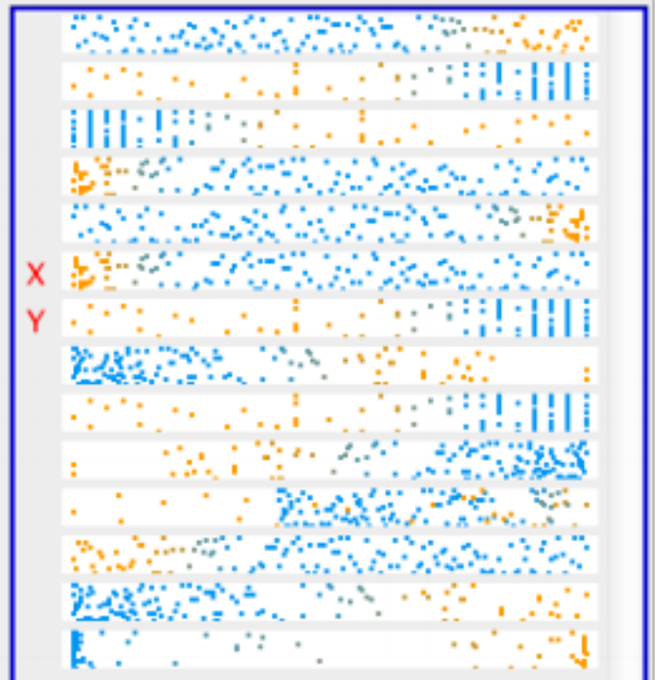
Select Instance

Reset | Clear | Open | Save

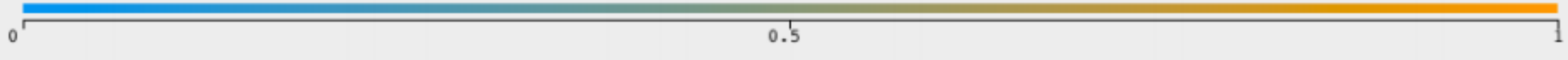
Jitter

Plot (Area under ROC = 0.8595)



