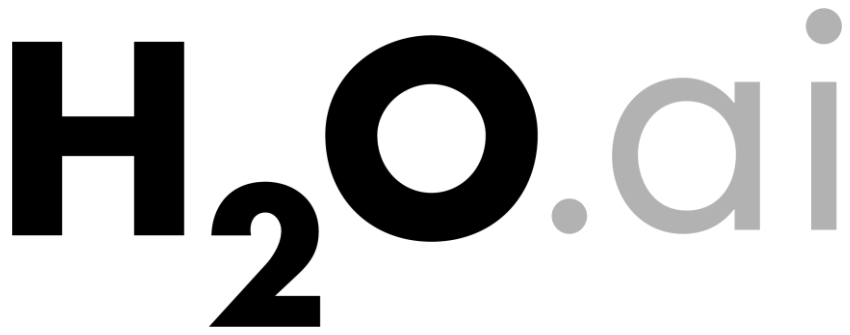
Machine Learning with H_2O



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

@matlabulous

ODSC Masterclass Summit

1st March, 2017

About Me

- Civil (Water) Engineer

  2010 – 2015

  - Consultant (UK)
    - Utilities
    - Asset Management
    - Constrained Optimization
  - Industrial PhD (UK)
    - Infrastructure Design Optimization
    - Machine Learning + Water Engineering
    - Discovered H_2O in 2014

- Data Scientist

  2015

  - Virgin Media (UK)
  - Domino Data Lab (Silicon Valley)

  2016 – Present

  - H_2O.ai (Silicon Valley)

Learning Objectives

- Start and connect to a local H_2O cluster from R/Python.
- Import data from R/Python data frames, local files or web.
- Perform basic data transformation and exploration.
- Train classification and regression models using H_2O machine learning algorithms.
- Evaluate models and make predictions.
- Improve performance by tuning and stacking.

H_2O Examples in R & Python

bit.ly/odsc_h2o_ml_2017

Materials for Machine Learning with H2O Open Platform at ODSC Masterclass Summit 2017

Edit

New Add topics

20 commits | 1 branch | 0 releases | 1 contributor | Apache-2.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

woobe draft — Latest commit 2a7dd40 8 hours ago

LICENSE	Initial commit	5 days ago
README.md	Updated content	2 days ago
kaggle_titanic.csv	Titanic dataset from Kaggle (https://www.kaggle.com/c/titanic/data)	5 days ago
py_01_data_in_h2o.ipynb	renamed	2 days ago
py_02_data_manipulation.ipynb	draft	2 days ago
py_03a_regression_basics.ipynb	renamed	2 days ago
py_03b_regression_grid_search.ipynb	renamed	2 days ago
py_03c_regression_ensembles.ipynb	draft	8 hours ago
py_04a_classification_basics.ipynb	renamed	2 days ago
py_04b_classification_ensembles.ipy...	renamed	2 days ago
r_01_data_in_h2o.ipynb	renamed	2 days ago
r_02_data_manipulation.ipynb	draft	2 days ago
r_03a_regression_basics.ipynb	draft	9 hours ago
r_03b_regression_grid_search.ipynb	draft	9 hours ago
r_03c_regression_ensembles.ipynb	draft	8 hours ago
winequality-white.csv	Wine Quality Data from https://archive.ics.uci.edu/ml/machine-learnin...	5 days ago

About H_2O.ai

Company Overview

Founded	2011 Venture-backed, debuted in 2012
Products	• H_2O Open Source In-Memory AI Prediction Engine • Sparkling Water • Steam
Mission	Operationalize Data Science, and provide a platform for users to build beautiful data products
Team	70 employees • Distributed Systems Engineers doing Machine Learning • World-class visualization designers
Headquarters	Mountain View, CA




Joe the Outlier

Figure 1. Magic Quadrant for Data Science Platforms

Source: Gartner (February 2017)

