Analysis Tools and Libraries for BigData

Lecture 02

Abhijit Bendale

01/28/2014

Office Hours

- Terry Boult (Waiting to Confirm)
- Abhijit Bendale (Tue – 2:45 to 4:45 pm). Best if you email me in advance, but I will try to be in my office in this time.
  - abendale@vast.uccs.edu, abhijitbendale@gmail.com

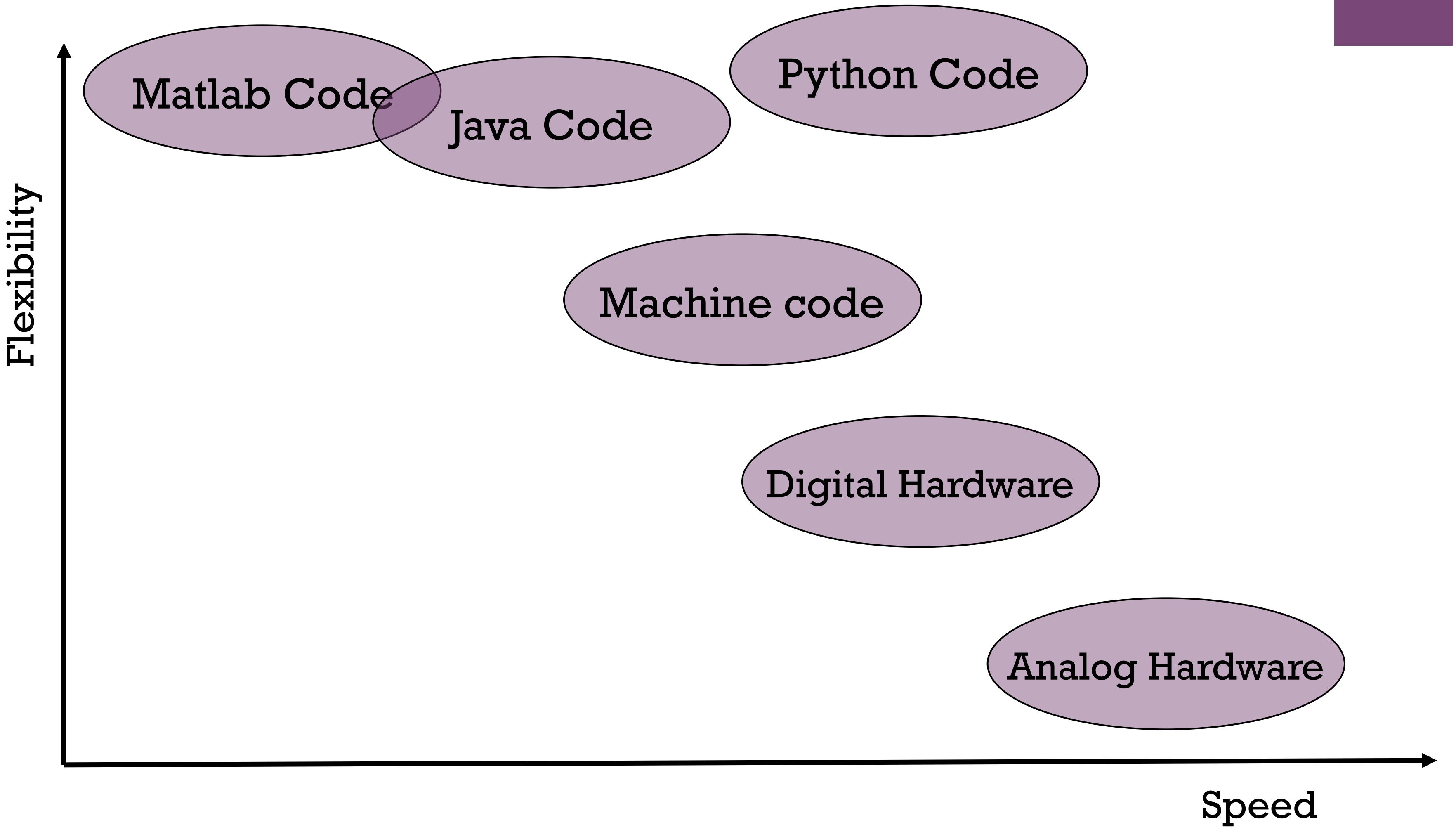
Agenda

- Introduction to various Statistics and Machine Learning libraries
  - Interface Languages: Python, Matlab, R, Java, C++
  - Machine Learning Tools : Scikits, Weka, R, Matlab
  - Case Study with Scikits
- Dataset Repositories
  - Amazon Public Datasets
  - Government Data Resources
- Tips for getting started on your **Semester Project**

Statistics and Machine Learning Tools

- What do you want to accomplish?
  - Basic Statistical Function
  - Advanced statistical and learning algorithm
  - Using plotting tools to understand trends in Data
  - Performance Evaluation Metrics
- Using the right tools for right task with right amount of abstraction

+ The speed/flexibility tradeoff



Theory Vs. Practice

- **Theoretician**: I want a polynomial-time algorithm which is guaranteed to perform arbitrarily well in "**all**" situations.

  - I prove theorems.

- **Practitioner**: I want a real-time algorithm that performs well on **my** problem.

  - I experiment.

- Approach for **BigData**: I want combining algorithms whose performance and speed is guaranteed relative to the performance and speed of their components.

  - You want to do both, or atleast the latter

Tools at hand..!

scikits learn
machine learning in Python

For use with Python

WEKA
The University of Waikato

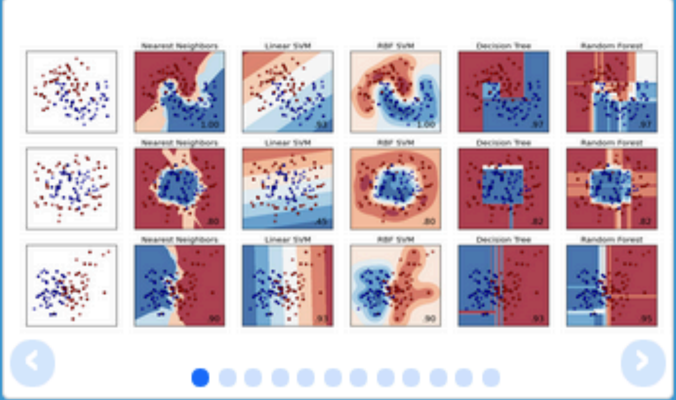
GUI based, for use with Java

dlib C++ Library

Machine Learning
With C++

De-facto
Scientific
Computing
Tool

Introduction to Scikits-Learn



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*. — *Examples*

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*. — *Examples*

Objectives

- Understanding Features and Feature Extraction
- Knowing basics of Classification / Regression
  - Supervised Classification
  - Unsupervised Classification
  - Understand Difference between linearly separable and non-linearly separable data

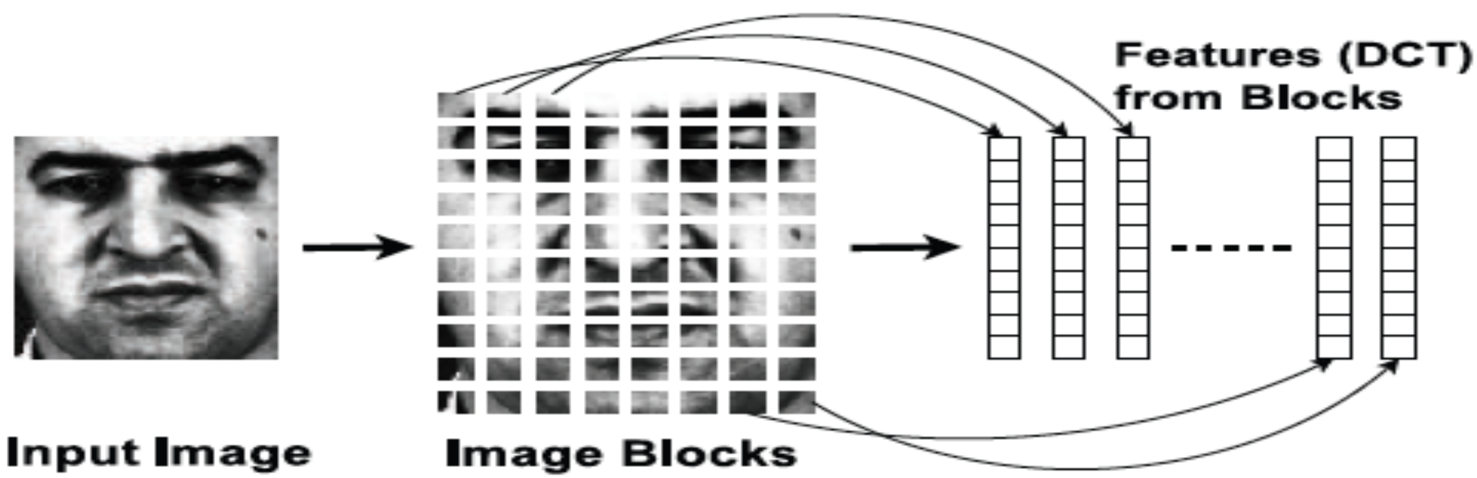
What is Machine Learning?

- A sub-field of Artificial Intelligence
- Often called as applied statistics
- Goal is to learn (understand) nature of given set of observation and build a predictive model for new observation

Features and Feature Extraction

- Most machine learning algorithms implemented in scikit-learn expect a numpy array as input X. The expected shape of X is (n_samples, n_features).

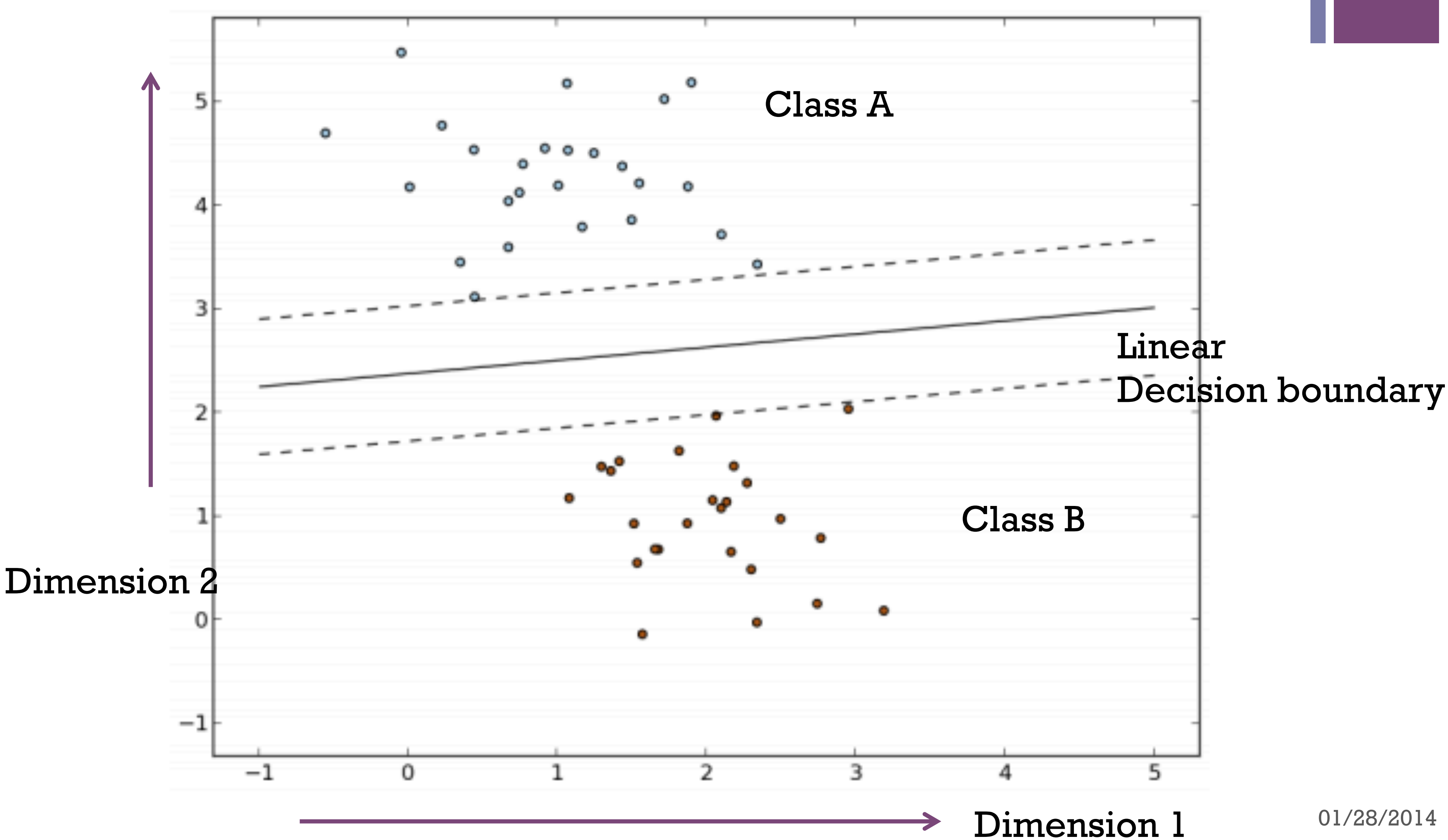
→ [10 11 25 128 220 245 …..]$_{nrows\ x\ ncols}$



Why feature extraction?

- Data often unstructured:
  - Text Documents
  - Sound
  - Climate measurements
  - Images
  - Videos
- Transform into more structured format

+ Feature Space and Linear Decision Boundary



Iris Flower Dataset

Iris Setosa

Iris Versicolor

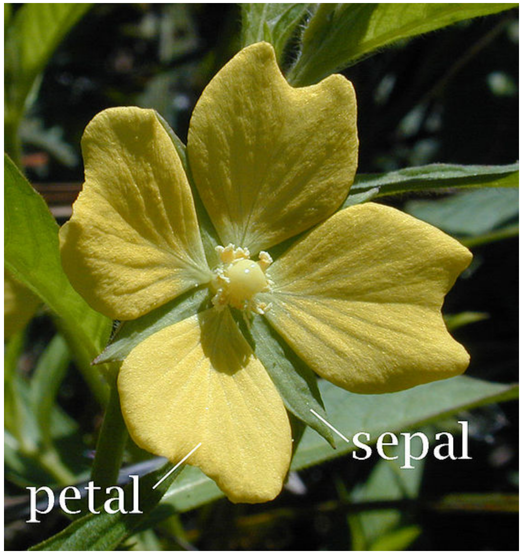
Iris Virginica

Measurements:

- Sepal length in cm
- Sepal width in cm
- Petal length in cm
- Petal width in cm

Target Classes

- 0 - Iris Setosa
- 1 – Iris Versicolor
- 2 – Iris Virginica



Iris Flower Dataset

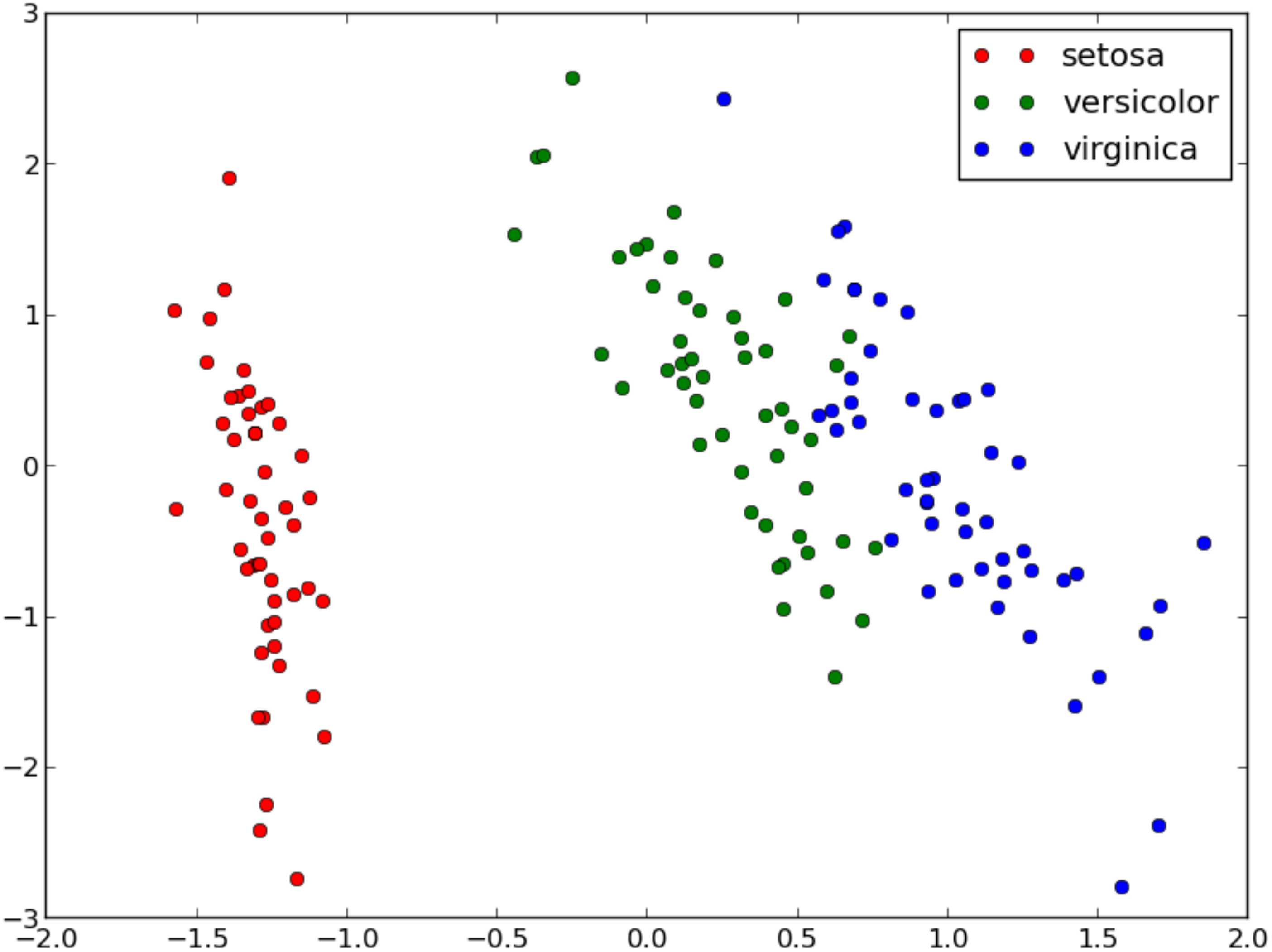
Iris Setosa

Iris Versicolor

Iris Virginica

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.1	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*
6.8	3.0	5.5	2.1	*I. virginica*
5.7	2.5	5.0	2.0	*I. virginica*
5.8	2.8	5.1	2.4	*I. virginica*
6.1	2.6	5.6	1.4	?

