

# Gentle Introduction to Machine Learning with scikit-learn

Rob Zinkov

January 19th, 2012

# Outline

- 1 Introduction
- 2 Machine Learning Basics
- 3 Scikit-Learn
- 4 Conclusion

# What is the point of this talk?

- Get you playing around with Machine Learning techniques
- Get you excited about scikit-learn

# Caveats

- This talk won't change your life
- I won't focus too much on techniques
- This talk is low on math
- This talk won't make you an expert in scikit-learn

# What is Machine Learning?

Machine Learning is the art of creating a compact explanation of the world using a large amount of data from the world

# Definitons

- **Model** the collection of parameters you are trying to fit
- **Data** what you are using to fit the model
- **Target** the value you are trying to predict with your model
- **Features** attributes of your data that will be used in prediction
- **Methods** algorithms that will use your data to fit a model

Note: Many methods are made to fit particular models

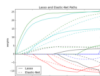
# Which method should I use?



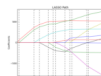
*Automatic Relevance  
Determination Regression  
(ARD)*



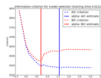
*Bayesian Ridge  
Regression*



*Lasso and Elastic Net*



*Lasso path using LARS*



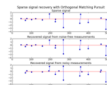
*Lasso model selection:  
Cross-Validation / AIC /  
BIC*



*Path with L1-Logistic  
Regression*



*Ordinary Least Squares*



*Orthogonal Matching  
Pursuit*



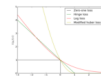
*Polynomial interpolation*



*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



But yes its not the important question

Good Features are more important than Good Methods

# Good Features is what counts



# Simple Example Model

OK I don't want to cheat you

# Meet the Boston Housing Dataset



```

Boston House Prices dataset

Notes
-----
Data Set Characteristics:

: Number of Instances: 506

: Number of Attributes: 13 numeric/categorical predictive

: Median Value (attribute 14) is usually the target

: Attribute Information (in order):
- CRIM    per capita crime rate by town
- ZN      proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS   proportion of non-retail business acres per town
- CHAS    Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX     nitric oxides concentration (parts per 10 million)
- RM      average number of rooms per dwelling
- AGE     proportion of owner-occupied units built prior to 1940
- DIS     weighted distances to five Boston employment centres
- RAD     index of accessibility to radial highways
- TAX     full-value property-tax rate per $10,000
- PTRATIO pupil-teacher ratio by town
- B       1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT   % lower status of the population
- MEDV    Median value of owner-occupied homes in $1000's

: Missing Attribute Values: None

: Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
http://archive.ics.uci.edu/ml/datasets/Housing

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980. N.B. Various transformations are used in the table on
pages 244-261 of the latter.

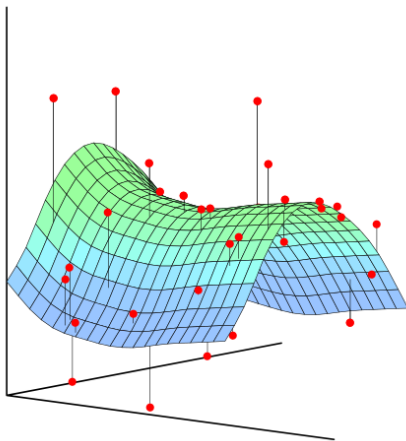
```

```

In [21]: boston.data[1:100,1:6]
Out[21]:
array([[ 0. ,  7.07 ,  0. ,  0.469 ,  6.421 ],
       [ 0. ,  7.07 ,  0. ,  0.469 ,  7.185 ],
       [ 0. ,  2.18 ,  0. ,  0.458 ,  6.998 ],
       [ 0. ,  2.18 ,  0. ,  0.458 ,  7.147 ],
       [ 0. ,  2.18 ,  0. ,  0.458 ,  6.43  ],
       [12.5 ,  7.07 ,  0. ,  0.524 ,  6.012 ],
       [12.5 ,  7.07 ,  0. ,  0.524 ,  6.172 ],
       [12.5 ,  7.07 ,  0. ,  0.524 ,  5.631 ],
       [12.5 ,  7.07 ,  0. ,  0.524 ,  6.004 ],
       [12.5 ,  7.07 ,  0. ,  0.524 ,  6.377 ],
       [12.5 ,  7.07 ,  0. ,  0.524 ,  6.009 ],
       [12.5 ,  7.07 ,  0. ,  0.524 ,  5.889 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.949 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  6.096 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.834 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.935 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.99  ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.456 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.727 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.57  ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.965 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  6.142 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.813 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.924 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.599 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.813 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  6.047 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  6.405 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  6.674 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.713 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  6.072 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.95  ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  5.701 ],
       [ 0. ,  8.14 ,  0. ,  0.538 ,  6.096 ],
       [ 0. ,  5.96 ,  0. ,  0.499 ,  5.933 ],
       [ 0. ,  5.96 ,  0. ,  0.499 ,  5.841 ],
       [ 0. ,  5.96 ,  0. ,  0.499 ,  5.85  ],
       [ 0. ,  5.96 ,  0. ,  0.499 ,  5.966 ],
       [75. ,  2.95 ,  0. ,  0.428 ,  6.595 ],
       [75. ,  2.95 ,  0. ,  0.428 ,  7.024 ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  6.77  ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  6.169 ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  6.211 ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  6.069 ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  5.682 ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  5.786 ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  6.03  ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  5.399 ],
       [ 0. ,  6.91 ,  0. ,  0.448 ,  5.682 ],
       [21. ,  5.64 ,  0. ,  0.439 ,  5.963 ],