Joe (Water Engineer)

http://www.h2o.ai/gartner-magic-quadrant/

H2O.ai

Algorithms Overview

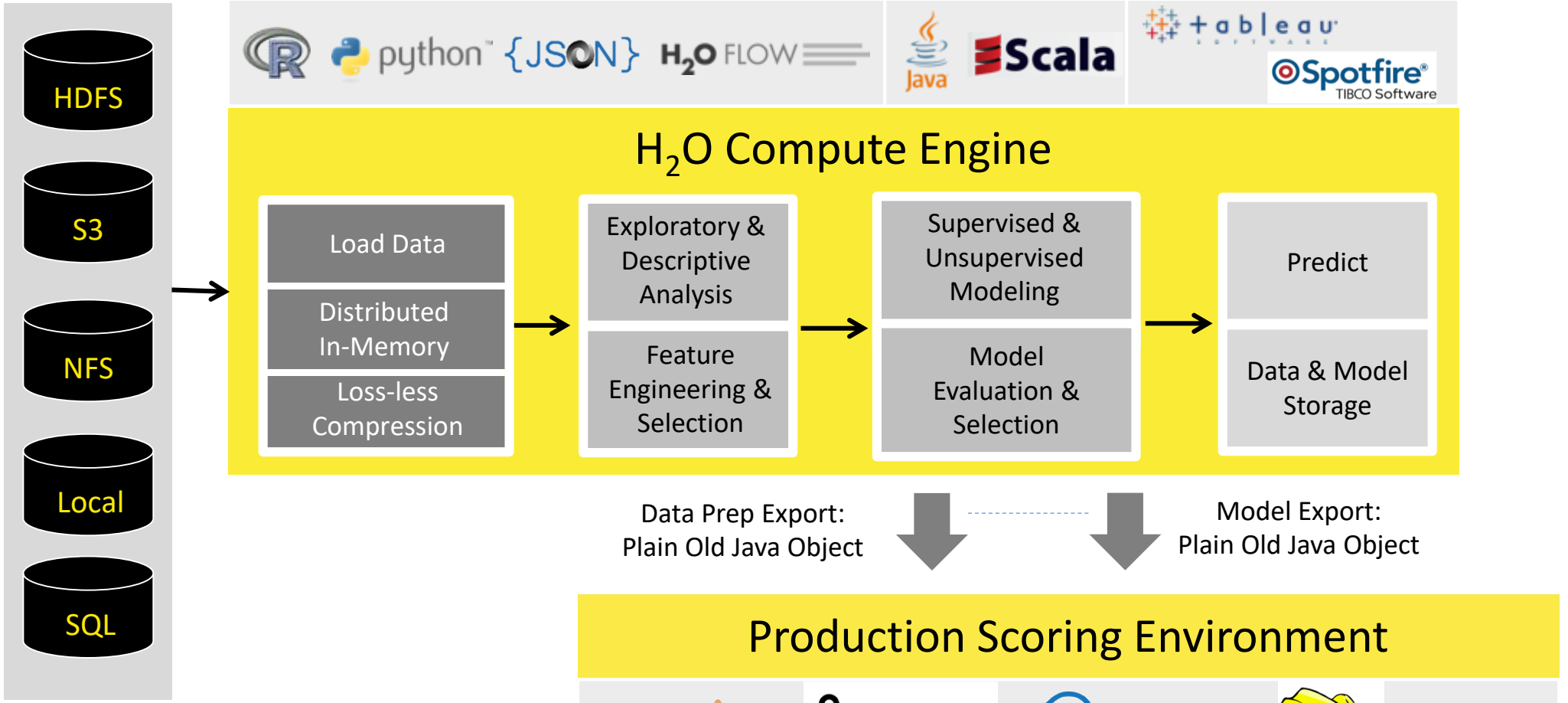
Supervised Learning

Statistical Analysis	• **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations
Deep Neural Networks	• **Deep learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