2D PCA on Iris Dataset

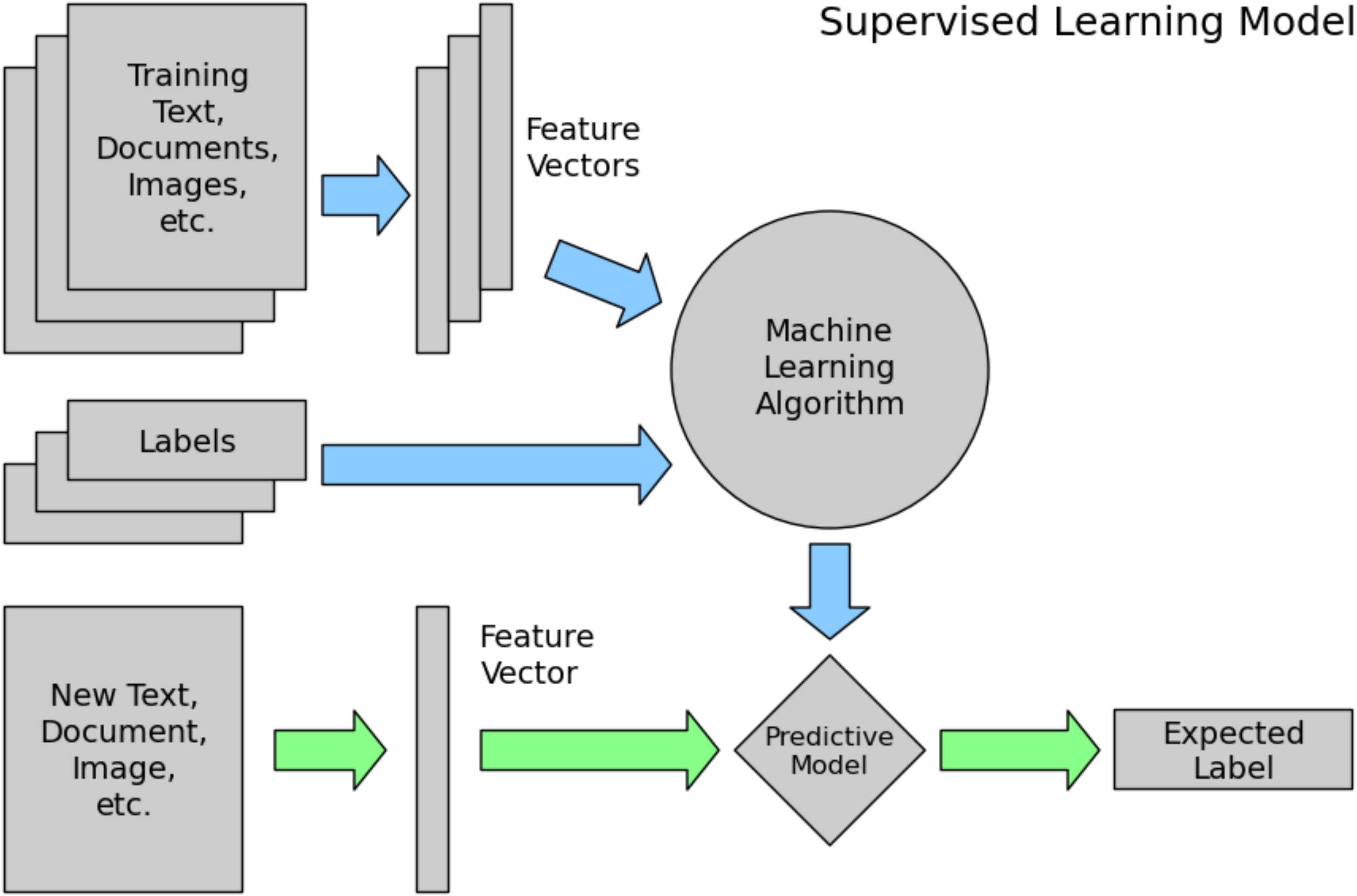


Transforming the data so that it is easy to operate on

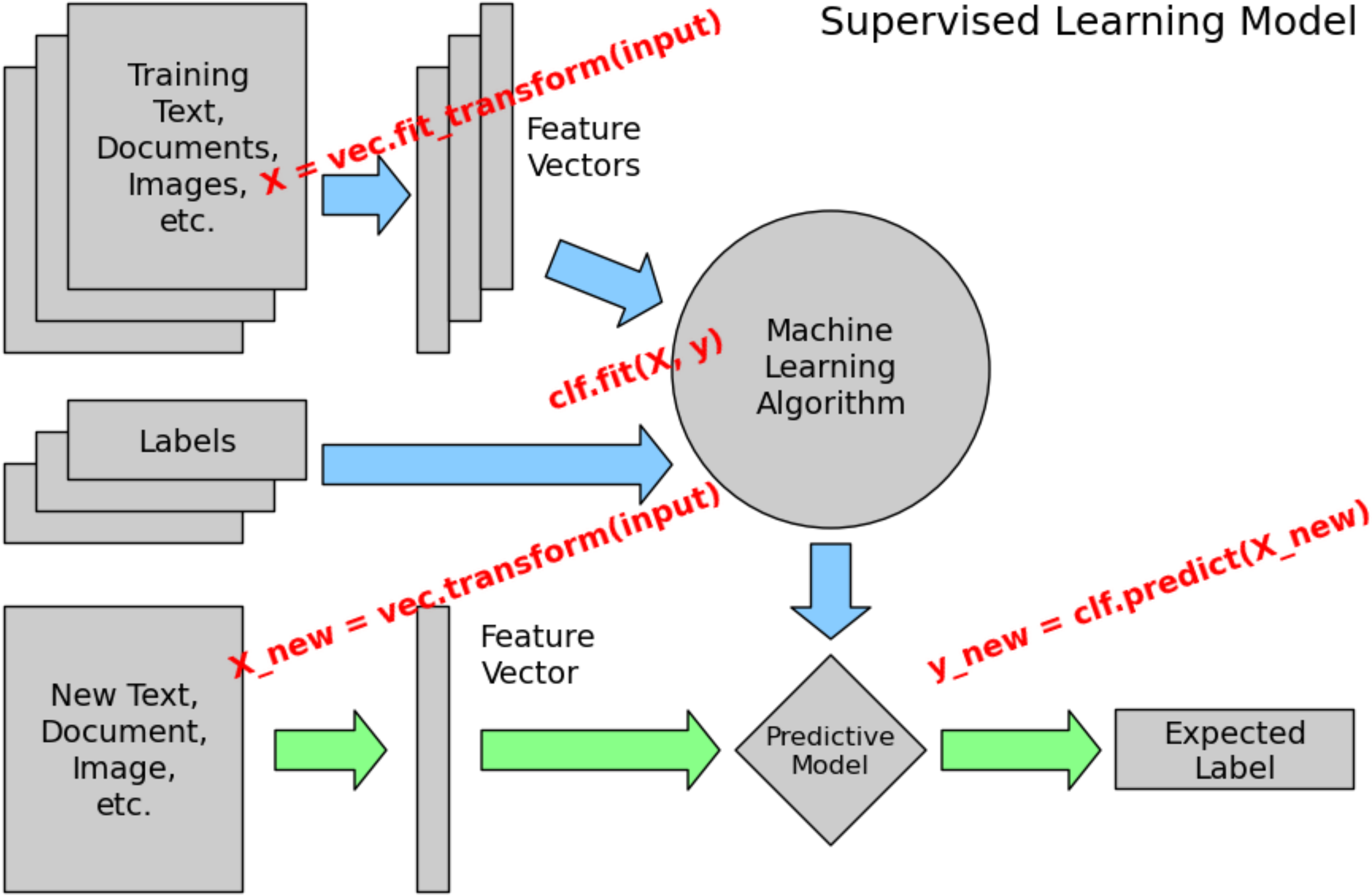
+ Re-using existing code-base


```
from itertools import cycle
import pylab as pl

from sklearn.datasets import load_iris
from sklearn.decomposition import PCA


def plot_2D(data, target, target_names):
    colors = cycle('rgbcmykw')
    target_ids = range(len(target_names))
    pl.figure()
    for i, c, label in zip(target_ids, colors, target_names):
        pl.plot(data[target == i, 0],
                data[target == i, 1], 'o',
                c=c, label=label)
    pl.legend(target_names)

#------------------------------------------------------------------------
# Load iris data
iris = load_iris()
X, y = iris.data, iris.target


#------------------------------------------------------------------------
# First figure: PCA
pca = PCA(n_components=2, whiten=True).fit(X)
X_pca = pca.transform(X)
plot_2D(X_pca, iris.target, iris.target_names)
```

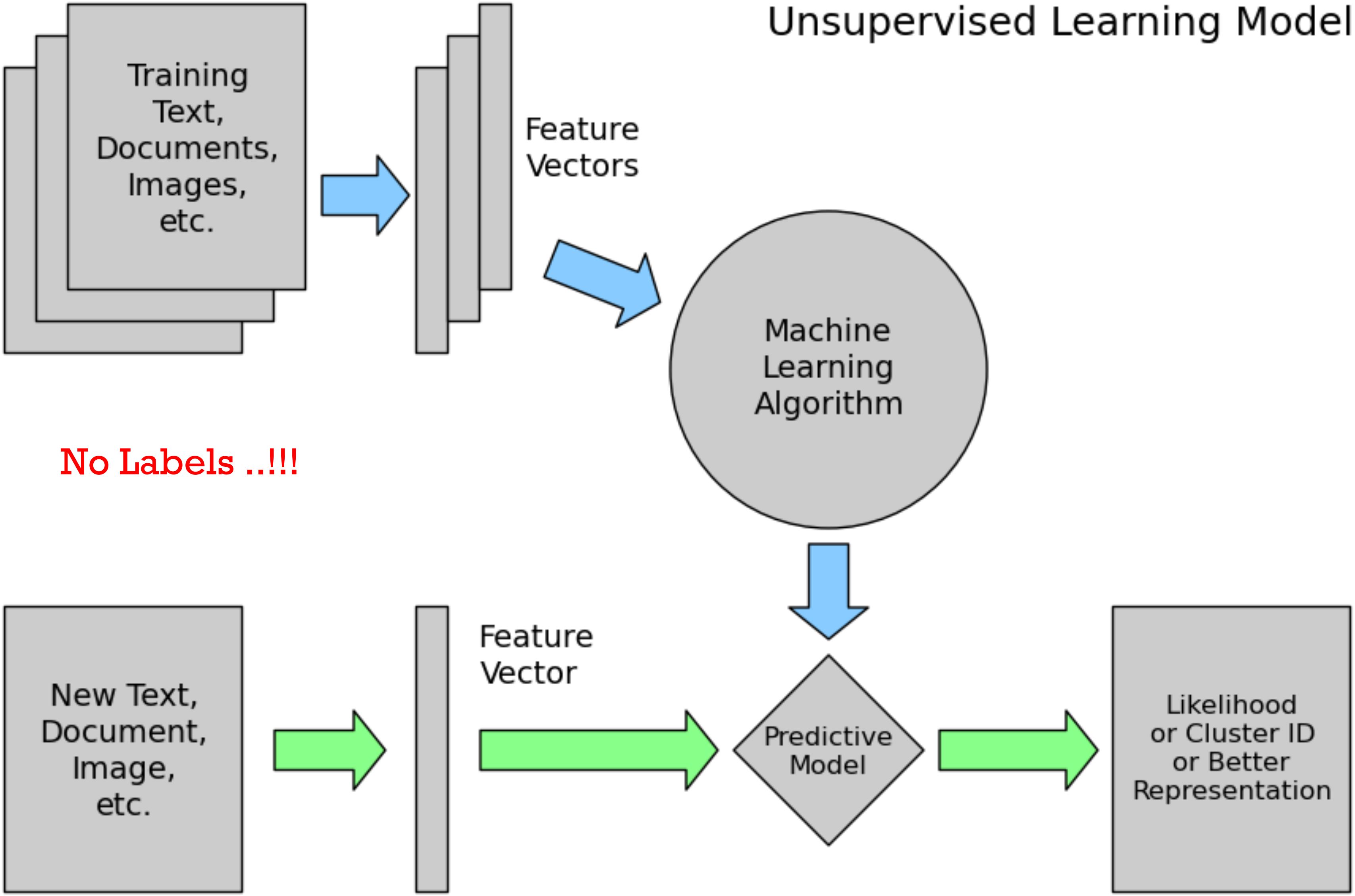
Supervised Learning Model



Supervised Learning Model



Unsupervised Learning Model



No Labels ..!!!

2.6. Hyperparameters, training set, test set and overfitting

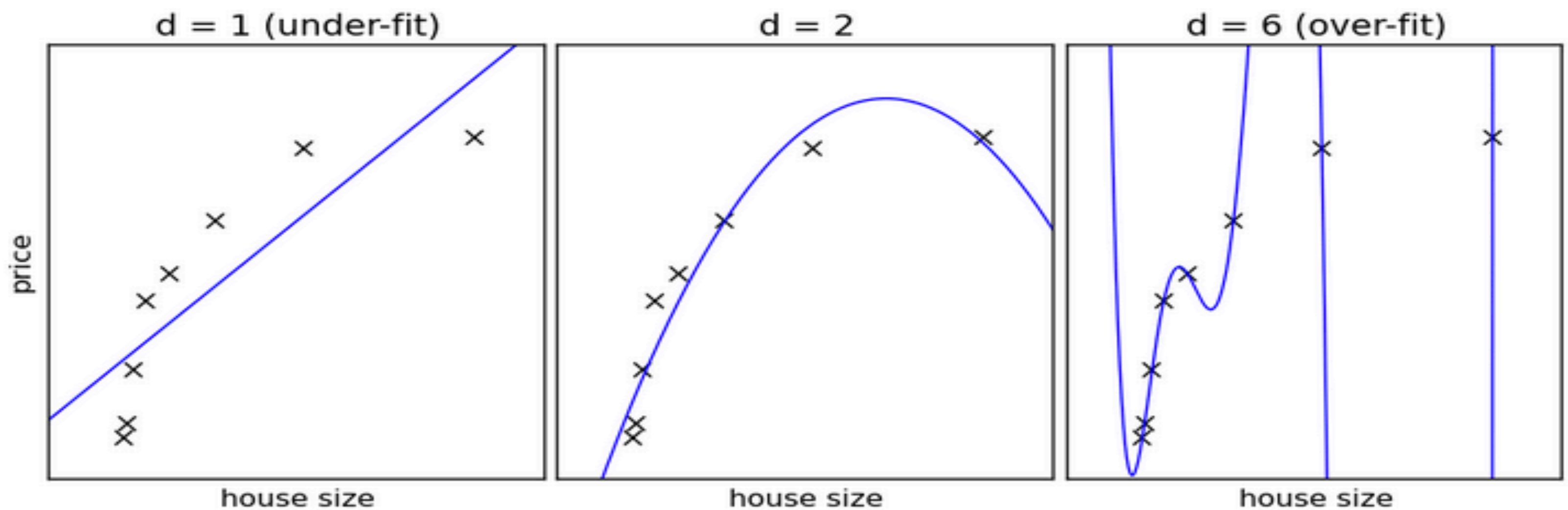
The above SVM example displays an example of *hyperparameters*, which are model parameters set before the training process. For example, when using an RBF model, we choose the kernel coefficient `gamma` before fitting the data. We must be able to then evaluate the goodness-of-fit of our model given this choice of hyperparameter.

The most common mistake beginners make when training statistical models is to evaluate the quality of the model on the same data used for fitting the model:

> If you do this, **you are doing it wrong!**

2.6.1. The overfitting issue

Evaluating the quality of the model on the data used to fit the model can lead to *overfitting*. Consider the following dataset, and three fits to the data (we'll explore this example in more detail in the *next section*).



Examples of over-fitting and under-fitting a two-dimensional dataset.

2.6.2. Solutions to overfitting

The solution to this issue is twofold:

1. Split your data into two sets to detect overfitting situations:
   - one for training and model selection: the **training set**
   - one for evaluation: the **test set**
2. Avoid overfitting by using simpler models (e.g. linear classifiers instead of gaussian kernel SVM) or by increasing the regularization parameter of the model if available (see the docstring of the model for details)

scikits learn

Machine learning 101 | Classification | Regression | Exercises | Examples

Previous: 1. Tutorial Setu... | Next: 3. Machine Learn...

2. Machine Learning 101: General Concepts

This page

Objectives

By the end of this section you will

1. Know how to extract features from real-world data in order to perform machine learning tasks.
2. Know the basic categories of *supervised learning*, including *classification* and *regression* problems.
3. Know the basic categories of *unsupervised learning*, including dimensionality reduction and clustering.
4. Understand the distinction between linearly separable and non-linearly separable data.

In addition, you will know several tools within scikit-learn which can be used to accomplish the above tasks.

In this section we will begin to explore the basic principles of machine learning. Machine Learning is about building **programs with tunable parameters** (typically an array of floating point values) that are adjusted automatically so as to improve their behavior by **adapting to previously seen data**

More details on sklearn website
There is ready made code for you to try out..!

sklearn.metrics: Metrics

See the *Model evaluation: quantifying the quality of predictions* section and the *Pairwise metrics, Affinities and Kernels* section of the user guide for further details.

The **sklearn.metrics** module includes score functions, performance metrics and pairwise metrics and distance computations.

Model Selection Interface

See the *The scoring parameter: defining model evaluation rules* section of the user guide for further details.

metrics.make_scorer(score_func[, ...])	Make a scorer from a performance metric or loss function.

Classification metrics

See the *Classification metrics* section of the user guide for further details.

metrics.accuracy_score(y_true, y_pred[, ...])	Accuracy classification score.
metrics.auc(x, y[, reorder])	Compute Area Under the Curve (AUC) using the trapezoidal rule
metrics.average_precision_score(y_true, y_score)	Compute average precision (AP) from prediction scores
metrics.classification_report(y_true, y_pred)	Build a text report showing the main classification metrics
metrics.confusion_matrix(y_true, y_pred[, ...])	Compute confusion matrix to evaluate the accuracy of a classification
metrics.f1_score(y_true, y_pred[, labels, ...])	Compute the F1 score, also known as balanced F-score or F-measure
metrics.fbeta_score(y_true, y_pred, beta[, ...])	Compute the F-beta score
metrics.hamming_loss(y_true, y_pred[, classes])	Compute the average Hamming loss.
metrics.hinge_loss(y_true, pred_decision[, ...])	Average hinge loss (non-regularized)

sklearn.svm.SVC

class `sklearn.svm.`**`SVC`**(*C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, random_state=None*)

C-Support Vector Classification.

The implementations is a based on libsvm. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to dataset with more than a couple of 10000 samples.

The multiclass support is handled according to a one-vs-one scheme.

For details on the precise mathematical formulation of the provided kernel functions and how gamma, coef0 and degree affect each, see the corresponding section in the narrative documentation: *Kernel functions*.

Parameters :

C : float, optional (default=1.0)

Penalty parameter C of the error term.

Reading the documentation

kernel : string, optional (default='rbf')

Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to precompute the kernel matrix.

degree : int, optional (default=3)

Degree of the polynomial kernel function ('poly'). Ignored by all other kernels.

gamma : float, optional (default=0.0)

Kernel coefficient for 'rbf', 'poly' and 'sigm'. If gamma is 0.0 then 1/n_features will be used instead.

Installing scikit-learn

There are different ways to get scikit-learn installed:

- Install the version of scikit-learn provided by your *operating system or Python distribution*. This is the quickest option for those who have operating systems that distribute scikit-learn.
- *Install an official release*. This is the best approach for users who want a stable version number and aren't concerned about running a slightly older version of scikit-learn.
- *Install the latest development version*. This is best for users who want the latest-and-greatest features and aren't afraid of running brand-new code.