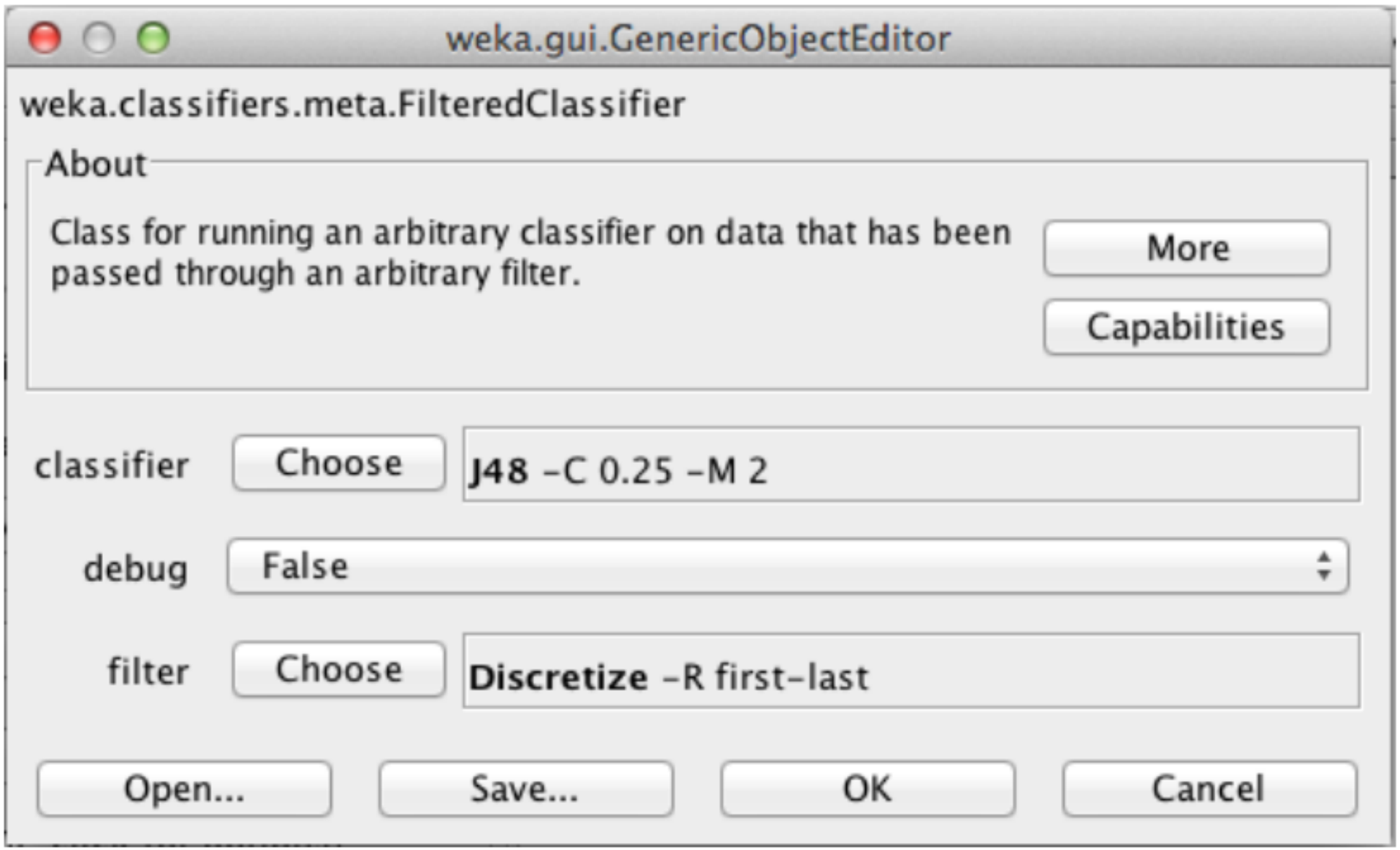


Class colour



Filtering data on-the-fly

PENNSTATE



Weka 3.6.3 - Explorer

Program Applications Tools Visualization Windows Help

Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose

.functions.supportVector.PolyKernel -C 250007 -E 1.0"

Test options

Use trainin

Supplied t

Cross-val

Percentage

M

(Nom) Class

Start

Result list (righ

19:32:56 - bay

weka.gui.GenericObjectEditor

weka.classifiers.functions.SMO

About

Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier.

More

Capabilities

buildLogisticModels	True
c	1.0
checksTurnedOff	False
debug	False
epsilon	1.0E-12
filterType	Normalize training data
kernel	Choose — PolyKernel -C 250007 -E 1.0
numFolds	-1
randomSeed	1
toleranceParameter	0.0010

Open... Save... OK Cancel

```
1                84.5161 %
4                15.4839 %
0.5483
0.1661
0.3638
0.3116 %
9.8459 %
5

ion   Recall  F-Measure  ROC Area  Class
11     0.688      0.647      0.86     DIE
16     0.886      0.901      0.86     LIVE
53     0.845      0.848      0.86

=== Confusion Matrix ===

  a   b   <-- classified as
 22  10 |   a = DIE
 14 109 |   b = LIVE
```

Status

OK

Log

x 0

Weka 3.6.3 - Explorer

Program Applications Tools Visualization Windows Help

Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier

Choose

'weka.attributeSelection.InfoGainAttributeEval \" -S \"weka.attributeSelection.Ranker -T -1.79769

Test options

Use trainin

Supplied t

Cross-vali

Percentage

(Nom) Class

Start

Result list (right-click for options)

19:32:56 - bayes.NaiveBayes
19:39:06 - functions.SMO
19:39:45 - functions.SMO
19:40:00 - functions.SMO
19:40:10 - functions.SMO
19:40:22 - functions.SMO
19:40:33 - functions.SMO
19:40:42 - functions.SMO
19:40:53 - functions.SMO
19:41:23 - trees.RandomForest
19:42:10 - trees.RandomForest
19:42:23 - trees.ADTree

Time

=== S
=== S

Corre
Incor
Kappa
Mean
Root
Relat
Root
Total

=== D

Weigh

=== C

```
   a   b   <-- classified as
  11  21 |   a = DIE
  16 107 |   b = LIVE
```

Class
DIE
LIVE

Status

OK

Log

x 0

weka.gui.GenericObjectEditor

weka.classifiers.meta.FilteredClassifier

About

Class for running an arbitrary classifier on data that has been passed through an arbitrary filter.

More

Capabilities

classifier Choose **NaiveBayes**

debug False

filter Choose **AttributeSelection** -E "weka.attributeSelection.In

Open... Save... OK Cancel

weka.gui.GenericObjectEditor

weka.filters.supervised.attribute.AttributeSelection

About

A supervised attribute filter that can be used to select attributes.