Unsupervised Learning

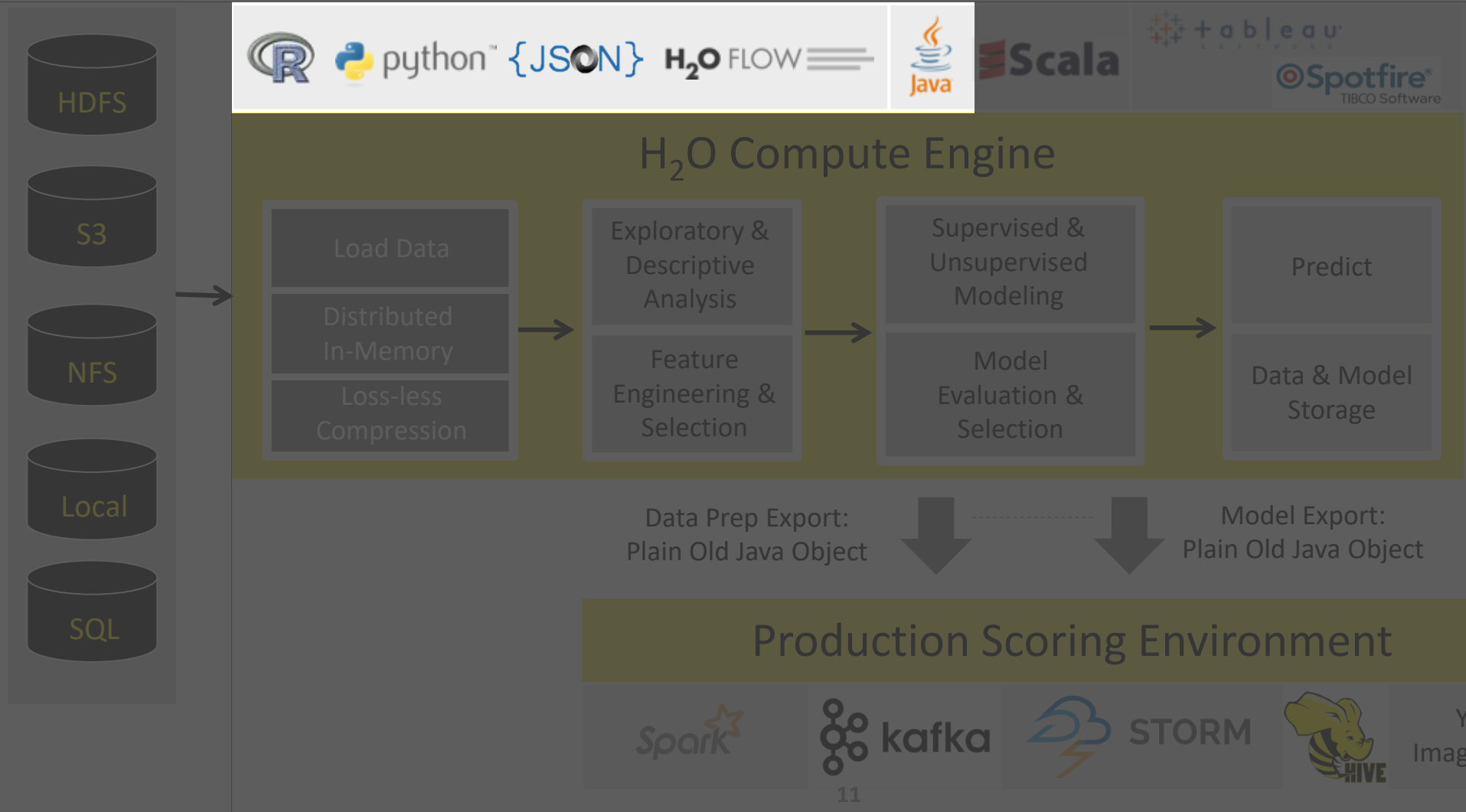
Clustering	• **K-means**: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k
Dimensionality Reduction	• **Principal Component Analysis**: Linearly transforms correlated variables to independent components • **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data
Anomaly Detection	• **Autoencoders**: Find outliers using a nonlinear dimensionality reduction using deep learning

High Level Architecture



High Level Architecture

Flow (Web), R, Python API
Java for computation



docs.h2o.ai

Languages

R

- Quick Start Video - R
- R Package Docs
- R Booklet
- Examples and Demos
- R FAQ
- Ensemble R Package Readme
- RSparkling Readme
- Migrating from H2O-2

Python

- Quick Start Video - Python
- Python Module Docs
- Python Booklet
- Examples and Demos
- Python FAQ
- PySparkling Readme 2.0 | 1.6
- skutil Docs