Note: If you wish to contribute to the project, it's recommended you *install the latest development version*.

Installing an official release

Getting the dependencies

Installing from source requires you to have installed Python (>= 2.6), NumPy (>= 1.3), SciPy (>= 0.7), setuptools, Python development headers and a working C++ compiler. Under Debian-based operating systems, which include Ubuntu, you can install all these requirements by issuing:

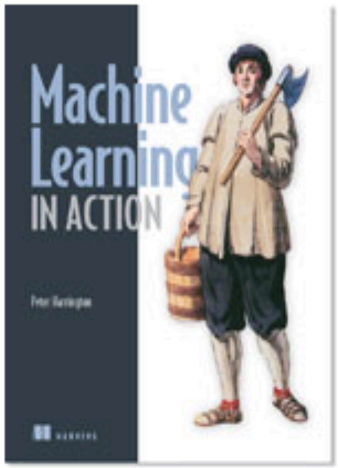
```
sudo apt-get install build-essential python-dev python-numpy python-setuptools python-scipy libatla
s-dev libatlas3-base
```

Note: In order to build the documentation and run the example code contains in this documentation you will need matplotlib:

```
sudo apt-get install python-matplotlib
```

Manning Publications Co.

Home | Ordering Info | Shopping Cart



Machine Learning in Action

Peter Harrington

April, 2012 | 384 pages
ISBN: 9781617290183

ADD TO CART **$44.99** pBook + eBook (includes PDF, ePub, and Kindle)

ADD TO CART **$35.99** eBook Only (includes PDF, ePub, and Kindle)

Browse all our mobile format ebooks.

RESOURCES

Look Inside

- Preface
- About this book
- Table of Contents
- Index
- Errata

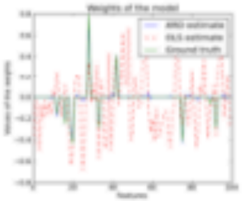
Resources

- Author Online
- **Machine Learning with Watson: The Greatest Achievement of Artificial Intelligence to Date (PDF)**
- **Stochastic Gradient Ascent (PDF)**
- **The Premise of Machine Learning (PDF)**

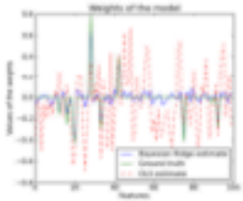
Downloads

- Source code (34.2 MB)
- Sample chapter 1
- Sample chapter 7

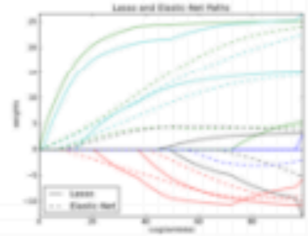
SUMMARY



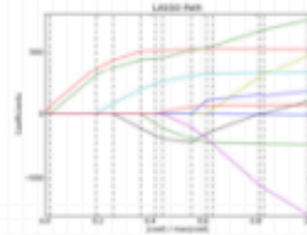
Automatic Relevance Determination Regression (ARD)



Bayesian Ridge Regression

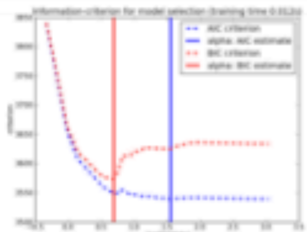


Lasso and Elastic Net

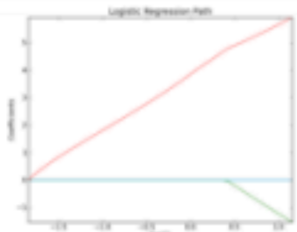


Lasso path using LARS

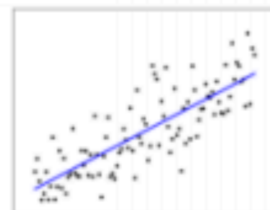
Which method shall I use?



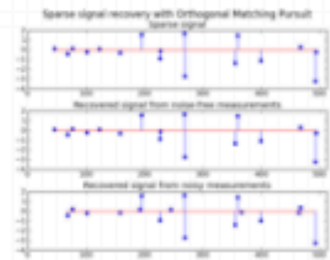
Lasso model selection: Cross-Validation / AIC / BIC



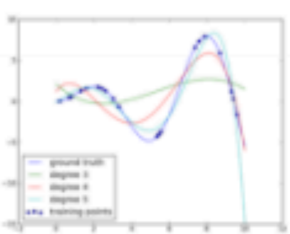
Path with L1- Logistic Regression



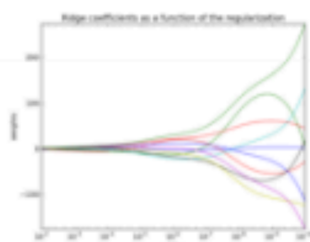
Ordinary Least Squares



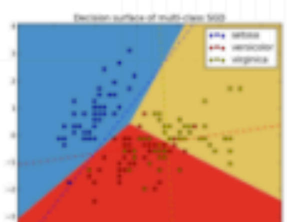
Orthogonal Matching Pursuit



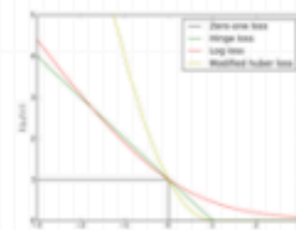
Polynomial interpolation



Plot Ridge coefficients as a function of the regularization



Plot multi-class SGD on the iris dataset

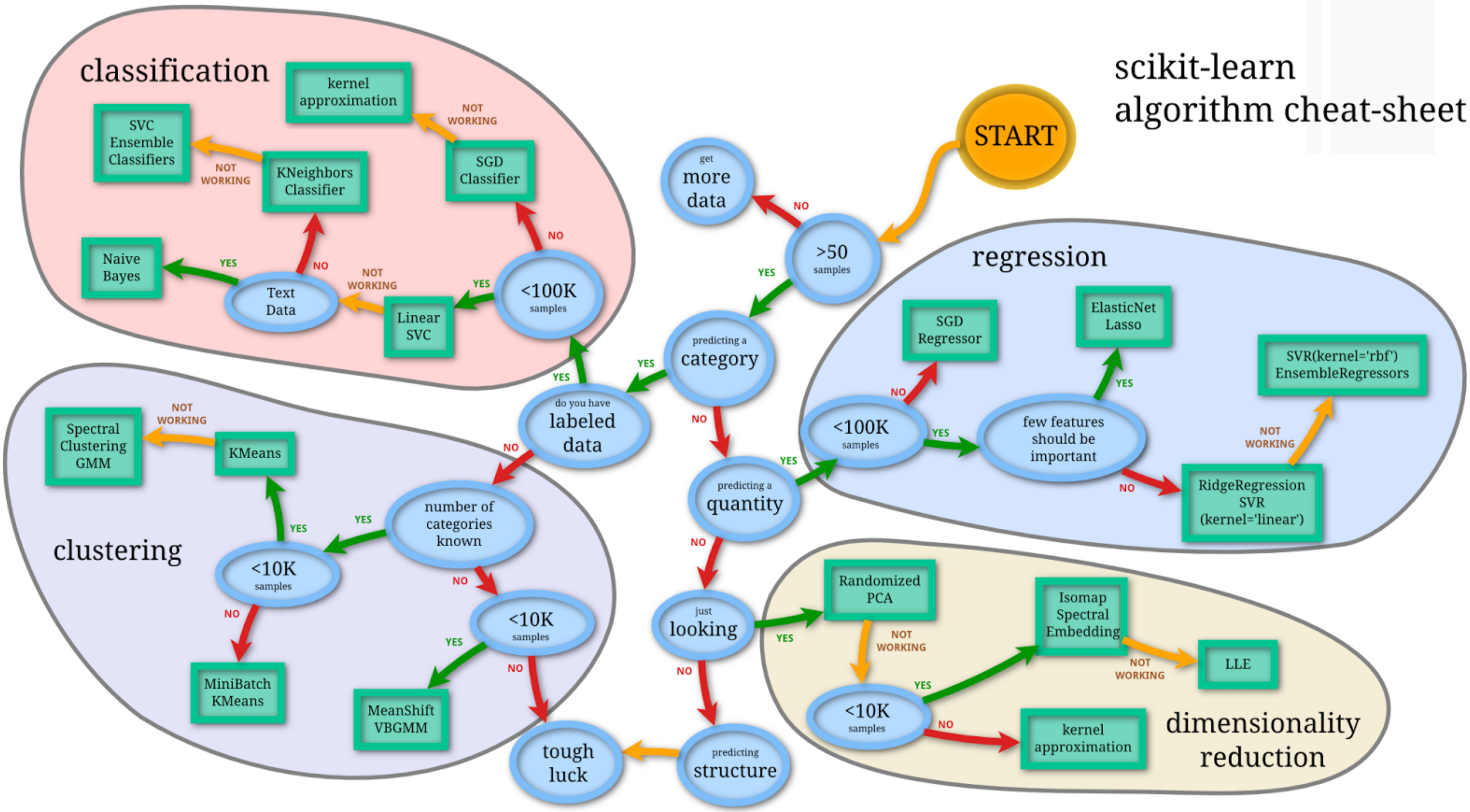


SGD: Convex Loss Functions

Which method should I use?

- Standard Answer: Not really that important
- Cynical Answer: Whichever one performs the best
- Less Cynical Answer: The model that makes the most reasonable assumptions about your problem domain

scikit-learn algorithm cheat-sheet



Things to do:

- Setup your laptop/desktop with tools of choice
  - Install Machine Learning Library
  - Try out demo examples given with that library
- Try out different algorithms
  - E.g. Support Vector Machines
  - Principal Component Analysis
  - Performance Evaluation Metrics (e.g. Area Under the Curve, Average Precision, Average Classification Error etc.)
- Familiarize yourself with reading documentation and understanding parameters

Introduction to Weka

Machine Learning with Weka

- Comprehensive set of tools:
  - Pre-processing and data analysis
  - Learning algorithms (for classification, clustering, etc.)
  - Evaluation metrics
- Three modes of operation:
  - GUI
  - command-line (not discussed today)
  - Java API (not discussed today)

Sample database: the sensus data ("adult")

- Binary classification:
  - Task: predict whether a person earns > $50K a year
  - Attributes: age, education level, race, gender, etc.
  - Attribute types: nominal and numeric
  - Training/test instances: 32,000/16,300

- Original UCI data available at:

  ftp.ics.uci.edu/pub/machine-learning-databases/adult

- Data already converted to ARFF:

  http://www1.cs.columbia.edu/~galley/weka/datasets/

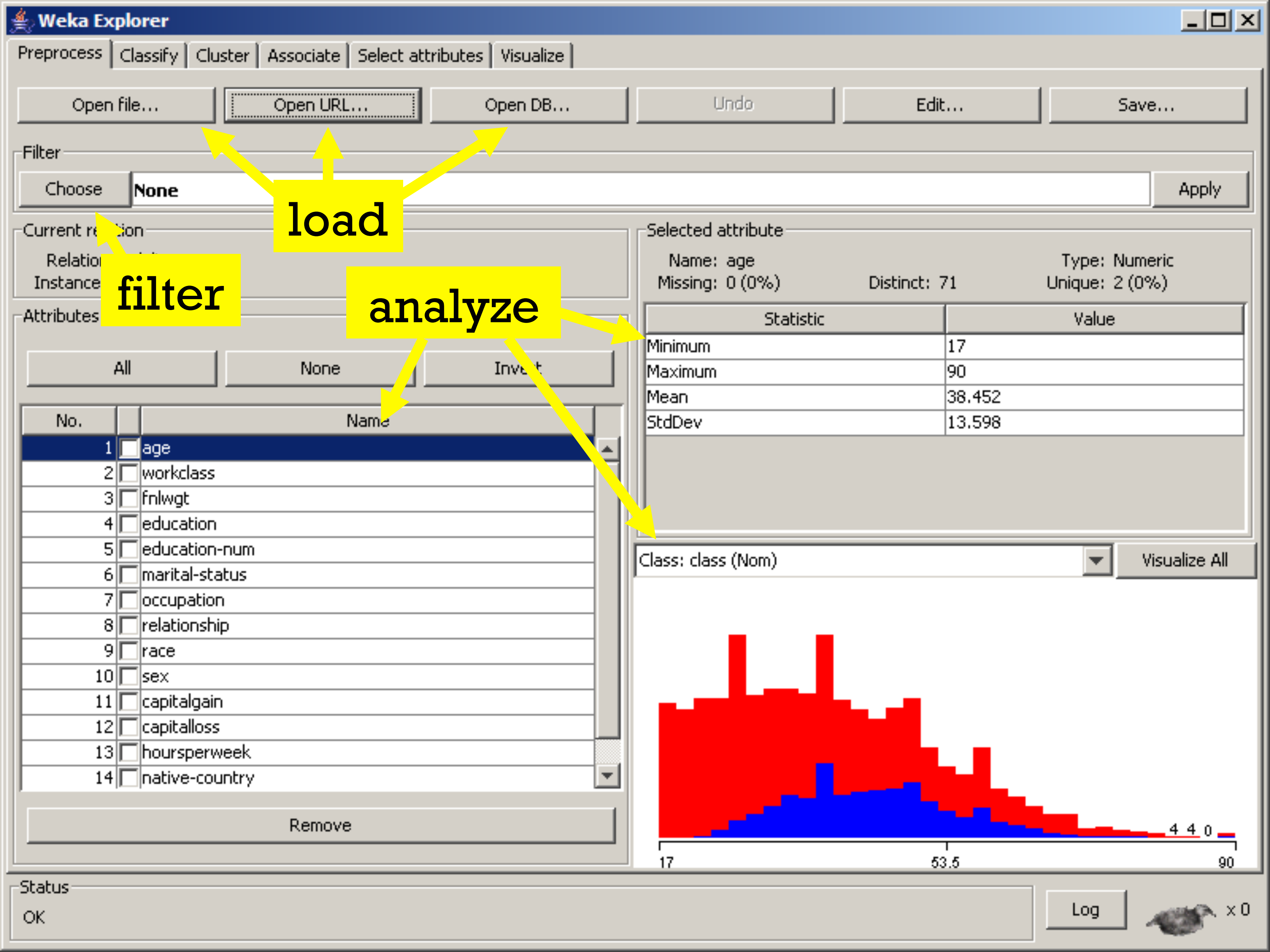
Weka Explorer

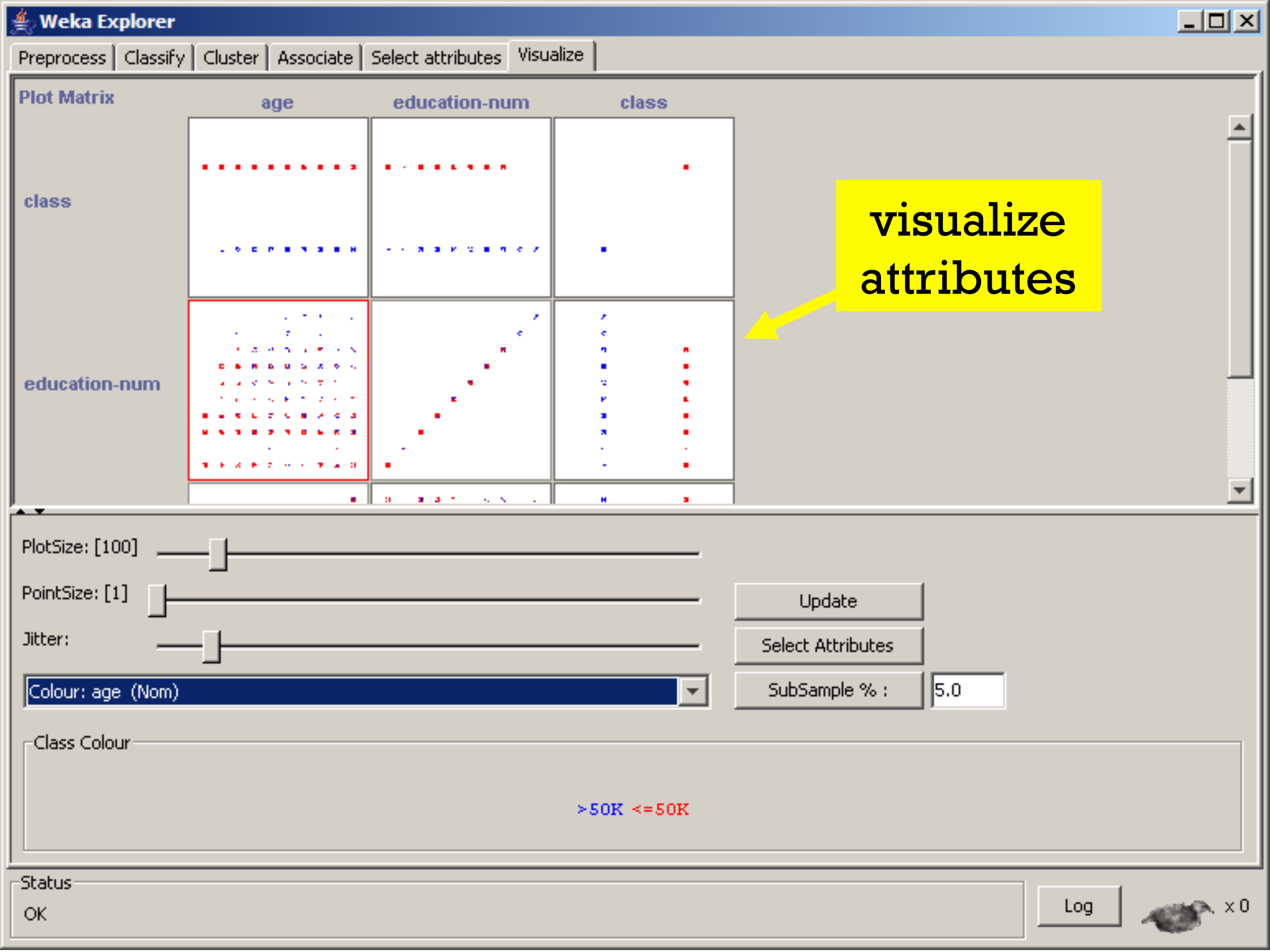
What we will use today in Weka:

I. Pre-process:
   - Load, analyze, and filter data

II. Visualize:
   - Compare pairs of attributes
   - Plot matrices

III. Classify:
   - All algorithms seem in class (Naive Bayes, etc.)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Edit...
Save...
Filter
Choose
None
Apply
load
filter
analyze
Current relation
Attributes
All
None
Invert
No.
Name
1 age
2 workclass
3 fnlwgt
4 education
5 education-num
6 marital-status
7 occupation
8 relationship
9 race
10 sex
11 capitalgain
12 capitalloss
13 hoursperweek
14 native-country
Remove
Selected attribute
Name: age
Type: Numeric
Missing: 0 (0%)
Distinct: 71
Unique: 2 (0%)
Statistic
Value
Minimum 17
Maximum 90
Mean 38.452
StdDev 13.598
Class: class (Nom)
Visualize All
4 4 0
17
53.5
90
Status
OK
Log
x 0



Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Plot Matrix
age
education-num
class
class
education-num
visualize attributes
PlotSize: [100]
PointSize: [1]
Jitter:
Update
Select Attributes
Colour: age (Nom)
SubSample % :
5.0
Class Colour
>50K <=50K
Status
OK
Log
x 0

Linear classifiers

- Prediction is a **linear** function of the input
  - in the case of binary predictions, a linear classifier splits a high-dimensional input space with a **hyperplane** (i.e., a plane in 3D, or a straight line in 2D).