More

Capabilities

evaluator Choose **InfoGainAttributeEval**

search Choose **Ranker** -T -1.7976931348623157E308 -N 10

Open... Save... OK Cancel

Weka GUI is useful for experimenting with data quickly; however, sometime you want to implement your own classifiers

Java preliminaries for implementing a classifier

- Object - Class, Object, etc.
- Inheritance
- Basic data types and structures
- If you are not familiar with Java. Follow this tutorial

http://www.tutorialspoint.com/java/

You have 2 options

- Stand alone-classifier that can be invoked from command line or within your application code
- Implement classifier to be used in the Weka Explorer

Base classes

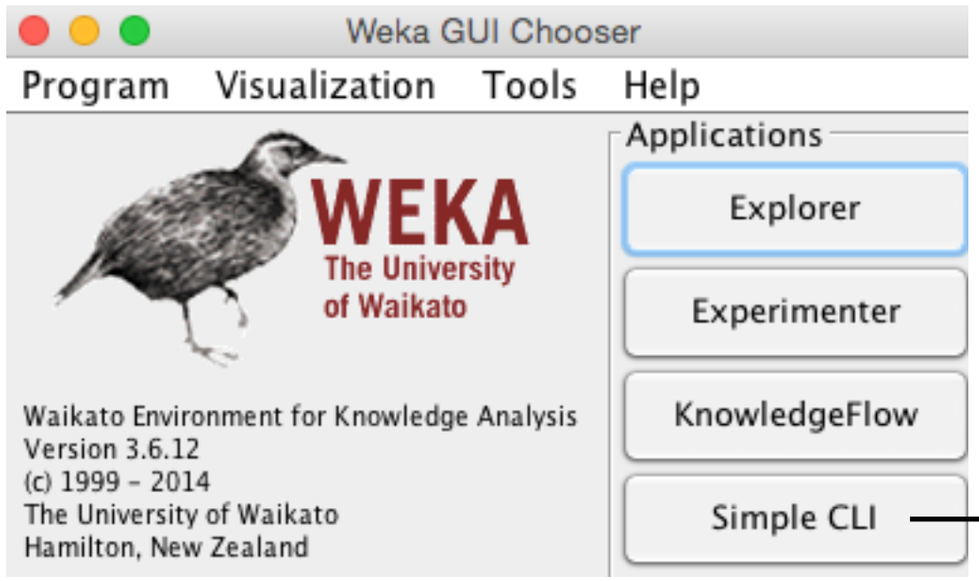
- All Classifiers extends **weka.classifiers.Classifier**
    - Weka already includes some basic classifiers:
        - weka.classifiers.bayes.NaiveBayes
        - weka.classifiers.tree
        - etc..
- If you find that your classifier is going to be closely related to a certain Classifier, extends that classifier instead of the generic parent class Classifier.

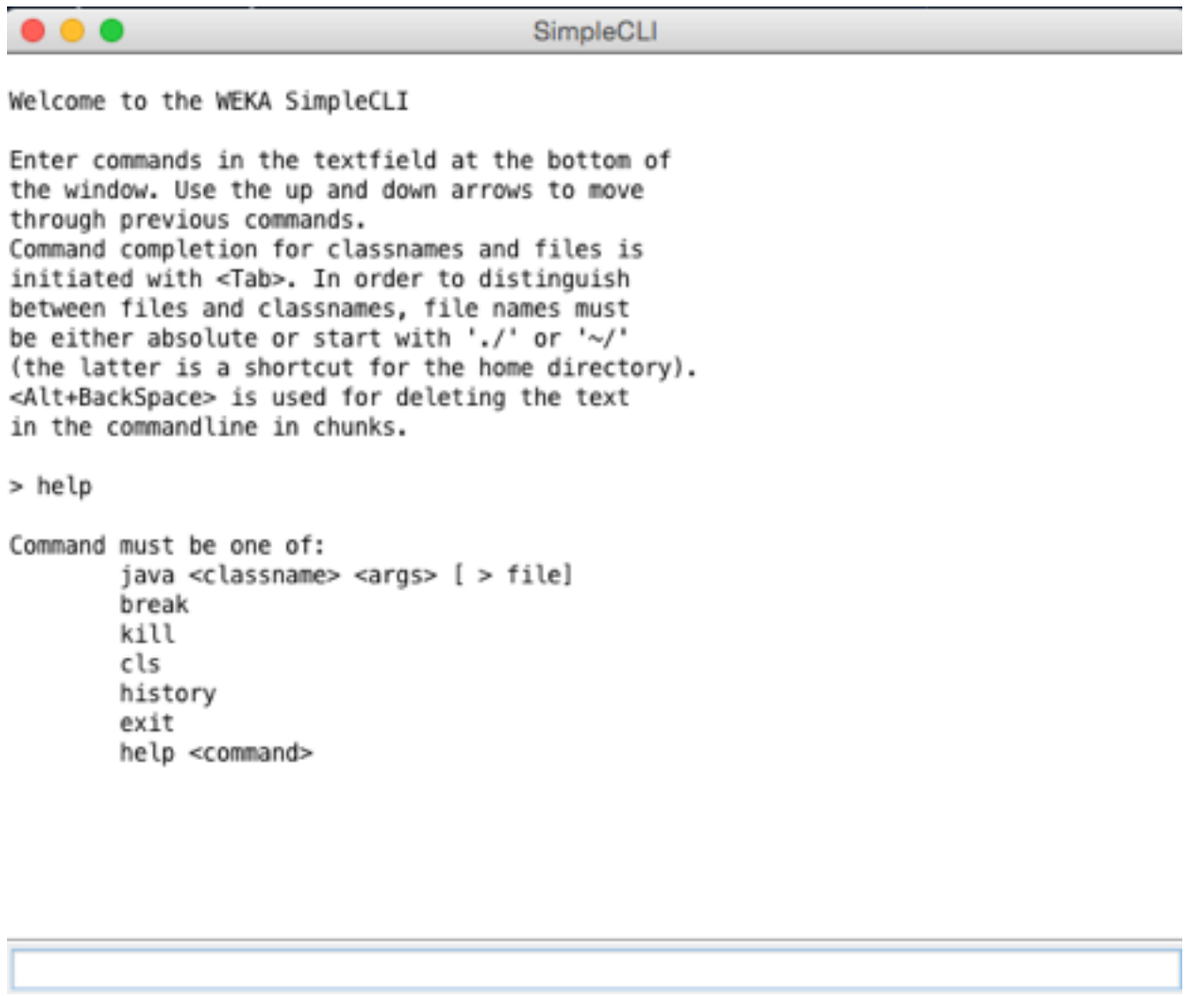
If you choose to implement a classifier to be run in the weka explorer