```

This data plotted might resemble this



We assume these numbers can be linearly combined to predict housing price

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + \dots + w_Dx_D$$



## Now this is how its done

```
from sklearn.linear_model import RidgeRegression
from sklearn import datasets

boston = datasets.load_boston()
X = boston.data
y = boston.target

clf = RidgeRegression()
clf.fit(X, y)
clf.predict(X)
```

# What just happened?

# Why Scikit-Learn?

- Weka is terrible

# No really it is very bad

- Documentation is scattered
- Interfaces are terrible
- Code has well-known bugs
- Software is not actively maintained
- It's Java code of the worst kind

# Why Scikit-Learn?

- Weka is terrible
- Most libraries are just research code

# If you've been there you know

- Documentation is non-existent (libsvm)
- Interfaces are idiosyncratic
- Software is unmaintained (libsvm, pybrain)
- Software is for educational purposes (nltk)


# Why Scikit-Learn?

- Made on top of cython and scipy



- Fantastic community

# Full of practitioners and researchers



## scikit-learn (scikit-learn)

---

Name **scikit-learn**

Website/Blog <http://scikit-learn.org>

Member Since **Aug 16, 2010**

**5**

Public Repos


**26**

Members

---


### Public Repositories (5)

All Repositories Sources Forks Mirrors


 **scikit-learn**
C ↔ 390 ↗ 176

scikit-learn: machine learning in Python

Last updated about 6 hours ago




52 week participation


 **scikit-learn.org**
Python ↔ 2 ↗ 1

Source repository to build the HTML website for the scikit-learn project.

Last updated 4 days ago




52 week participation

 **ml-benchmarks**
Python ↔ 11 ↗ 6













Benchmarks for various machine learning packages


Last updated October 07, 2011



52 week participation

### Organization Members (26)

-  **agramfort** (Alexandre Gramfort)  
19 Public Repositories, 36 followers
-  **alextp** (Alexandre Passos)  
4 Public Repositories, 16 followers
-  **amueller** (Andreas Mueller)  
7 Public Repositories, 4 followers
-  **bdholt1** (Brian Holt)  
6 Public Repositories, 3 followers
-  **bthirion** (bthirion)  
7 Public Repositories, 6 followers
-  **christfilo** (Chris Filo Gorgolewski)  
15 Public Repositories, 4 followers
-  **cournape** (David Coumpeau)  
40 Public Repositories, 33 followers
-  **duchesnay**  
1 Public Repositories, 2 followers
-  **dwf** (David Warde-Farley)  
24 Public Repositories, 38 followers
-  **fabianp** (Fabian Pedregosa)  
11 Public Repositories, 33 followers
-  **GaelVaroquaux** (Gael Varoquaux)  
21 Public Repositories, 57 followers
-  **glouppe** (Gilles Louppe)  
1 Public Repositories, 11 followers



Rob Zinkov ()

Gentle Introduction to Machine Learning with

January 19th, 2012

24 / 39





# Very Active

## scikit-learn / Commit History


Keyboard shortcuts available 


**Jan 19, 2012**




**Mutual Information docstring incorrectly said it was the adjusted mut...** 

robertlayton authored about 6 hours ago


[1b81d3b98e](#) 


[Browse code](#) 




**Fix doctest.**

mblondel authored about 6 hours ago


[11f4c071fb](#) 


[Browse code](#) 





**Merge branch 'warm\_start'**

mblondel authored about 7 hours ago



[58b348e559](#) 


[Browse code](#) 





**Revert "COSMIT refactor SGD code further"** 

larsmans authored about 12 hours ago


 1 [8e83572c2a](#) 


[Browse code](#) 




**COSMIT refactor SGD code further** 

larsmans authored about 15 hours ago


[c912ab0008](#) 


[Browse code](#) 