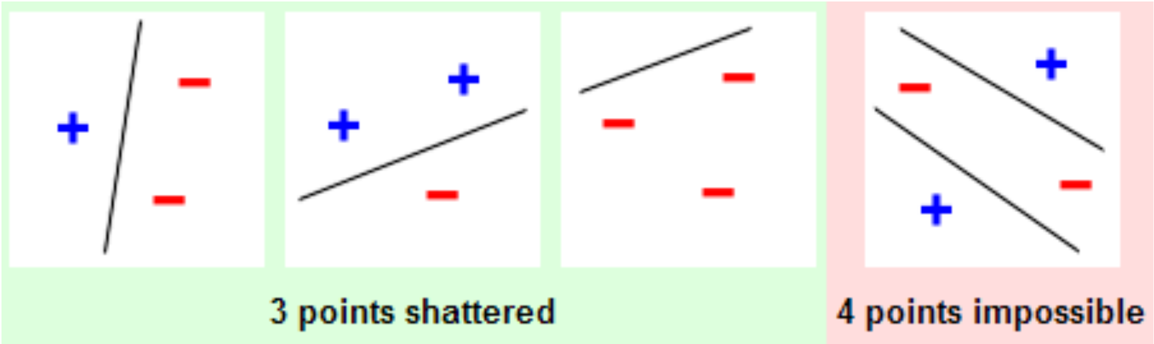
  - Many popular effective classifiers are linear: perceptron, linear SVM, logistic regression (a.k.a. maximum entropy, exponential model).

Comparing classifiers

- Results on "adult" data
  - Majority-class baseline: 76.51%
    (always predict <=50K)
    `weka.classifier.rules.ZeroR`
  - Naive Bayes: 79.91%
    `weka.classifier.bayes.NaiveBayes`
  - Linear classifier: 78.88%
    `weka.classifier.function.Logistic`
  - Decision trees: 79.97%
    `weka.classifier.trees.J48`

Why this difference?

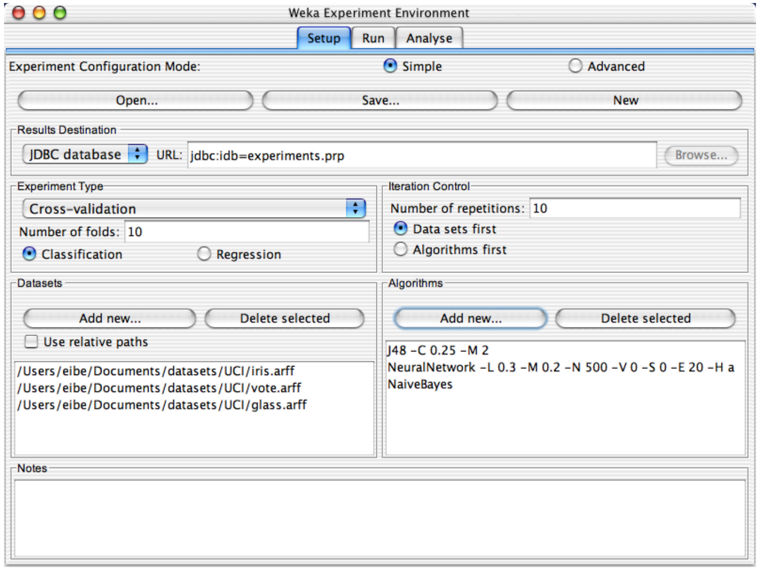
- A linear classifier in a 2D space:
  - it can classify correctly ("shatter") any set of 3 points;
  - not true for 4 points;
  - we say then that 2D-linear classifiers have *capacity* 3.

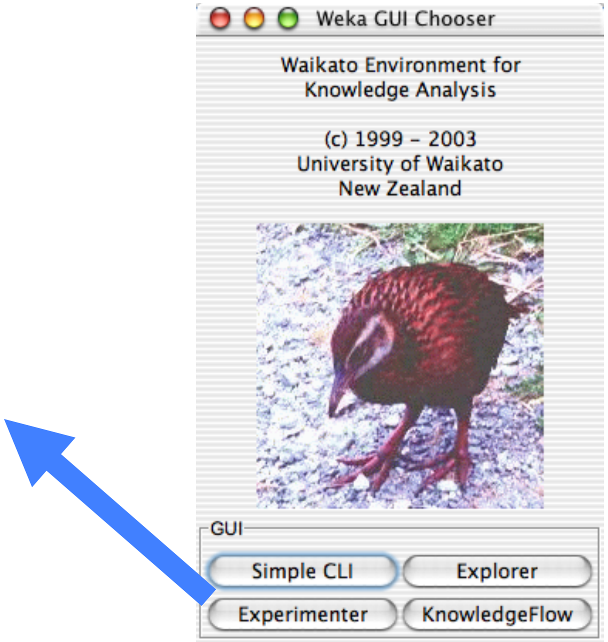


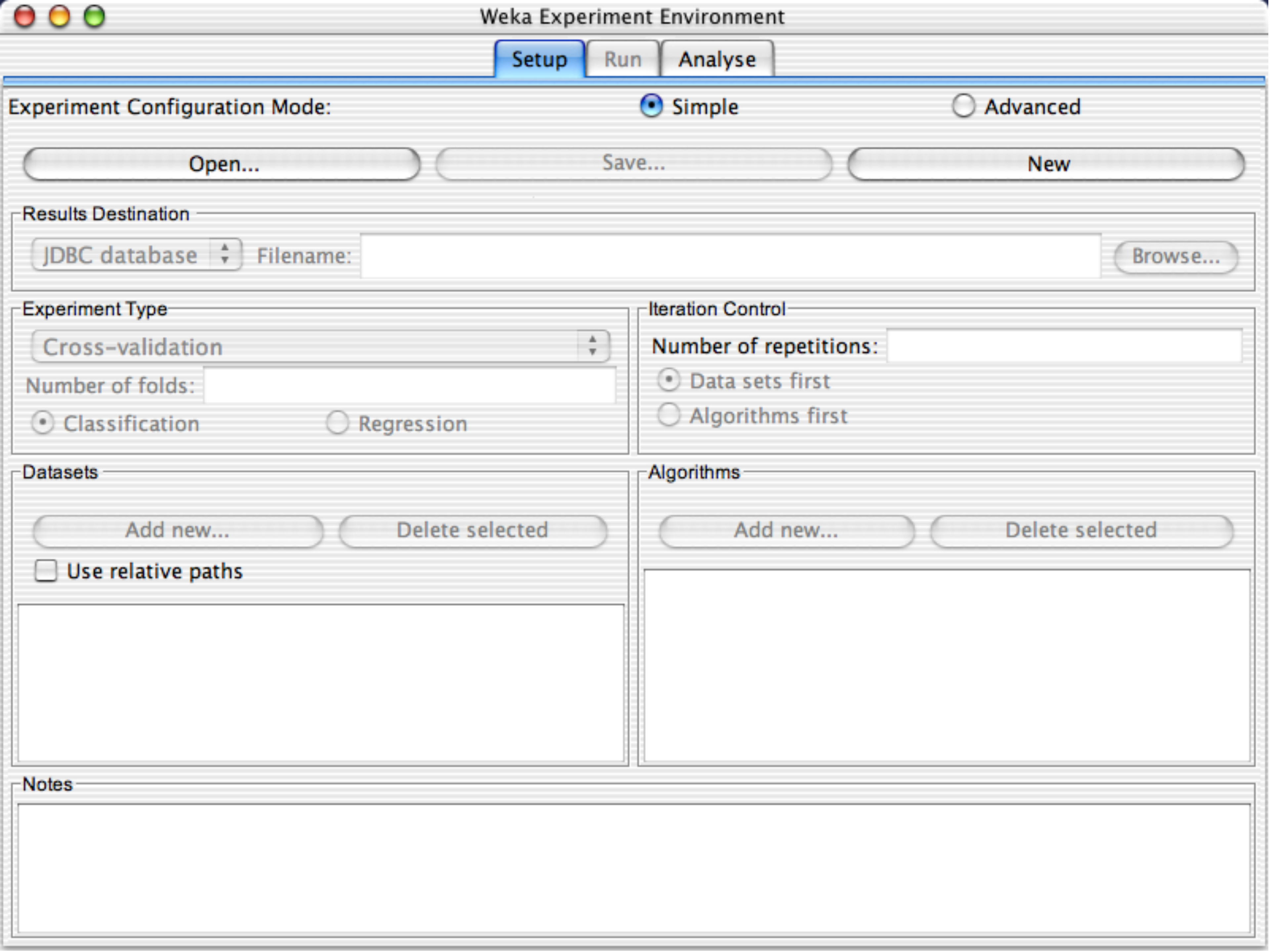
- A decision tree in a 2D space:
  - can shatter as many points as leaves in the tree;
  - potentially unbounded capacity! (e.g., if no tree pruning)

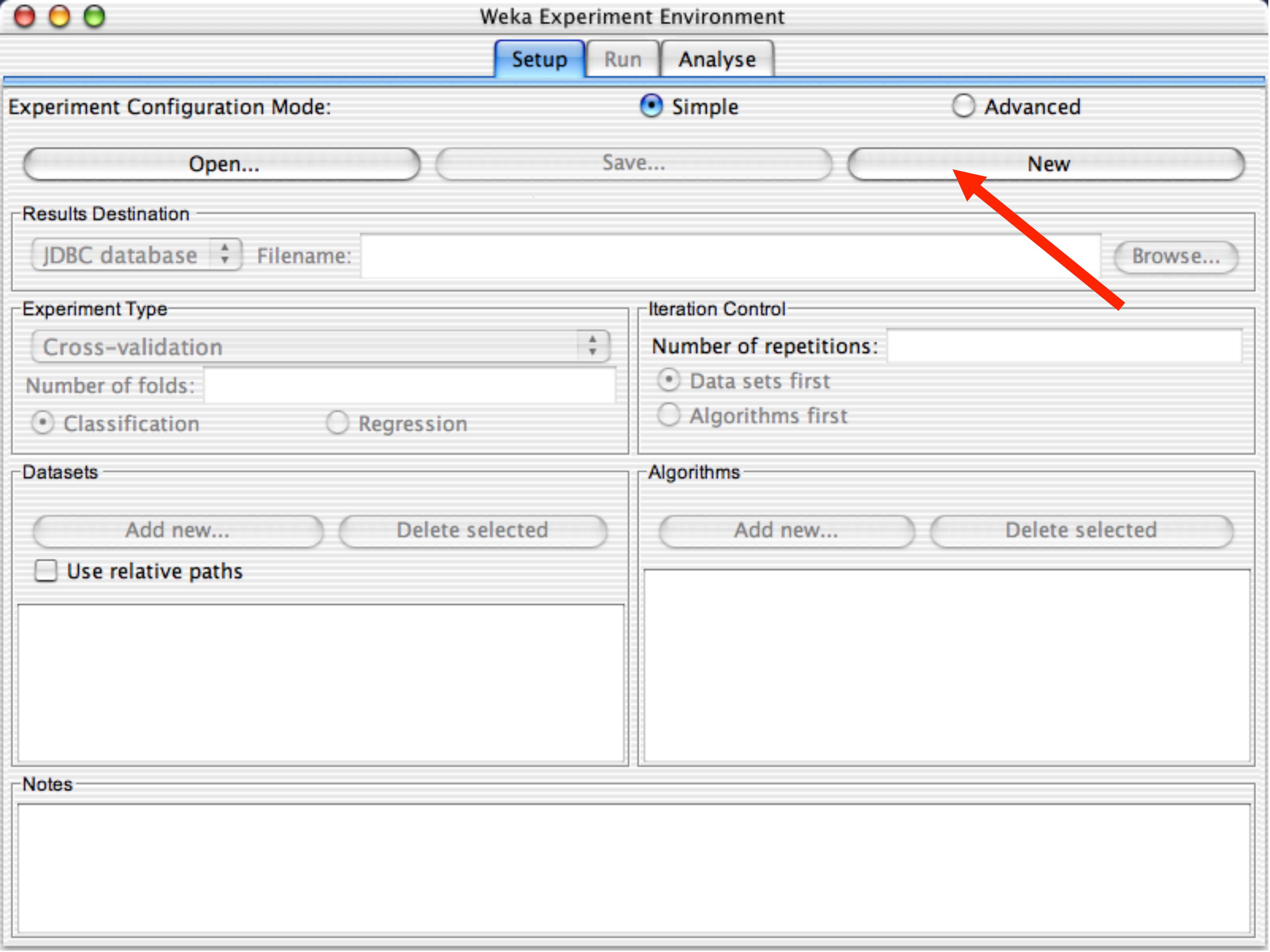
Weka Experimenter

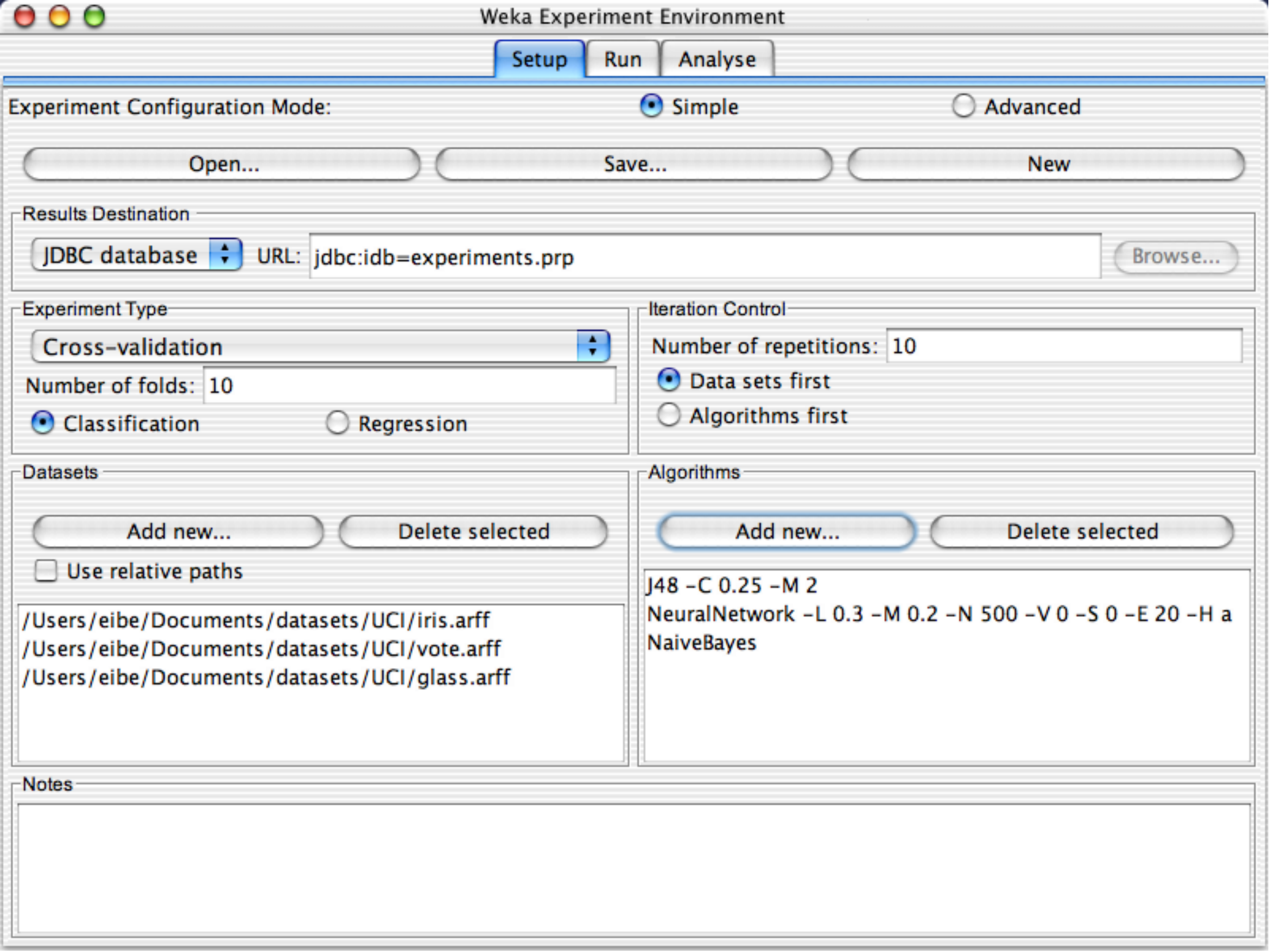
- If you need to perform many experiments:
  - Experimenter makes it easy to compare the performance of different learning schemes
  - Results can be written into file or database
  - Evaluation options: cross-validation, learning curve, etc.
  - Can also iterate over different parameter settings
  - Significance-testing built in.