- You must implement at least 3 public methods
  - **void buildClassifier(Instances data)**
  - **double classifyInstance(Instance instance)**
  - **String [] getOptions()**

```java
import weka.core.Instance;
import weka.core.Instances;

/**
 *
 * @author txl252
 */
public class AClassifier extends weka.classifiers.Classifier {

    //Declare some parameters

    public void buildClassifier(Instances data){
        //build your model here
    }
    public double classifyInstance(Instance instance){
        //Use your model to classify an input instance
        return 0; //Return the predicted label for instance
    }
    public String[] getOptions(){
        //This function is used to show the list of parameters options
        return new String[0];
    }

    private void someHelperFunction(){
        //Other helper method for your classifier
    }
}
```

After implementing the required methods, you can proceed to build your classifier -> jar file





Inside weka CLI

- First you need to generate GenericObjectEditor (GOE) file

```
> java weka.gui.GenericPropertiesCreator \
$HOME/GenericPropertiesCreator.props \
$HOME/GenericObjectEditor.props
```

- Then edit the GPC and GOE files to include your custom Classifier. remember To set *UseDynamic* to false

Tell Weka where to discover your Class

```
java -classpath "weka.jar:dummy.jar" weka.gui.GUIChooser
```

At point, you will see your classifier in the Weka GUI and it is ready to be used with your data

Resources

- 1. **Weka Documentation**
    - http://weka.sourceforge.net/doc.stable/
- 2. **Guide to build custom Classifiers for Weka and integrate the custom Classifier into Weka GUI**
    - http://cns-classes.bu.edu/cn710/uploads/Main/WekaNotes.pdf
    - https://weka.wikispaces.com/GenericObjectEditor+%28developer+version%29

Scikit-learn

Python preliminaries

- Class & Objects (you maybe able to get away without it)
- Python scripting basics
- If you are unfamiliar with python, you can follow this tutorial
  - **https://docs.python.org/2/tutorial/index.html**

Setting up Python Environment

- **pip** is a python package manager. If you have python version >= 2.7.9, pip is already installed
- Otherwise, download this python script
  - https://bootstrap.pypa.io/get-pip.py
- The run the following command
  - > python get-pip.py

Now that pip is setup, you can install dependencies for scikitlearn

Scikit-learn requires:

- Python (>= 2.6 or >= 3.3),
- NumPy (>= 1.6.1),
- SciPy (>= 0.9).