**Make sure order="C".**

mblondel authored about 19 hours ago


[0ae85d2034](#) 


[Browse code](#) 




**Merge branch 'warm\_start' of github.com:mblondel/scikit-learn into wa...** 

mblondel authored about 19 hours ago

[f8ebc29f7b](#) 


[Browse code](#) 


**Jan 18, 2012**




**Suppress deprecation warnings.**

mblondel authored 1 day ago



[835439ffed](#) 


[Browse code](#) 



**Remove if statement.**

mblondel authored 1 day ago

 1 [d21e74fcc6](#) 

[Browse code](#) 

## Very pythonic

**learn** | [Home](#) | [Tutorial](#) | [User Guide](#) | [Examples](#) | [References](#) |

**User guide: contents**

**1. Installing scikit-learn**

**1.1. Installing an official release**

- 1.1.1. Installing from source
- 1.1.1.1. Easy install
- 1.1.1.2. From source tarball
- 1.1.2. Windows installer
- 1.1.3. Building on windows

**1.2. Third party distributions of scikit-learn**

- 1.2.1. Debian and derivatives (Ubuntu)
- 1.2.2. PyPi (Python)
- 1.2.3. Conda/Anaconda Python distribution
- 1.2.4. RMagick
- 1.2.5. RWEIO

**1.3. Bleeding Edge**

**1.4. Testing**

**2. Getting started: an introduction to machine learning with scikit-learn**

**2.1. Machine learning: the problem setting**

**2.2. Loading an example dataset**

**2.3. Learning and Predicting**

**2.4. Model persistence**

**3. Supervised learning**

**3.1. Generalized Linear Models**

- 3.1.1. Ordinary Least Squares
- 3.1.1.1. Linear Least Squares Compressor
- 3.1.2. Ridge Regression
- 3.1.2.1. Ridge Complexity
- 3.1.2.2. Solving the regularization parameter: generalized Cross Validation
- 3.1.3. Lasso
- 3.1.3.1. Solving regularization problems
  - 3.1.3.1.1. Using `cvxopt`
  - 3.1.3.1.2. Initialization/iterative based solver solution
- 3.1.4. Elastic Net
- 3.1.5. Least Angle Regression
- 3.1.6. LARS Lasso
- 3.1.7. Orthogonal Matching Pursuit (OMP)
- 3.1.8. Bayesian Regression
- 3.1.8.1. Bayesian Linear Regression
- 3.1.8.2. Automatic Relevance Determination - ARD
- 3.1.9. Logistic regression
- 3.1.10. Stochastic Gradient Descent - SGD

**3.2. Support Vector Machines**

- 3.2.1. Classification
- 3.2.1.1. Multi-class classification
- 3.2.1.2. Classification problems
- 3.2.2. Regression
- 3.2.3. Sparse estimation, novelty detection
- 3.2.4. Support Vector machines for sparse data
- 3.2.5. Complexity
- 3.2.6. Tips on Practical Use
- 3.2.7. Kernel functions
- 3.2.7.1. Custom Kernels
  - 3.2.7.1.1. Using kernel functions as matrices
  - 3.2.7.1.2. Using the Gram matrix
- 3.2.8. Mathematical formulation
- 3.2.8.1. SVC
- 3.2.8.2. NuSVC
- 3.2.9. Implementation details

**3.3. Stochastic Gradient Descent**

# Getting it: You want the latest

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

## A more *realistic* example

*That's what she said*

## Loading the example

```
import numpy as np
y = np.concatenate((np.zeros(5796),np.ones(2091)))

DATADIR = "/home/zv/custom_builds/twss-classifier/data"
data = itertools.chain(file(DATADIR+"/fmylife-parsed.txt"),
                       file(DATADIR+"/texts-from-last-night-pa",
                             file(DATADIR+"/twss-stories-parsed.txt")
```

# Feature Extraction

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data)
```

# Learning

```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB(0.01)
clf.fit(X,y)
```

# Testing

```
text = "Something inappropriate"  
if clf.predict_proba(vectorizer.transform(text)) > 0.995:  
    print "TWSS"
```



# Why this matters?

Machine Learning in Python isn't a coincidence

# Why this matters?

Python has a robust data ecosystem

- numpy
- scipy
- cython
- pandas

# Why this matters?

- Python is what data scientists are using
- Python will become the center of the data science universe

# Conclusions

Scikit-Learn is awesome, pythonic and fast

# Conclusions

Now go make some cool!

# References

- <http://www.scikit-learn.org>
- <https://github.com/scikit-learn/scikit-learn>
- <http://nltk.org>
- <http://wit.io/posts/ruby-is-beautiful-but-im-moving-to-python>

# Questions?