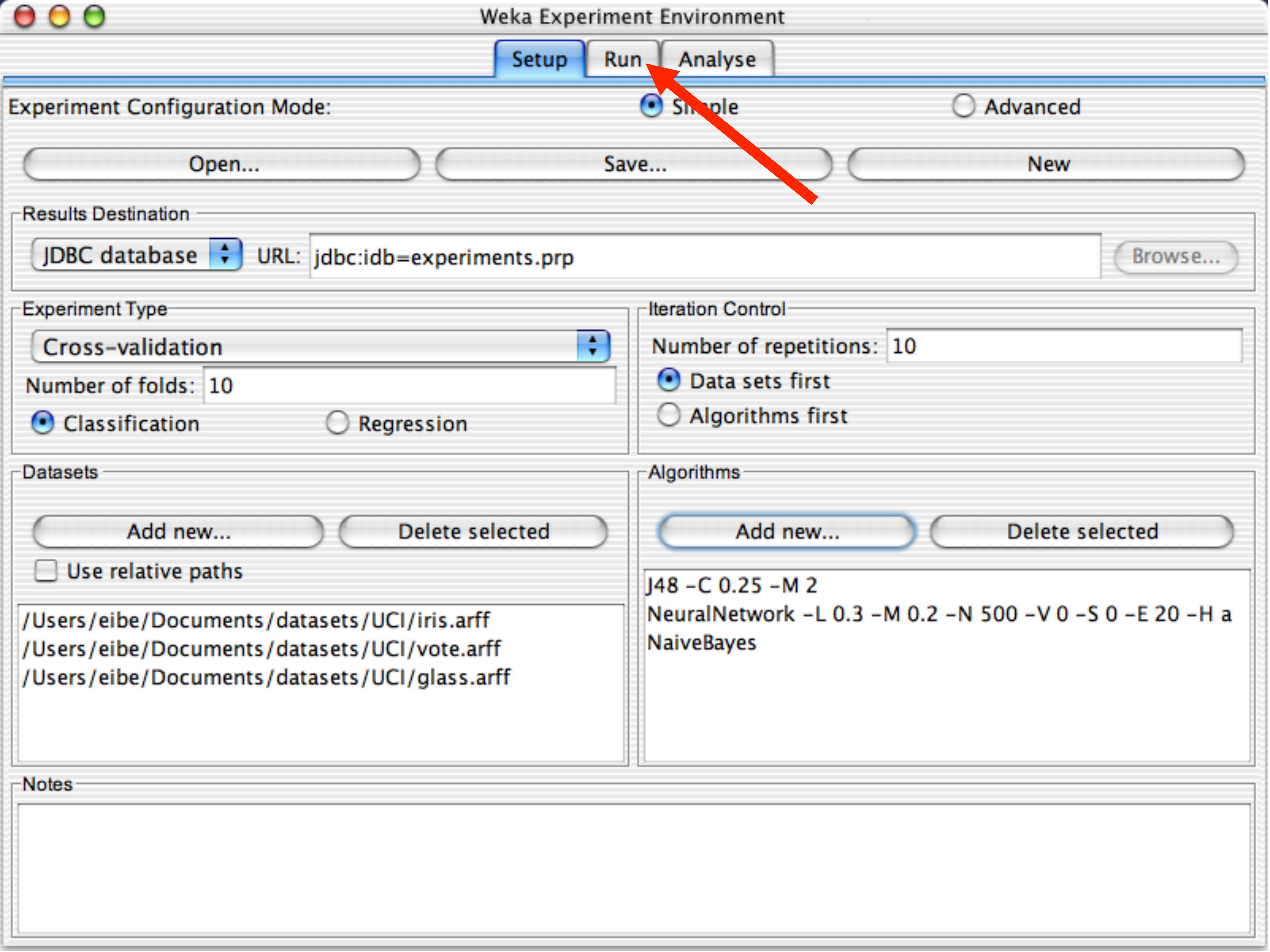




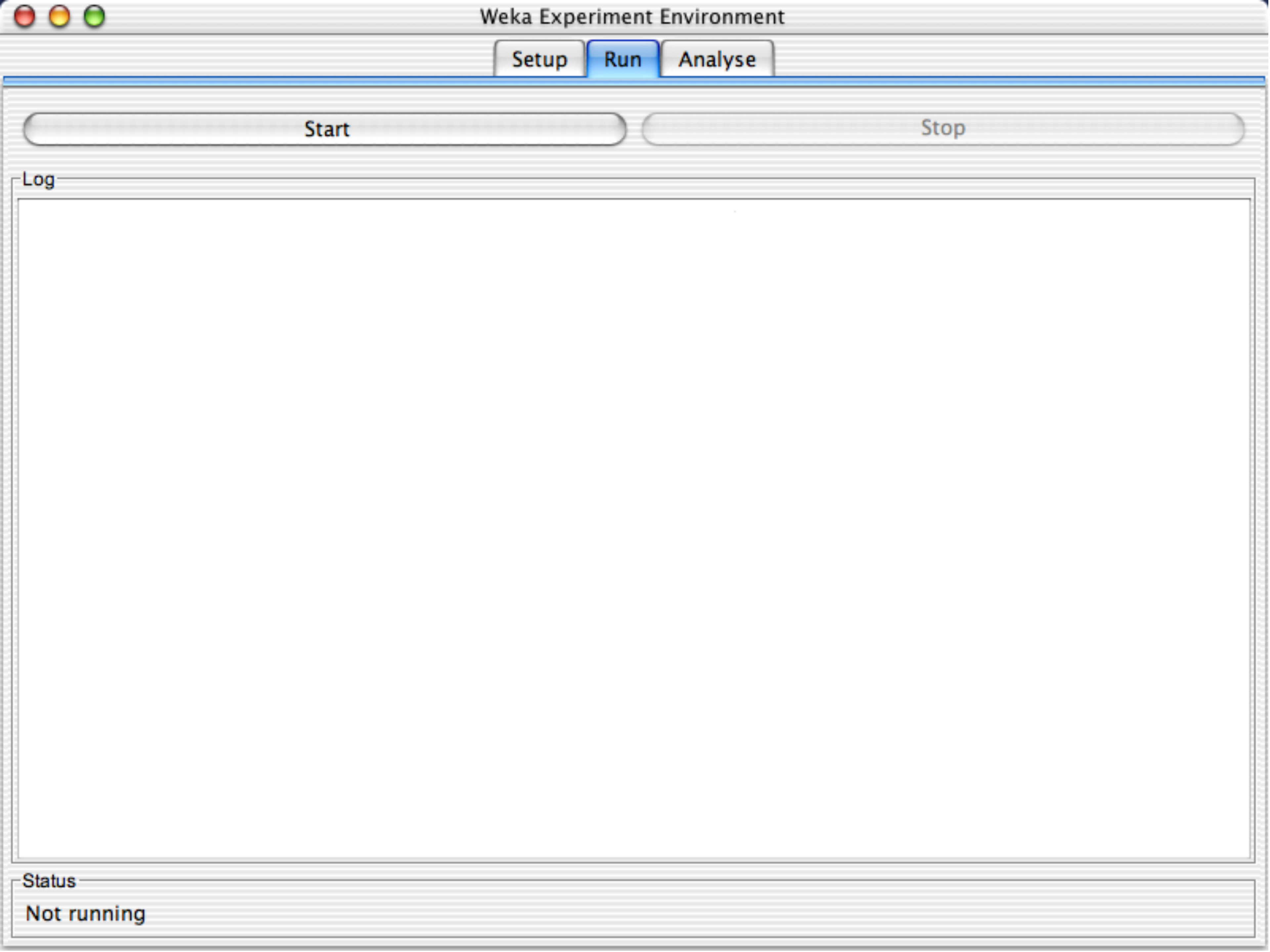

Weka Experiment Environment
Setup
Run
Analyse
Experiment Configuration Mode:
Simple
Advanced
Open...
Save...
New
Results Destination
JDBC database
Filename:
Browse...
Experiment Type
Cross-validation
Number of folds:
Classification
Regression
Iteration Control
Number of repetitions:
Data sets first
Algorithms first
Datasets
Add new...
Delete selected
Use relative paths
Algorithms
Add new...
Delete selected
Notes

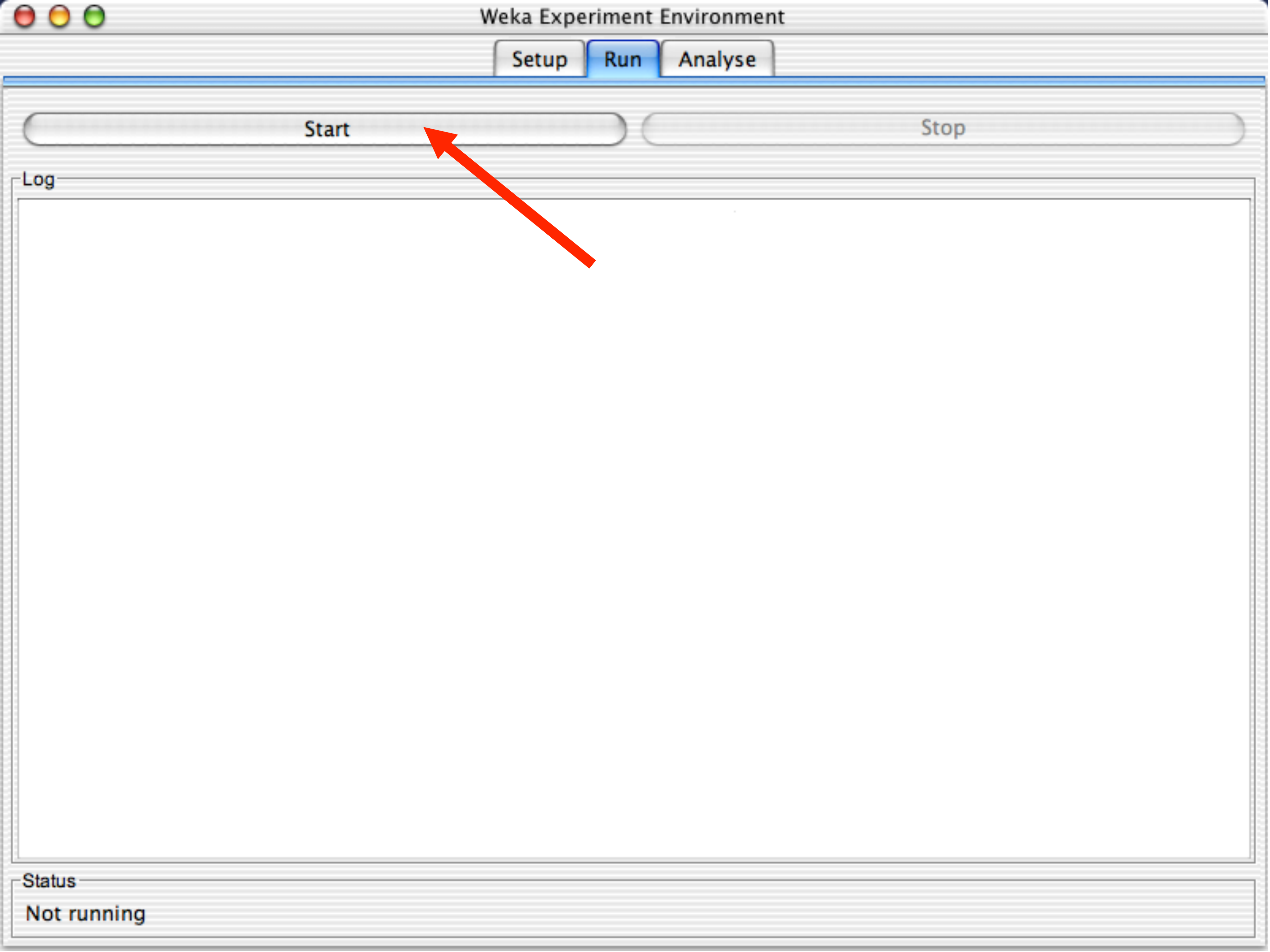

Weka Experiment Environment
Setup
Run
Analyse
Experiment Configuration Mode:
Simple
Advanced
Open...
Save...
New
Results Destination
JDBC database
Filename:
Browse...
Experiment Type
Cross-validation
Number of folds:
Classification
Regression
Iteration Control
Number of repetitions:
Data sets first
Algorithms first
Datasets
Add new...
Delete selected
Use relative paths
Algorithms
Add new...
Delete selected
Notes

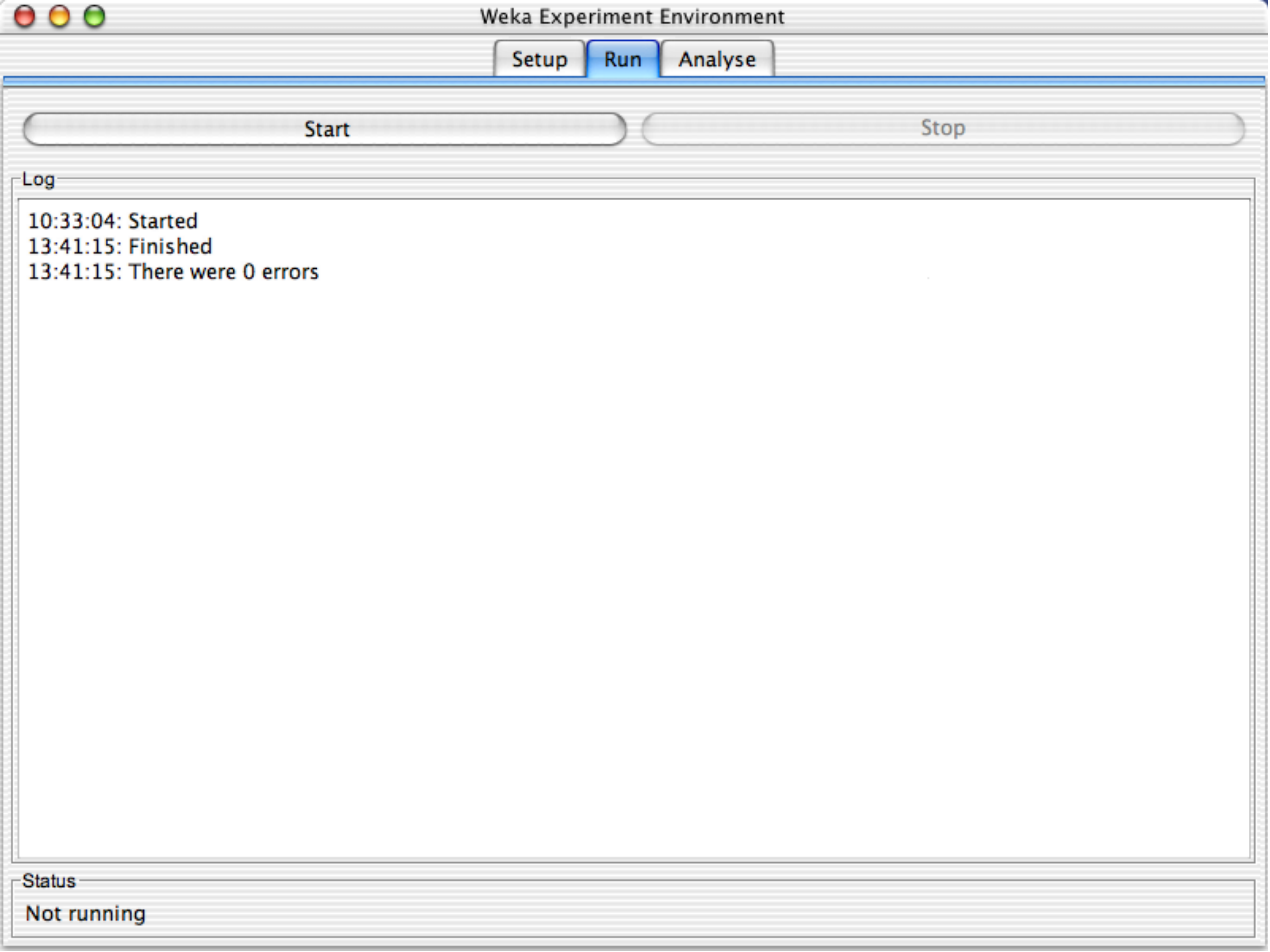

Weka Experiment Environment
Setup
Run
Analyse
Experiment Configuration Mode:
Simple
Advanced
Open...
Save...
New
Results Destination
JDBC database
URL:
jdbc:idb=experiments.prp
Browse...
Experiment Type
Cross-validation
Number of folds:
10
Classification
Regression
Iteration Control
Number of repetitions:
10
Data sets first
Algorithms first
Datasets
Add new...
Delete selected
Use relative paths
/Users/eibe/Documents/datasets/UCI/iris.arff
/Users/eibe/Documents/datasets/UCI/vote.arff
/Users/eibe/Documents/datasets/UCI/glass.arff
Algorithms
Add new...
Delete selected
J48 -C 0.25 -M 2
NeuralNetwork -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
NaiveBayes
Notes