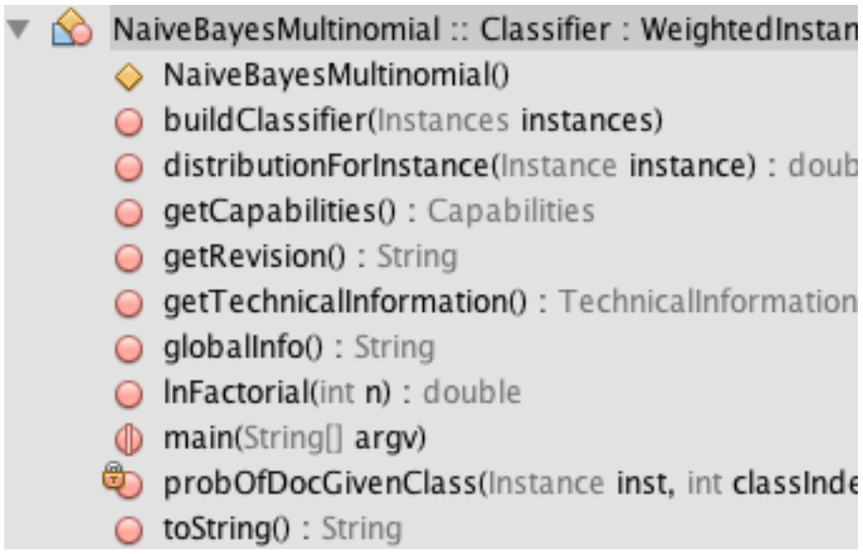
> pip install numpy
>pip install scipy

After complete installing both dependencies, you can install sklearn

```
pip install -U scikit-learn
```

The idea is very similar to weka.

Weka Multinomial NB



In fact, you will find that documentations for the 2 platforms closely resemble each other

Scikit Multinomial NB

Methods

`fit` (X, y[, sample_weight])	Fit Naive Bayes classifier according to X, y
`get_params` ([deep])	Get parameters for this estimator.
`partial_fit` (X, y[, classes, sample_weight])	Incremental fit on a batch of samples.
`predict` (X)	Perform classification on an array of test vectors X.
`predict_log_proba` (X)	Return log-probability estimates for the test vector X.
`predict_proba` (X)	Return probability estimates for the test vector X.
`score` (X, y[, sample_weight])	Returns the mean accuracy on the given test data and labels.
`set_params` (**params)	Set the parameters of this estimator.

There isn't a GUI for sklearn but the Python code is very intuitive

Let's take example of building a Multinomial NB classifier

1. import sklearn and build model

```
>>> from sklearn.naive_bayes import MultinomialNB
>>> clf = MultinomialNB().fit(X_train_tfidf, twenty_train.target)
```

Train matrix Label Vector

2. predict using the constructed model

```
>>> predicted = clf.predict(X_new_tfidf)
```

Predicted Label vector

3. Just like in weka, you can obtain probability est. for X

```
>>> prob = clf.predict_proba(X)
```

Resources

- **1. Python documentation**
    - https://docs.python.org/2/c-api/index.html
- **2. Scikit-learn documentation**
    - http://scikit-learn.org/stable/modules/classes.html

Other options

orange

Screenshots Download Docs Blog

Data Mining Fruitful and Fun

Open source data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Download Orange

The old version, Orange 2.7, is still available.

Python

http://orange.biolab.si/

Interactive workflows

Create your own interactive workflows to analyse and visualize your data.

Visualization

Orange is packed with different visualizations, from scatter plots, bar charts, trees, to dendrograms, networks and heat maps.

Large Toolbox

Over 100 widgets and growing. We cover most of standard data analysis tasks. Specialized add-ons available, like Orange Bioinformatics.

R-project

[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Mailing Lists
Bug Tracking
Development Site
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- **R version 3.2.4 (Very Secure Dishes) prerelease versions** will appear starting Monday 2016-02-29. Final release is scheduled for Thursday 2016-03-10.
- **R version 3.3.0 (Supposedly Educational) prerelease versions** will appear starting Monday 2016-03-14. Final release is scheduled for Thursday 2016-04-14.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, will take place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.

Matlab

MathWorks® Products Solutions Academia Support Community Events Company

MATLAB

Search MathWorks.com

Overview | Features | Code Examples | Videos | Webinars | What's New | Product Pricing

Trial software | Contact sales

The Language of Technical Computing

Explore new ideas

MATLAB® is the high-level language and interactive environment used by millions of engineers and scientists worldwide. It lets you explore and visualize ideas and collaborate across disciplines including signal and image processing, communications, control systems, and computational finance.

Put your ideas into action

You can use MATLAB in projects such as modeling energy consumption to build smart power grids, developing control algorithms for hypersonic vehicles, analyzing weather data to visualize the track and intensity of hurricanes, and running millions of simulations to pinpoint optimal dosing for antibiotics.

How will you use MATLAB?

» Explore the latest features

MATLAB Student

» Learn more

Question?