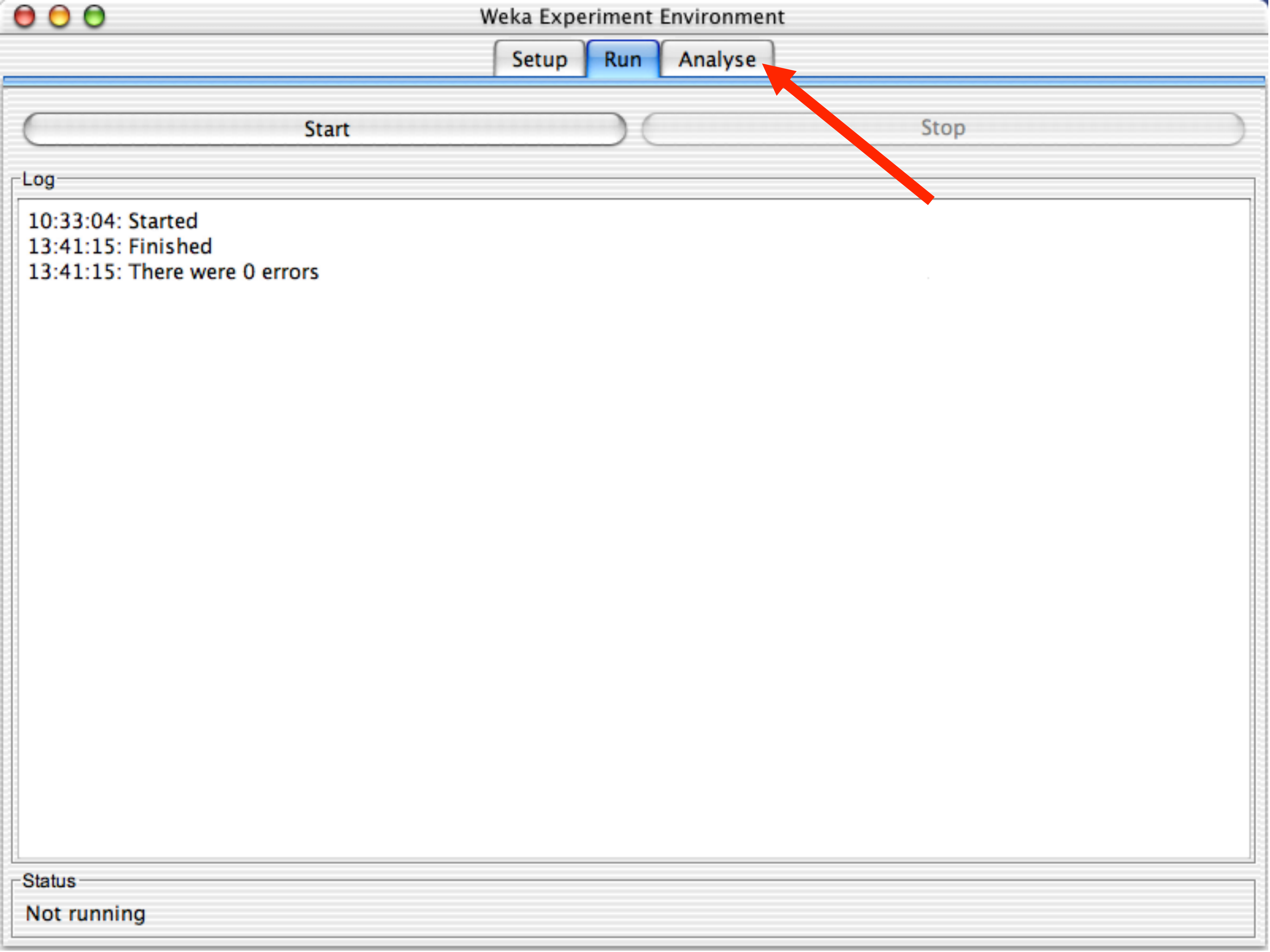


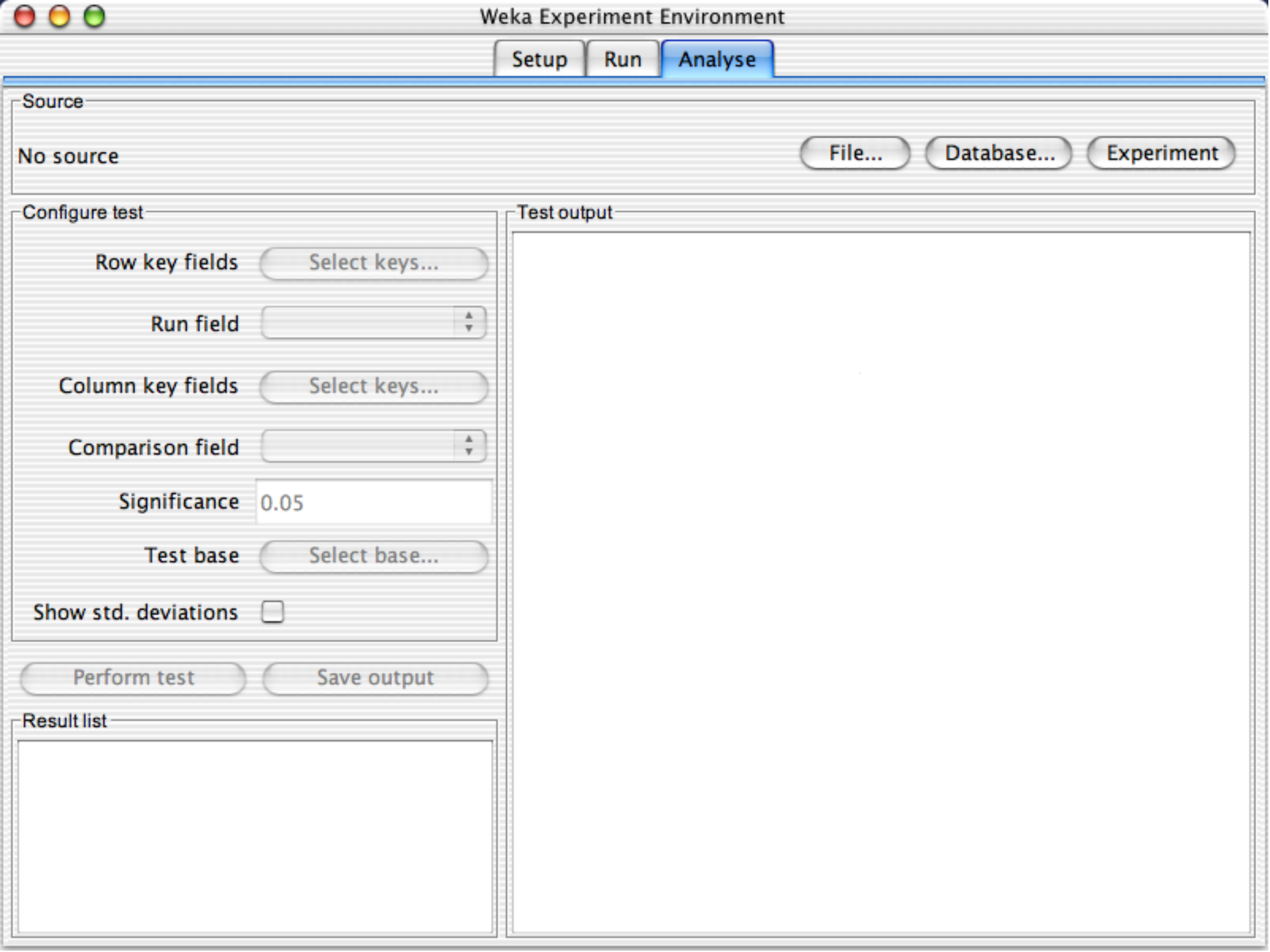
Weka Experiment Environment
Setup
Run
Analyse
Experiment Configuration Mode:
Simple
Advanced
Open...
Save...
New
Results Destination
JDBC database
URL:
jdbc:idb=experiments.prp
Browse...
Experiment Type
Cross-validation
Number of folds: 10
Classification
Regression
Iteration Control
Number of repetitions: 10
Data sets first
Algorithms first
Datasets
Add new...
Delete selected
Use relative paths
/Users/eibe/Documents/datasets/UCI/iris.arff
/Users/eibe/Documents/datasets/UCI/vote.arff
/Users/eibe/Documents/datasets/UCI/glass.arff
Algorithms
Add new...
Delete selected
J48 -C 0.25 -M 2
NeuralNetwork -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
NaiveBayes
Notes

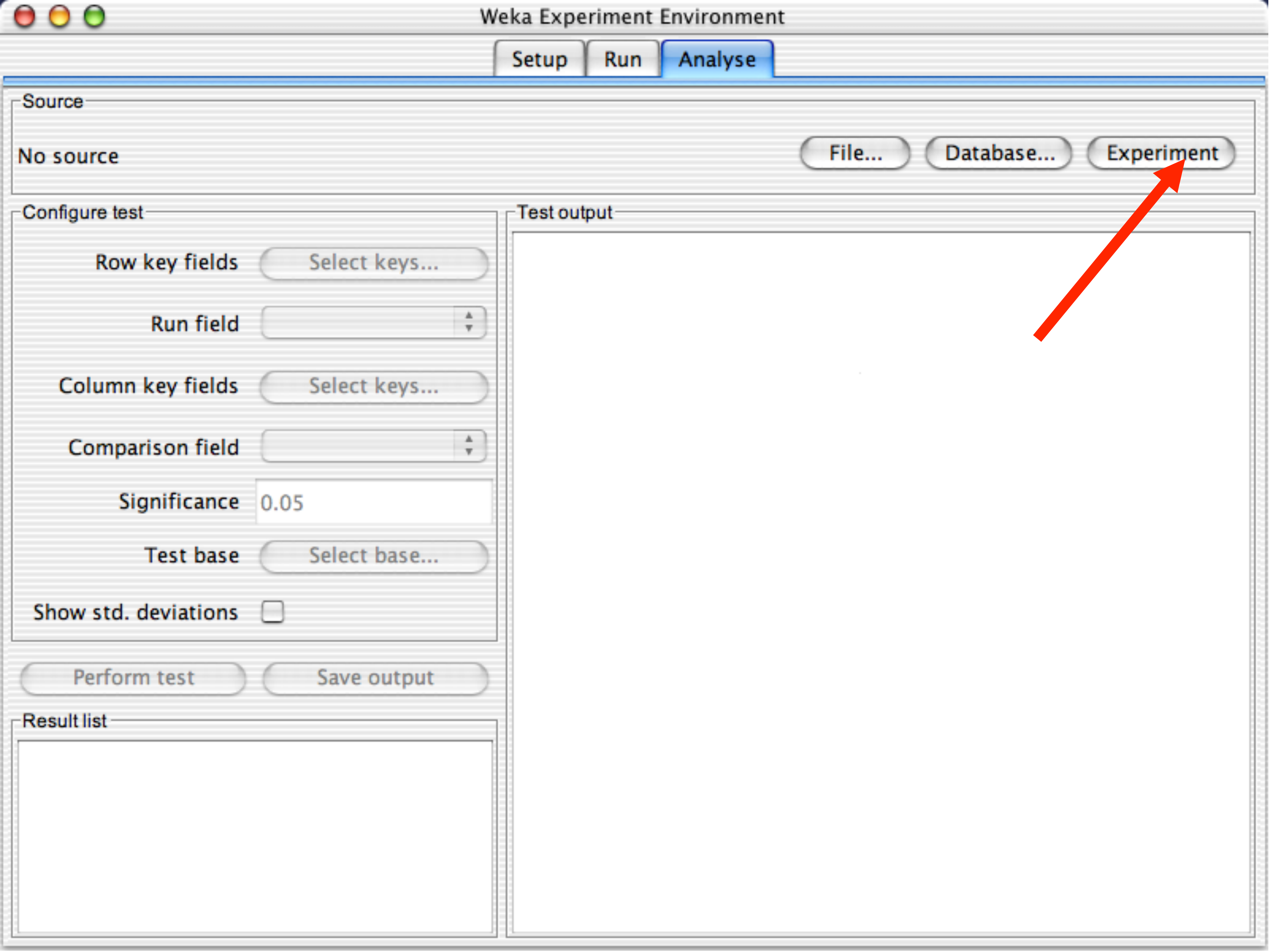

Weka Experiment Environment
Setup
Run
Analyse
Start
Stop
Log
Status
Not running


Weka Experiment Environment
Setup
Run
Analyse
Start
Stop
Log
Status
Not running


Weka Experiment Environment
Setup
Run
Analyse
Start
Stop
Log
10:33:04: Started
13:41:15: Finished
13:41:15: There were 0 errors
Status
Not running

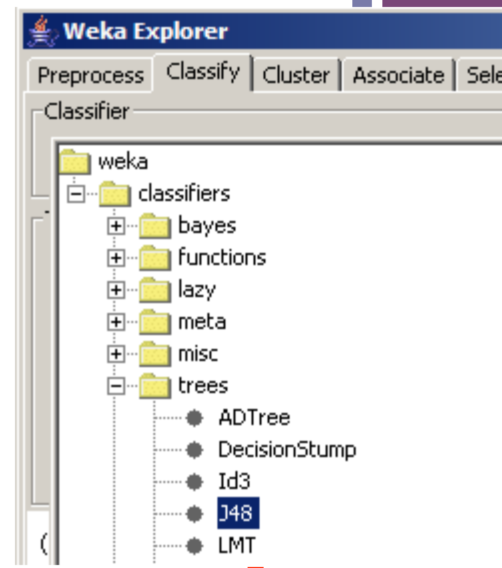

Weka Experiment Environment
Setup
Run
Analyse
Start
Stop
Log
10:33:04: Started
13:41:15: Finished
13:41:15: There were 0 errors
Status
Not running

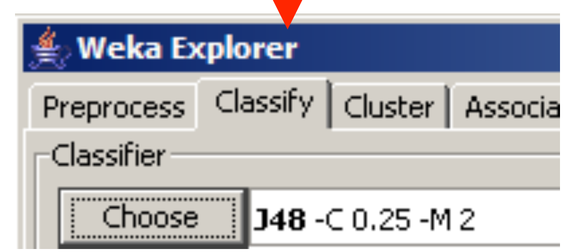

Weka Experiment Environment
Setup
Run
Analyse
Source
No source
File...
Database...
Experiment
Configure test
Row key fields
Select keys...
Run field
Column key fields
Select keys...
Comparison field
Significance
0.05
Test base
Select base...
Show std. deviations
Perform test
Save output
Result list
Test output


Weka Experiment Environment
Setup
Run
Analyse
Source
No source
File...
Database...
Experiment
Configure test
Row key fields
Select keys...
Run field
Column key fields
Select keys...
Comparison field
Significance
0.05
Test base
Select base...
Show std. deviations
Perform test
Save output
Result list
Test output

Beyond the GUI

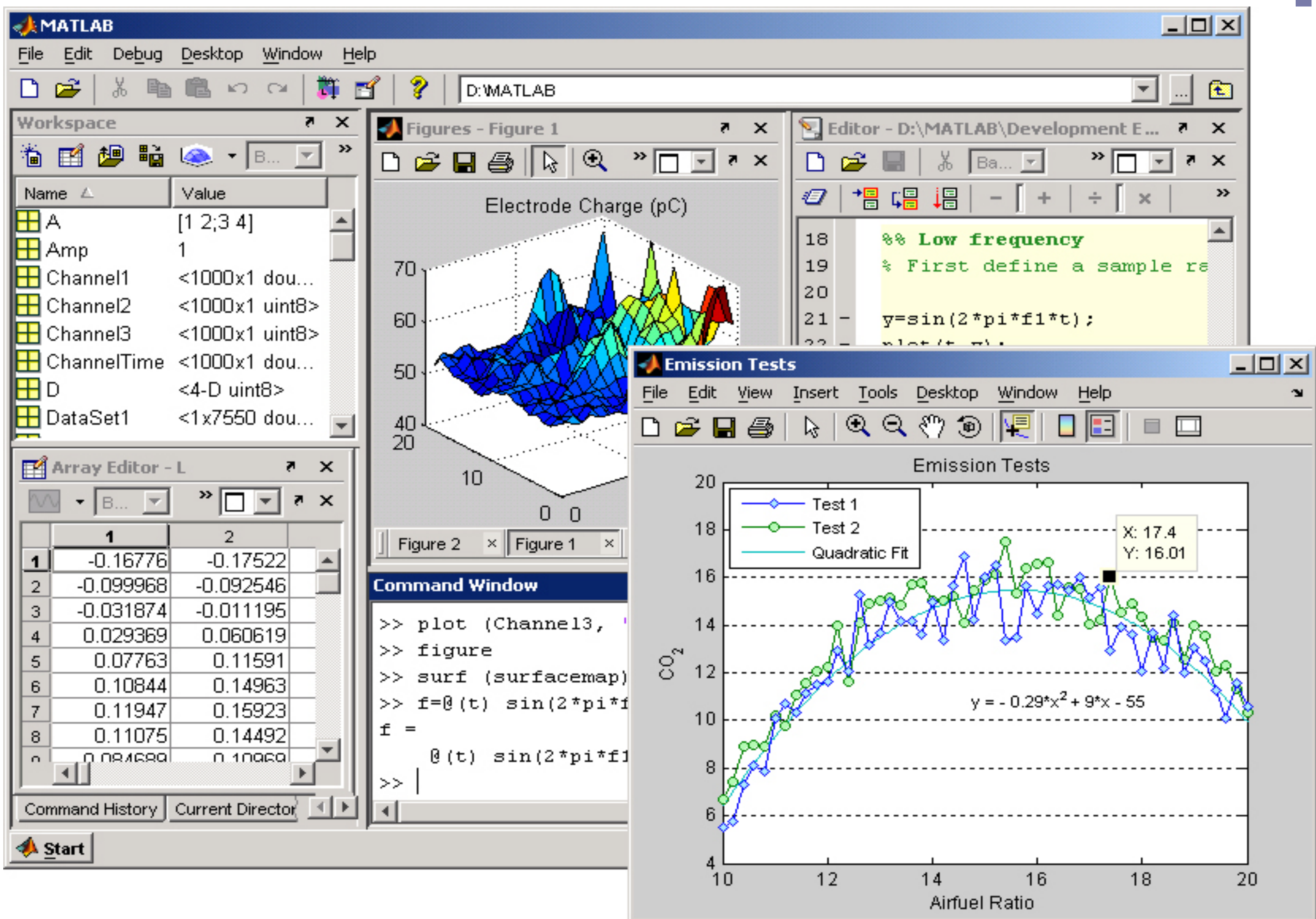
- How to reproduce experiments with the command-line/API
  - GUI, API, and command-line all rely on the same set of Java classes
  - Generally easy to determine what classes and parameters were used in the GUI.
  - Tree displays in Weka reflect its Java class hierarchy.





> java -cp ~galley/weka/weka.jar
weka.classifiers.trees.J48 –C 0.25 –M 2
-t <train_arff> -T <test_arff>

+ Matlab and BigData



BigData with Matlab

Advantages	Disadvantages
Easy to use	Costly
Great for data analysis: Really nice plotting tools	Parallelizing means need multiple licenses
Access to wide array of toolboxes	Rarely used as a backend in production
Parallel Computing toolbox - GPU computing	Memory heavy
Almost all machine learning algorithms are available	Proprietary

Machine Learning in Matlab

Classification

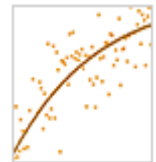
Build models to classify data into different categories.

Algorithms: support vector machine (SVM), boosted and bagged decision trees, k-nearest neighbor, Naïve Bayes, discriminant analysis, neural networks, and more

» Get started with introductory examples

Applications: credit scoring, tumor detection, image recognition

Regression



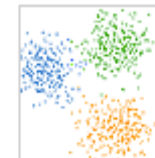
Build models to predict continuous data.

Algorithms: linear model, nonlinear model, regularization, stepwise regression, boosted and bagged decision trees, neural networks, and more

» Get started with introductory examples

Applications: electricity load forecasting, algorithmic trading, drug discovery

Clustering



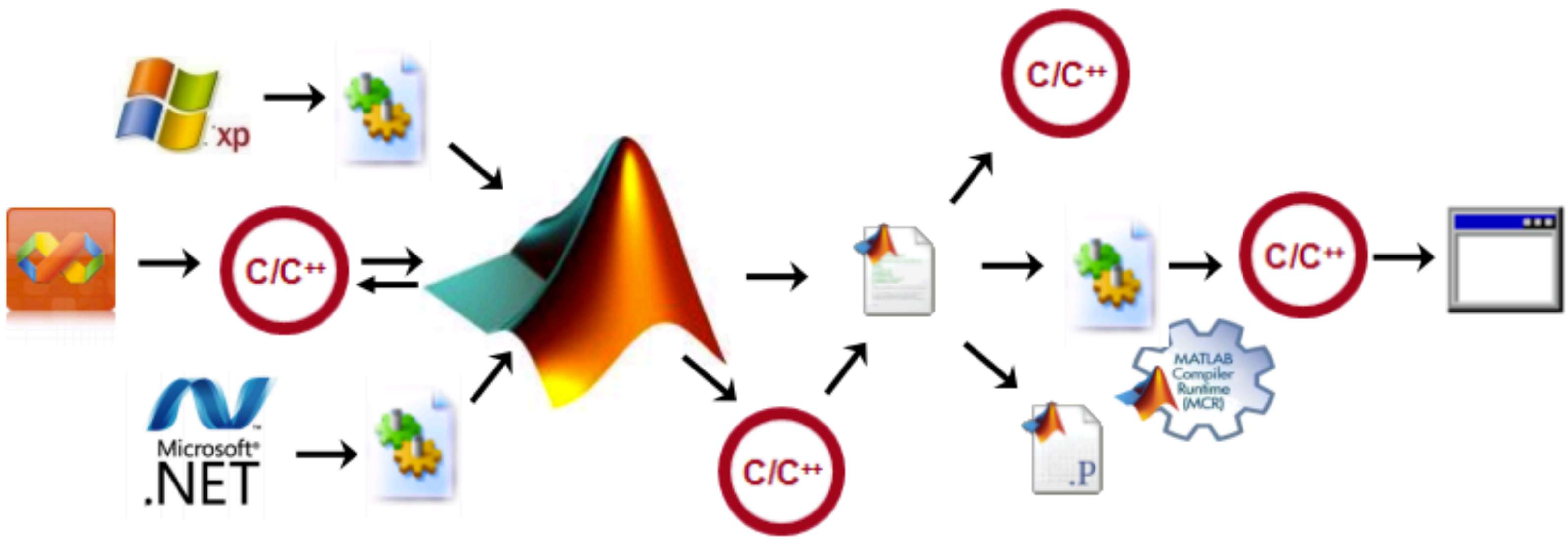
Find natural groupings and patterns in data.

Algorithms: k-means, hierarchical clustering, Gaussian mixture models, hidden Markov models, self-organizing maps, and more

» Get started with introductory examples

Applications: pattern mining, medical imaging, object recognition

Duct-taping external libraries



Bit tedious and time consuming. You need lot of experience doing this to get good performance.

Matlab Tools for BigData

MATLAB

Single Machine
Single Processor

Multithreaded

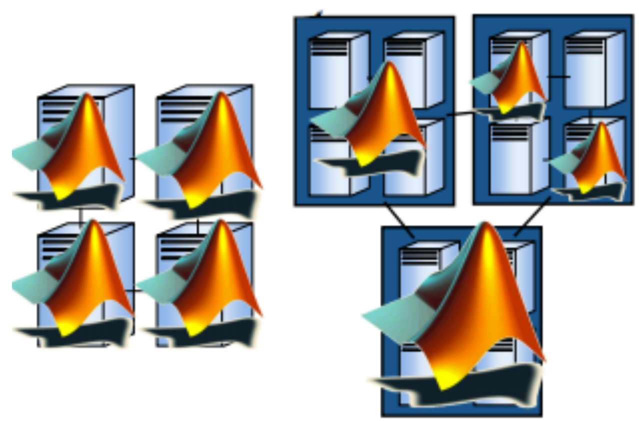
Single Machine
Use cores / processors efficiently

GPU Computing

Harness power of GPUs

Parallel Computing

Use all processors on a machines: e.g. efficiently using 8-core machine



Distributed Computing

Distribute across the cloud: e.g. using Amazon EC2 cloud

Statistics Toolbox in MATLAB

- Iris Data Analysis using matlab demo

Dlib C++ Library

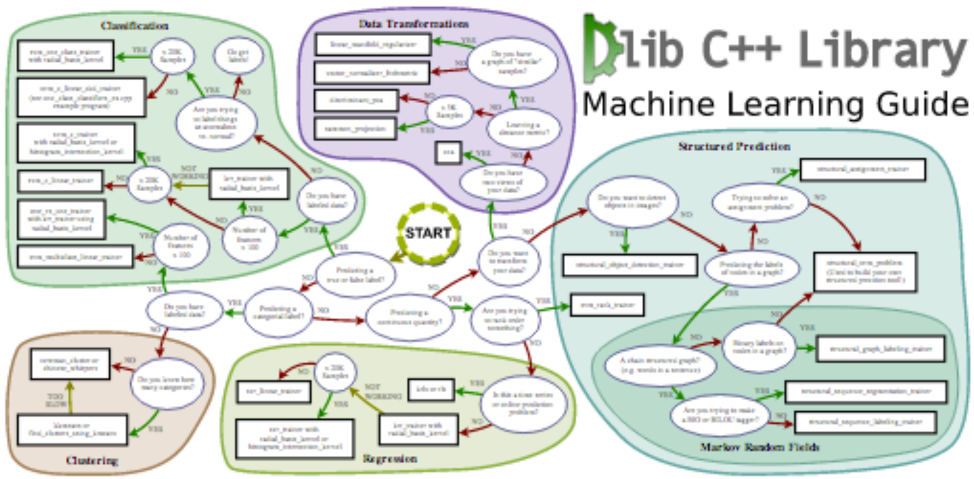
The Library

- Algorithms
- API Wrappers
- Bayesian Nets
- Compression
- Containers
- Graph Tools
- Image Processing
- Linear Algebra
- Machine Learning
- Metaprogramming
- Miscellaneous
- Networking
- Optimization
- Parsing

Help/Info

- ⊞Examples: C++
- ⊞Examples: Python
- FAQ
- Home
- How to compile
- How to contribute
- Index
- Introduction
- License
- Python API
- Suggested Books

Machine Learning



A major design goal of this portion of the library is to provide a highly modular and simple architecture for dealing with kernel algorithms. Towards this end, dlib takes a generic programming approach using C++ templates. In particular, each algorithm is parameterized to allow a user to supply either one of the predefined dlib kernels (e.g. RBF operating on column vectors), or a new user defined kernel. Moreover, the implementations of the algorithms are totally separated from the data on which they operate. This makes the dlib implementation generic enough to operate on any kind of data, be it column

Primary Algorithms

Binary Classification

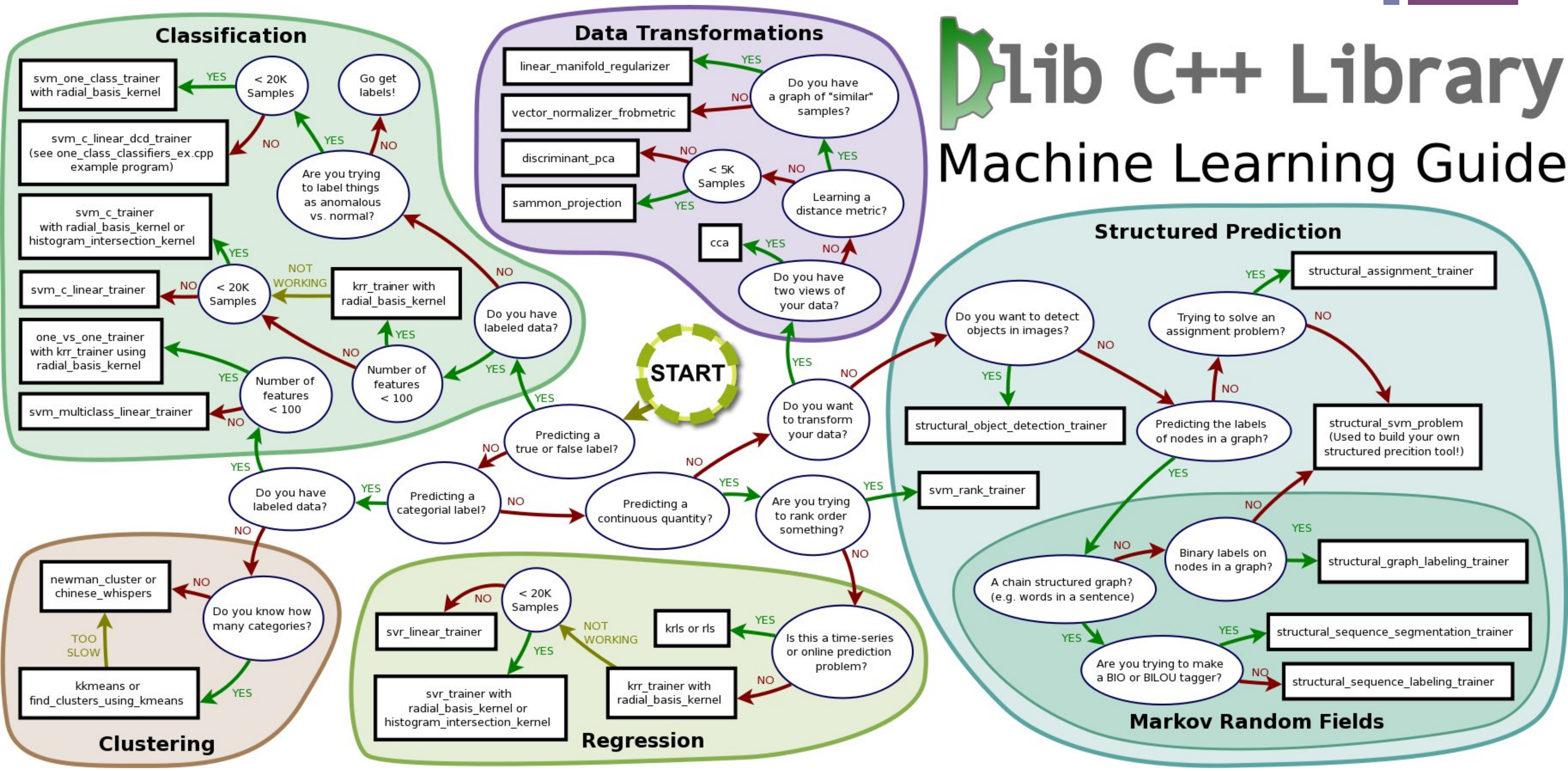
- rvm_trainer
- svm_c_ekm_trainer
- svm_c_linear_dcd_trainer
- svm_c_linear_trainer
- svm_c_trainer
- svm_nu_trainer
- svm_pegasos
- train_probabilistic_decision_function

Multiclass Classification

- one_vs_all_trainer
- one_vs_one_trainer
- svm_multiclass_linear_trainer

Regression

- krls
- krr_trainer
- mlp
- rbf_network_trainer
- rls
- rr_trainer
- rvm_regression_trainer
- svr_linear_trainer
- svr_trainer


Dlib C++ Library
Machine Learning Guide
START
Classification
Data Transformations
Structured Prediction
Markov Random Fields
Clustering
Regression
svm_one_class_trainer with radial_basis_kernel
< 20K Samples
Go get labels!
svm_c_linear_dcd_trainer (see one_class_classifiers_ex.cpp example program)
Are you trying to label things as anomalous vs. normal?
svm_c_trainer with radial_basis_kernel or histogram_intersection_kernel
svm_c_linear_trainer
< 20K Samples
NOT WORKING
krr_trainer with radial_basis_kernel
Do you have labeled data?
one_vs_one_trainer with krr_trainer using radial_basis_kernel
Number of features < 100
Number of features < 100
svm_multiclass_linear_trainer
Predicting a true or false label?
Do you have labeled data?
Predicting a categorial label?
Predicting a continuous quantity?
Are you trying to rank order something?
svm_rank_trainer
Do you want to transform your data?
linear_manifold_regularizer
Do you have a graph of "similar" samples?
vector_normalizer_frobmetric
discriminant_pca
< 5K Samples
Learning a distance metric?
sammon_projection
cca
Do you have two views of your data?
Do you want to detect objects in images?
Trying to solve an assignment problem?
structural_assignment_trainer
structural_object_detection_trainer
Predicting the labels of nodes in a graph?
structural_svm_problem (Used to build your own structured precition tool!)
A chain structured graph? (e.g. words in a sentence)
Binary labels on nodes in a graph?
structural_graph_labeling_trainer
structural_sequence_segmentation_trainer
Are you trying to make a BIO or BILOU tagger?
structural_sequence_labeling_trainer
newman_cluster or chinese_whispers
Do you know how many categories?
TOO SLOW
kkmeans or find_clusters_using_kmeans
svr_linear_trainer
< 20K Samples
NOT WORKING
krls or rls
Is this a time-series or online prediction problem?
svr_trainer with radial_basis_kernel or histogram_intersection_kernel
krr_trainer with radial_basis_kernel
YES
NO

Recap

Don't need to know everything..!

- Pick language of your choice
- Use machine learning tools related to that
- Install ML library
- Try out examples given with the library
- That's it..!

Publicly Available Large Datasets

Amazon Public Datasets

amazon web services

Sign Up | My Account / Console | English

AWS Products & Solutions | Public Data Sets | Developers | Support

Public Data Sets

Public Data Sets on AWS provides a centralized repository of public data sets that can be seamlessly integrated into AWS cloud-based applications. AWS is hosting the public data sets at no charge for the community, and like all AWS services, users pay only for the compute and storage they use for their own applications. Learn more about Public Data Sets on AWS and visit the Public Data Sets forum.

Featured Public Data Sets

Common Crawl Corpus
A corpus of web crawl data composed of over 5 billion web pages. This data set is freely available on Amazon S3 and is released under the Common Crawl Terms of Use.

1000 Genomes Project
The 1000 Genomes Project, initiated in 2008, is an international public-private consortium that aims to build the most detailed map of human genetic variation available.

Google Books Ngrams
A data set containing Google Books n-gram corpuses. This data set is freely available on Amazon S3 in a Hadoop friendly file format and is licensed under a Creative Commons Attribution 3.0 Unported License. The original dataset is available from http://books.google.com/ngrams/.

Browse By Category

- Astronomy
- Biology
- Chemistry
- Climate
- Economics
- Encyclopedic
- Geographic
- Mathematics

Developer Resources

- Amazon Machine Images (AMIs)
- Articles & Tutorials
- Customer Apps

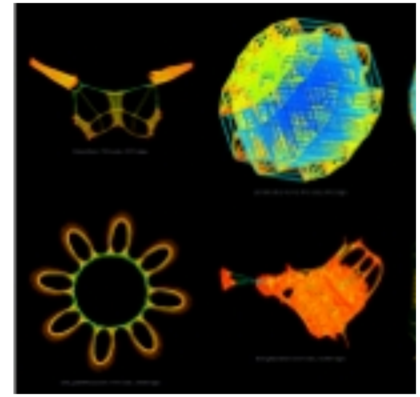
Amazon Public Datasets

- Wide Range of Domains
  - Astronomy, Biology, Climate, Economics, Mathematics, Encyclopedic , Geographic etc
- Wide Range of Sizes
  - 5 GB to 100s of TB
- Wide Range of Variations
  - Raw Data
  - Annotated Data

+ Example Datasets

Human Microbiome Project

SLOAN DIGITAL SKY SURVEY



University of Florida Sparse Matrix Collection

Wikipedia Traffic Statistics

WIKIPEDIA
The Free Encyclopedia

1000 Genomes
A Deep Catalog of Human Genetic Variation

OpenStreet Map

Quora Search

Data: Apple (company) Big Data Data Mining Data Science Datasets Linked Data Lists Open Data Open Science Science Scientific Research Social Sciences Statistics (collected data) Edit

Where can I find large datasets open to the public? Edit

Bigdata of web log files Edit

5+ Comments • Share (136) • Report • Options

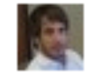
Answer Wiki

Here are many of the links mentioned so far:

Cross-disciplinary data repositories, data collections and data search engines:

- http://usgovxml.com
- http://aws.amazon.com/datasets
- http://databib.org
- http://datacite.org
- http://figshare.com
- http://linkeddata.org
- http://reddit.com/r/datasets
- http://thedatahub.org alias http://ckan.net
- http://quandl.com
- Social Network Analysis Interactive Dataset Library (Social Network Datasets)
- Datasets for Data Mining
- http://enigma.io

Single datasets and data repositories



Jeff Hammerbacher, Curious.

152

Votes by Gustav Meeuwenflatser, Robert Morton, Rob McQueen, Michael R. Bernstein, and 147 more.

I'll try to restrict my answers to datasets greater than 1 GB in size, and order my answers by the size of the dataset.

More than 1 TB

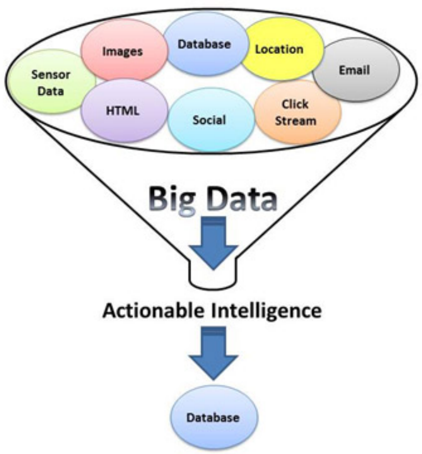
- The **1000 Genomes** project makes 260 TB of human genome data available [13]
- The **Internet Archive** is making an 80 TB web crawl available for research [17]
- The TREC conference made the **ClueWeb09** [3] dataset available a few years back. You'll have to sign an agreement and pay a nontrivial fee (up to $610) to cover the sneakernet data transfer. The data is about 5 TB compressed.
- **ClueWeb12** [21] is now available, as are the Freebase annotations, **FACC1** [22]
- **CNetS** at Indiana University makes a 2.5 TB click dataset available [19]
- **ICWSM** made a large corpus of blog posts available for their 2011 conference [2]. You'll have to register (an actual form, not an online form), but it's free. It's about 2.1 TB compressed.
- The **Proteome Commons** makes several large datasets the Personal Genome Project [11], is 1.1 TB in size. Ther over 100 GB in size.

From the same Quora page

More than 1 GB

- The **Reference Energy Disaggregation Data Set** [12] has data on home energy use; it's about 500 GB compressed.

+ Tips for Getting Started with your semester Project

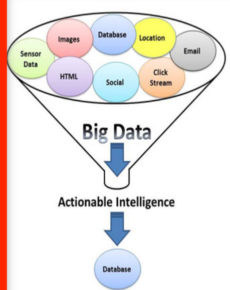


→ **Data Source** → **Understand Nature of Data**

Identify Problem Domain

Identify dataset you want to work with

Tips for Getting Started with your semester Project



Identify Problem Domain

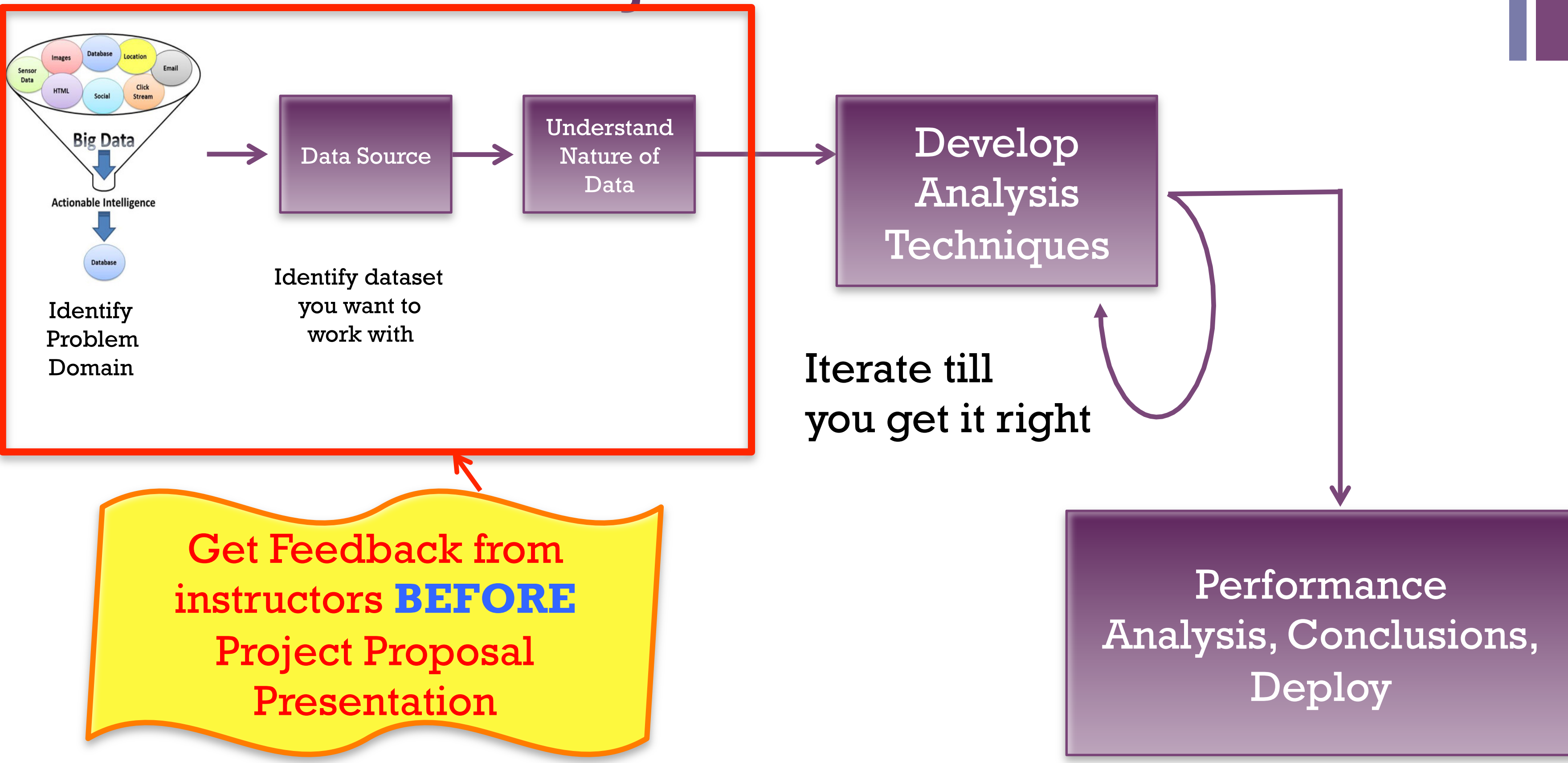
→ Data Source → Understand Nature of Data → Develop Analysis Techniques → Performance Analysis, Conclusions, Deploy

Identify dataset you want to work with

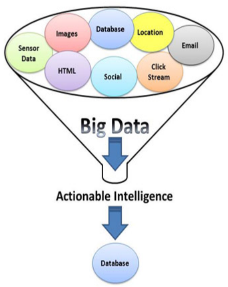
Iterate till you get it right

Can start right away

+ Tips for Getting Started with your semester Project



Tips for Getting Started with your semester Project



Identify Problem Domain

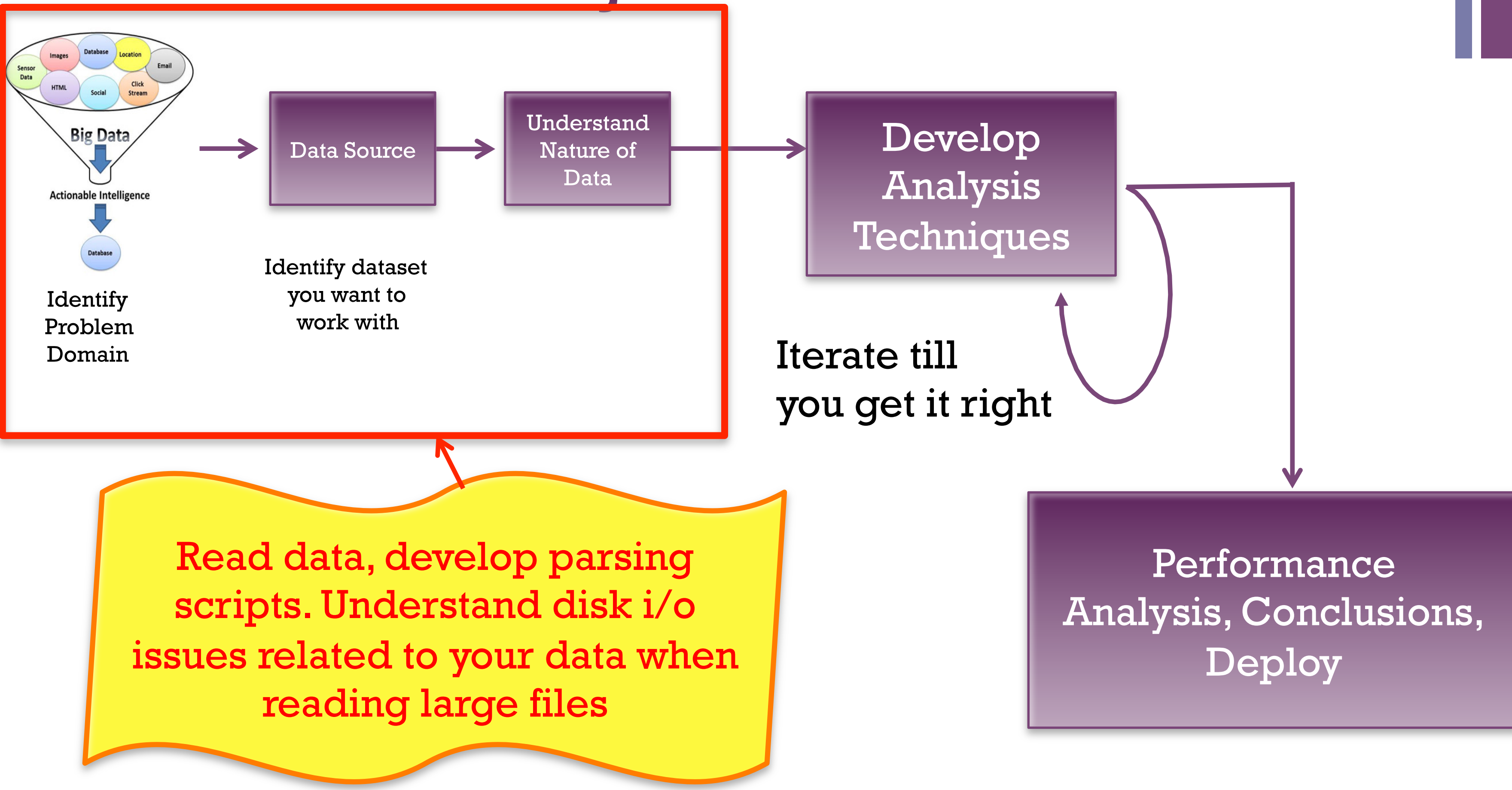
→ **Data Source** → **Understand Nature of Data** → **Develop Analysis Techniques** → **Performance Analysis, Conclusions, Deploy**

Identify dataset you want to work with

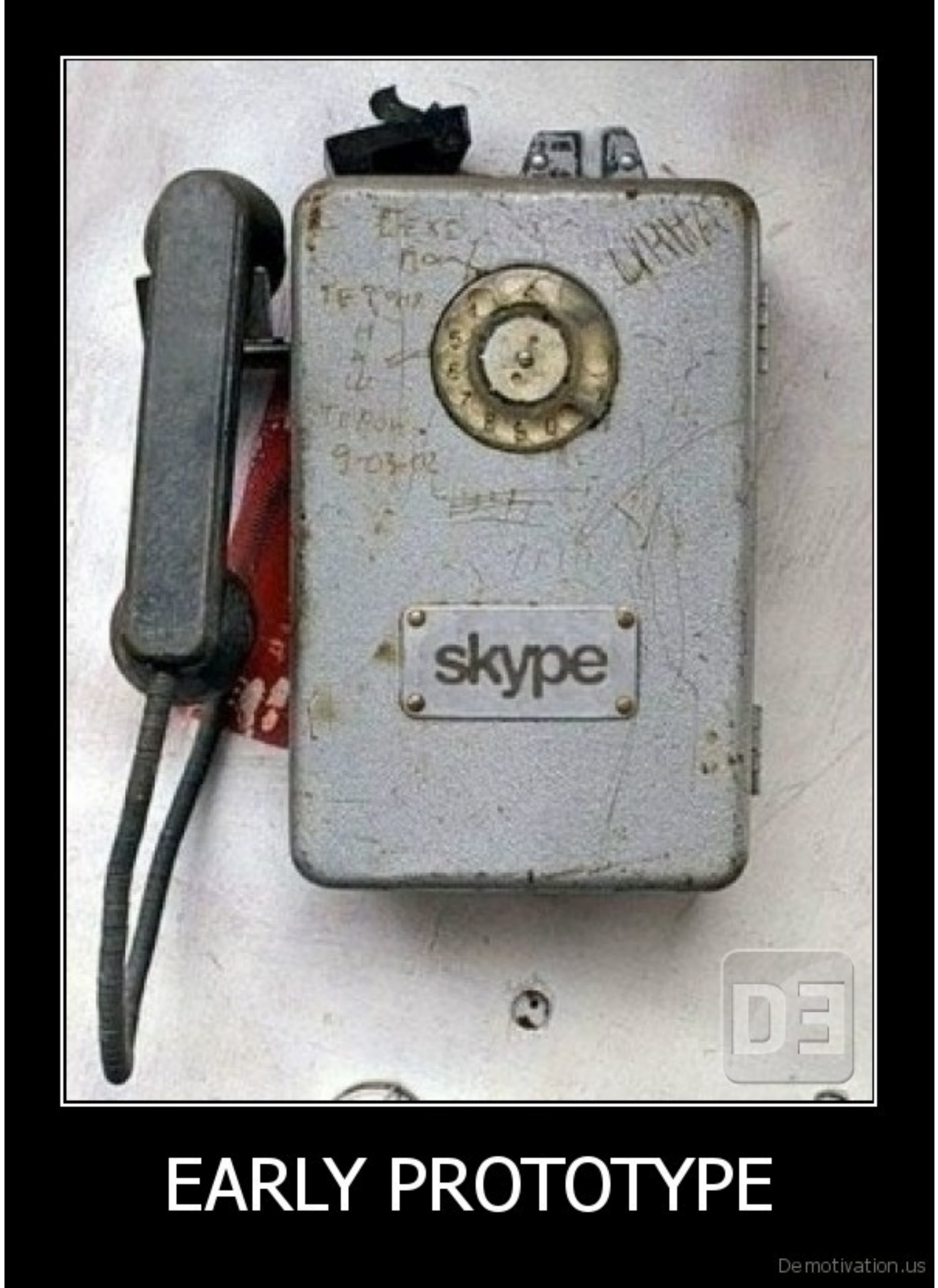
Iterate till you get it right

Develop a detailed design strategy, partial implementation. Get Feedback **DURING** proposal presentation

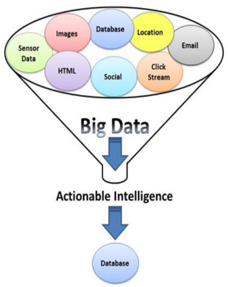
+ Tips for Getting Started with your semester Project



+


skype
EARLY PROTOTYPE
Demotivation.us

+ Tips for Getting Started with your semester Project



Identify Problem Domain

→ Data Source → Understand Nature of Data → Develop Analysis Techniques → Performance Analysis, Conclusions, Deploy

Identify dataset you want to work with

Iterate till you get it right

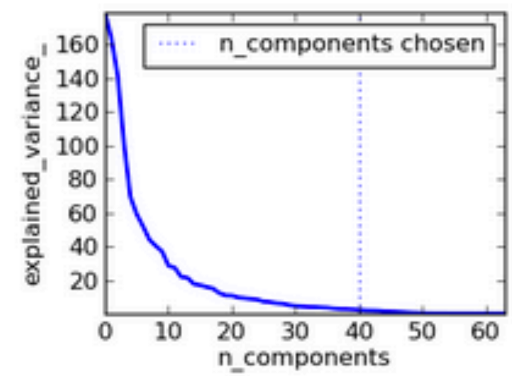
Create a small subset of the data.
Quickly prototype your idea
Test on small subset.
Iterate…
Deploy it to scale

Putting it all together

Pipelining

We have seen that some estimators can transform data and that some estimators can predict variables. We can also create combined estimators:

```
import pylab as pl

from sklearn import linear_model, decomposition, datasets

logistic = linear_model.LogisticRegression()

pca = decomposition.PCA()
from sklearn.pipeline import Pipeline
pipe = Pipeline(steps=[('pca', pca), ('logistic', logistic)])

digits = datasets.load_digits()
X_digits = digits.data
y_digits = digits.target

###############################################################
################
# Plot the PCA spectrum
pca.fit(X_digits)

pl.figure(1, figsize=(4, 3))
pl.clf()
pl.axes([.2, .2, .7, .7])
pl.plot(pca.explained_variance_, linewidth=2)
pl.axis('tight')
pl.xlabel('n_components')
pl.ylabel('explained_variance_')
```



Highly Recommended..!

+ Resources

http://snap.stanford.edu/class/cs341-2012/reports/index.html

By *Jure Leskovec* STANFORD UNIVERSITY

CS341
Project in Mining Massive Data Sets
Spring 2012

- Home
- Schedule
- Course info
- Data & Ideas

Project reports

- Resolving Ambiguity in Product/Brand Names by Evie Gillie, Saahil Shenoy, Corey Stein.
- Rethinking Recruitment: Learning Latent Interests for Better Job Recommendations by Anthony Chow, Chang Su, Junji Ma.
- Automated Essay Scoring and The Repair of Electronics by Dan Preston, Danny Goodman.
- Halloween Costume Predictions using Twitter by Anirudh Venkatesh, Onkar Dalal, Praveen Bommannavar.
- Social Data and College Statistics by Sean Choi, Elena Grewal, Kai Wen.
- Resolving Student Entities in the Facebook Social Graph by Jim Sproch, Jason Jong.
- Wikipedia: Nowhere to grow by Austin Gibbons, David Vetrano, Susan Biancani.
- Predicting User Purpose on BranchOut by Ben Holtz, Ben Lasley, Garrett Schlesinger.
- Adverse Event Profiles for Multi-drug Combinations by Srinivasan Iyer, Kushal Tayal, Siddhi Soman.
- Meetup Group Life Cycle Study by Tongda Zhang, Haomiao Jiang, Yinan Na.
- Wikipedia Mathematical Models and Reversion Prediction by Jia Ji, Bing Han, Dingyi Li.

Example Project Report

Social Data and College Statistics

Sean Choi
Stanford University
yo2seol@stanford.edu

Elena Grewal
Stanford University
etgrewal@stanford.edu

Kai Wen
Stanford University
kaiwen@stanford.edu

ABSTRACT

We correlate aspects of Twitter data with college statistics. We find that the amount of "buzz" about a college on twitter predicts the number of applicants to the college, even when controlling for the number of applicants in the previous year. We also explore various methods to classify the sentiment of tweets about a school. We find that the sentiment of insiders at a college predicts the freshman retention rate, but that this result is explained by average SAT score and school size. The sentiment of Twitter messages about a college does not predict the number of applicants, the acceptance rate, or the graduation rate. The paper adds to the growing literature on the predictive power of social data, documenting its strengths and limitations, and applies these techniques to a novel set of outcomes.

the average SAT score or average GPA of students in attendance.

We hypothesize that the "buzz" about a school as measured by the number of tweets that mention a school will be a positive predictor of the number of applicants to the school and the acceptance rate of applicants at the school. In addition, we hypothesize that the sentiment of tweets about a school will predict the number of applicants, as well as other outcomes such as the freshman retention rate and the graduation rate. The sentiments of Twitter users who know more about a school and possibly attend the school should be an even better indicator of measures such as the freshman retention rate and the graduation rate. If those people express positive sentiments about a school then we predict

+ Example Project Report

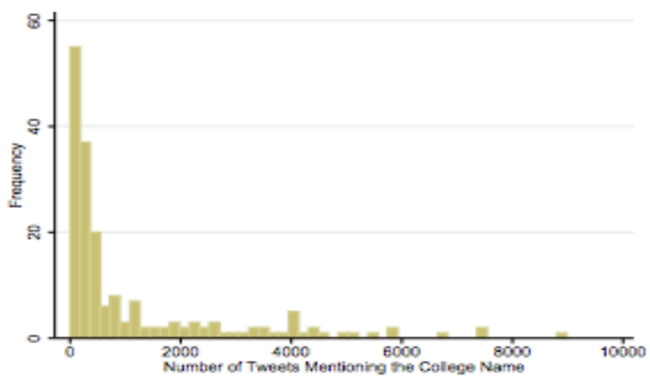


Figure 2: Distribution of the counts of mentions of colleges.

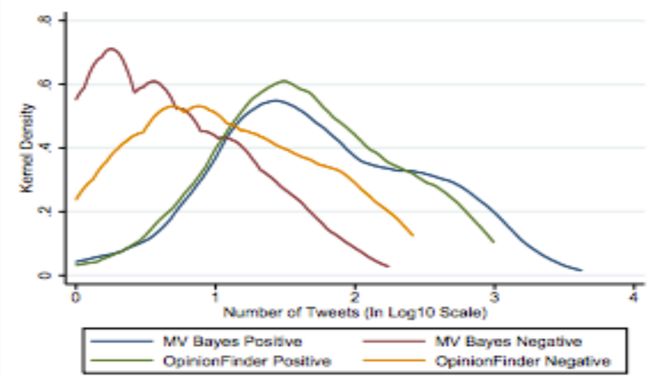


Figure 4: Kernel Density Plot of the Log of Number of Tweets Classified as Positive or Negative.

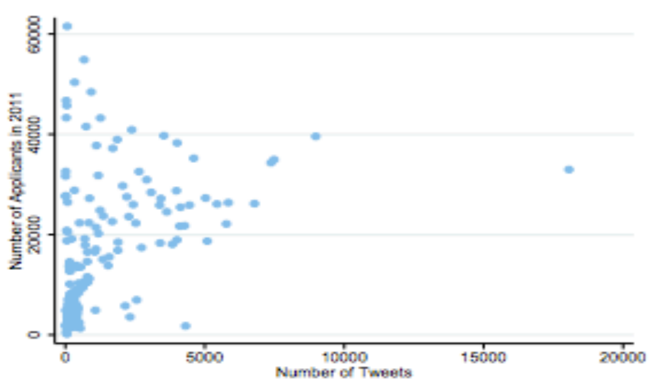


Figure 3: Number of tweets versus Number of Applicants in 2011.

on our hand-labeled test data, but the ratio of positive to negative tweets identifi... was much higher than the actual ratio. ... the Opin- ionFinder classifier fou... found a ratio of 9.5. T... likely that the MV Ba... positive tweets and tha... mating the ratio of pos... to negati...

Add concrete results from your analysis

5.3 Regression results

The regression results indicate that the count of the number of mentions of a college name is predictive of the number of applicants in 2011. The coefficient on the count is positive and statistically significant even when controlling for the number of applicants in 2010 and the size of the school and mean SAT score of the school. In contrast, the sentiment of the tweets is not predictive of the number of applicants when the number of applicants in the prior year is included as well as the other variables. The sample size changes because in

Write Report in IEEE Conference Format

IEEE
Advancing Technology for Humanity

The world's largest professional association for the advancement of technology

About IEEE | Membership & Services | Societies & Communities | Publications & Standards | Conferences & Events | Contac

Education & Careers | Search | Follow:

Home > Conferences & Events > Publishing

Manuscript Templates for Conference Proceedings

- Conference Organizers Menu
  - Running an IEEE Conference
- Quick Links
- Getting Started with Organizing a Conference
- Activities by Committee
  - Conference Organizer Education
  - POCO - Panel of Conference Organizers
  - Conference Organizers' Newsletter
  - Who Do You Talk to at IEEE?

Although IEEE does not require a specific format for their conference articles, IEEE eXpress Conference Publishing provides these optional MS Word and LaTeX templates free for use. If you wish, you may link to this Web page in its entirety. However, we do not recommend that you link to individual files because they may be updated or replaced without notice.

Grateful acknowledgement is made to the IEEE Computational Intelligence Society, which provided the current LaTeX template.

Note: Other templates (maintained by trans@ieee.org) that more closely align with the printed Transactions format are available.

Accessing the templates

- Select **Save** when the File Download window appears. The files cannot open directly from the server.

Microsoft Word 2003	LaTeX Archive Contents (PDF, 63 KB)	LaTeX (Bibliography Files)*
US letter (DOC, 58 KB)	Unix (TAR.GZ, 683 KB)	Unix (TAR.GZ, 307 KB)

Deliverables

- During Proposal Presentation
  - 1 Page Report
  - Project Proposal Presentation
- During Final Presentation
  - Detailed Report (IEEE Conference Format)
  - Presentation
  - You might be asked to show demo, data, results, code


KEEP
CALM
WE'RE
HERE TO
HELP

Use emails aggressively

- Terry Boult: tboult@vast.uccs.edu
- Abhijit Bendale abendale@vast.uccs.edu, abhijitbendale@gmail.com
- We know it is a tough course
  - Start early
  - Seek more help earlier in the semester and gain more independence later on
  - There is not much homework: doesn't mean you should not do anything..
  - Use code given in documentation to develop your understanding
- For each library/tool there are tons of tutorials/videos available online.