Java

- POJO and MOJO Model Javadoc
- H2O Core Javadoc
- H2O Algorithms Javadoc

Scala

Sparkling Water API	2.0	1.6
Sparkling Water Scaladoc	2.0	1.6
H2O Scaladoc	2.11	2.10

Tutorials, Examples, & Presentations

Tutorials and Blogs

- H2O Tutorials HTML | PDF
- H2O Blogs
- H2O University

Use Case Examples

Chicago crime prediction	R	Python	ScalaSW	PySW
Airlines delays prediction	R	Python	ScalaSW	PySW
Lending Club loan prediction	R	Python	ScalaSW	PySW
Ham or Spam	R	Python	ScalaSW	PySW
Prediction with prostate dataset	R	Python	ScalaSW	PySW

Presentations

- H2O Meetups
- H2O World 2014 Videos
- H2O World 2015 Videos
- Open Tour Chicago Videos
- Open Tour NYC Videos
- Open Tour Dallas Videos

Start and Connect to a Local H2O Cluster

…_01_data_in_h2o.ipynb

In [1]:
```
# Start and connect to a local H2O cluster
import h2o
h2o.init(nthreads = -1)
```

```
Checking whether there is an H2O instance running at http://localhost:54321..... not found.
Attempting to start a local H2O server...
  Java Version: java version "1.8.0_121"; Java(TM) SE Runtime Environment (build 1.8.0_121-b13); Java HotS
pot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
  Starting server from /home/joe/anaconda3/lib/python3.5/site-packages/h2o/backend/bin/h2o.jar
  Ice root: /tmp/tmpbz7az1l0
  JVM stdout: /tmp/tmpbz7az1l0/h2o_joe_started_from_python.out
  JVM stderr: /tmp/tmpbz7az1l0/h2o_joe_started_from_python.err
  Server is running at http://127.0.0.1:54321
Connecting to H2O server at http://127.0.0.1:54321... successful.
```

H2O cluster uptime:	01 secs
H2O cluster version:	3.10.3.5
H2O cluster version age:	6 days
H2O cluster name:	H2O_from_python_joe_88i2yg
H2O cluster total nodes:	1
H2O cluster free memory:	5.210 Gb
H2O cluster total cores:	8
H2O cluster allowed cores:	8
H2O cluster status:	accepting new members, healthy
H2O connection url:	http://127.0.0.1:54321
H2O connection proxy:	None
Python version:	3.5.2 final

In [1]:
```
# Start and connect to a local H2O cluster
suppressPackageStartupMessages(library(h2o))
h2o.init(nthreads = -1)
```

```
H2O is not running yet, starting it now...

Note:  In case of errors look at the following log files:
    /tmp/RtmpY7Pu5l/h2o_joe_started_from_r.out
    /tmp/RtmpY7Pu5l/h2o_joe_started_from_r.err


Starting H2O JVM and connecting: .. Connection successful!

R is connected to the H2O cluster:
    H2O cluster uptime:         2 seconds 196 milliseconds
    H2O cluster version:        3.10.3.5
    H2O cluster version age:    6 days
    H2O cluster name:           H2O_started_from_R_joe_jiw390
    H2O cluster total nodes:    1
    H2O cluster total memory:   5.21 GB
    H2O cluster total cores:    8
    H2O cluster allowed cores:  8
    H2O cluster healthy:        TRUE
    H2O Connection ip:          localhost
    H2O Connection port:        54321
    H2O Connection proxy:       NA
    R Version:                  R version 3.3.2 (2016-10-31)
```

Importing Data into H_2O

…_01_data_in_h2o.ipynb

In [2]:
```
# Method 1 - Import data from a local CSV file
data_from_csv = h2o.import_file("winequality-white.csv")
data_from_csv.head(5)
```

Parse progress: |██| 100%

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6

Out[2]:

In [3]:
```
# Method 2 - Import data from the web
data_from_web = h2o.import_file("https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv")
data_from_web.head(5)
```

Parse progress: |██| 100%

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6

Out[3]:

In [4]:
```
# Method 3 - Convert Python data frame into H2O data frame

## Import Wine Quality data using Pandas
import pandas as pd
wine_df = pd.read_csv('winequality-white.csv', sep = ';')
wine_df.head(5)
```

Out[4]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

In [5]:
```
## Convert Pandas data frame into H2O data frame
data_from_df = h2o.H2OFrame(wine_df)
data_from_df.head(5)
```

Parse progress: |  | 100%

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6

Out[5]:

In [4]:
```
# Method 3 - Convert R data frame into H2O data frame

## Import Wine Quality data using R
wine_df = read.csv('winequality-white.csv', sep = ';')
head(wine_df, 5)
```

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sul
7.0	0.27	0.36	20.7	0.045	45	170	1.0010	3.00	0.4
6.3	0.30	0.34	1.6	0.049	14	132	0.9940	3.30	0.4
8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.4
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4

In [5]:
```
## Convert R data frame into H2O data frame
data_from_df = as.h2o(wine_df)
head(data_from_df, 5)
```

```
|======================================================================| 100%
```

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sul
7.0	0.27	0.36	20.7	0.045	45	170	1.0010	3.00	0.4
6.3	0.30	0.34	1.6	0.049	14	132	0.9940	3.30	0.4
8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.4
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4

Basic Data Transformation & Exploration

…_02_data_manipulation.ipynb

(see notebooks)

Regression & Classification Models (Basics)

…_03a_regression_basics.ipynb

…_04a_classification_basics.ipynb

Algorithms Overview

Supervised Learning

Statistical Analysis	• **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations
Deep Neural Networks	• **Deep learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

Unsupervised Learning

Clustering	• **K-means**: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k
Dimensionality Reduction	• **Principal Component Analysis**: Linearly transforms correlated variables to independent components • **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data
Anomaly Detection	• **Autoencoders**: Find outliers using a nonlinear dimensionality reduction using deep learning

docs.h2o.ai

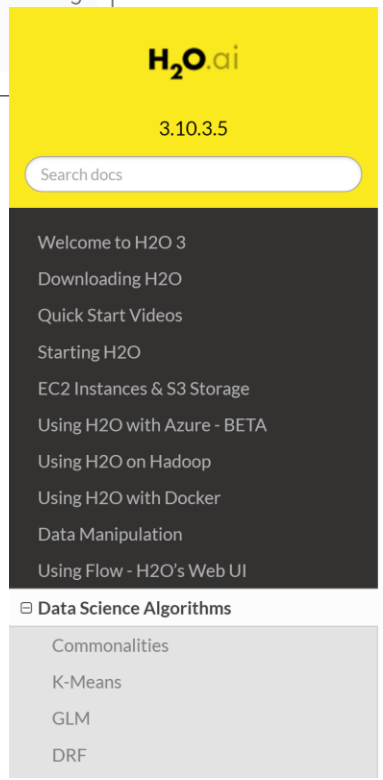
Data Science Algorithms

Supervised Learning

Generalized Linear Modeling (GLM)	Tutorial	Booklet	Reference	Tuning
Gradient Boosting Machine (GBM)	Tutorial	Booklet	Reference	Tuning
Deep Learning	Tutorial	Booklet	Reference	Tuning
Distributed Random Forest	Tutorial	Booklet	Reference	Tuning
Naive Bayes	Tutorial	Booklet	Reference	Tuning
Ensembles (Stacking)	Tutorial	Booklet	Reference	

Unsupervised Learning

Generalized Low Rank Models (GLRM)	Tutorial	Reference
K-Means Clustering	Tutorial	Reference
Principal Components Analysis (PCA)	Tutorial	Reference



Docs » Data Science Algorithms » GBM View page source

GBM

Introduction

Gradient Boosting Machine (for Regression and Classification) is a forward learning ensemble method. The guiding heuristic is that good predictive results can be obtained through increasingly refined approximations. H2O's GBM sequentially builds regression trees on all the features of the dataset in a fully distributed way - each tree is built in parallel.

The current version of GBM is fundamentally the same as in previous versions of H2O (same algorithmic steps, same histogramming techniques), with the exception of the following changes:

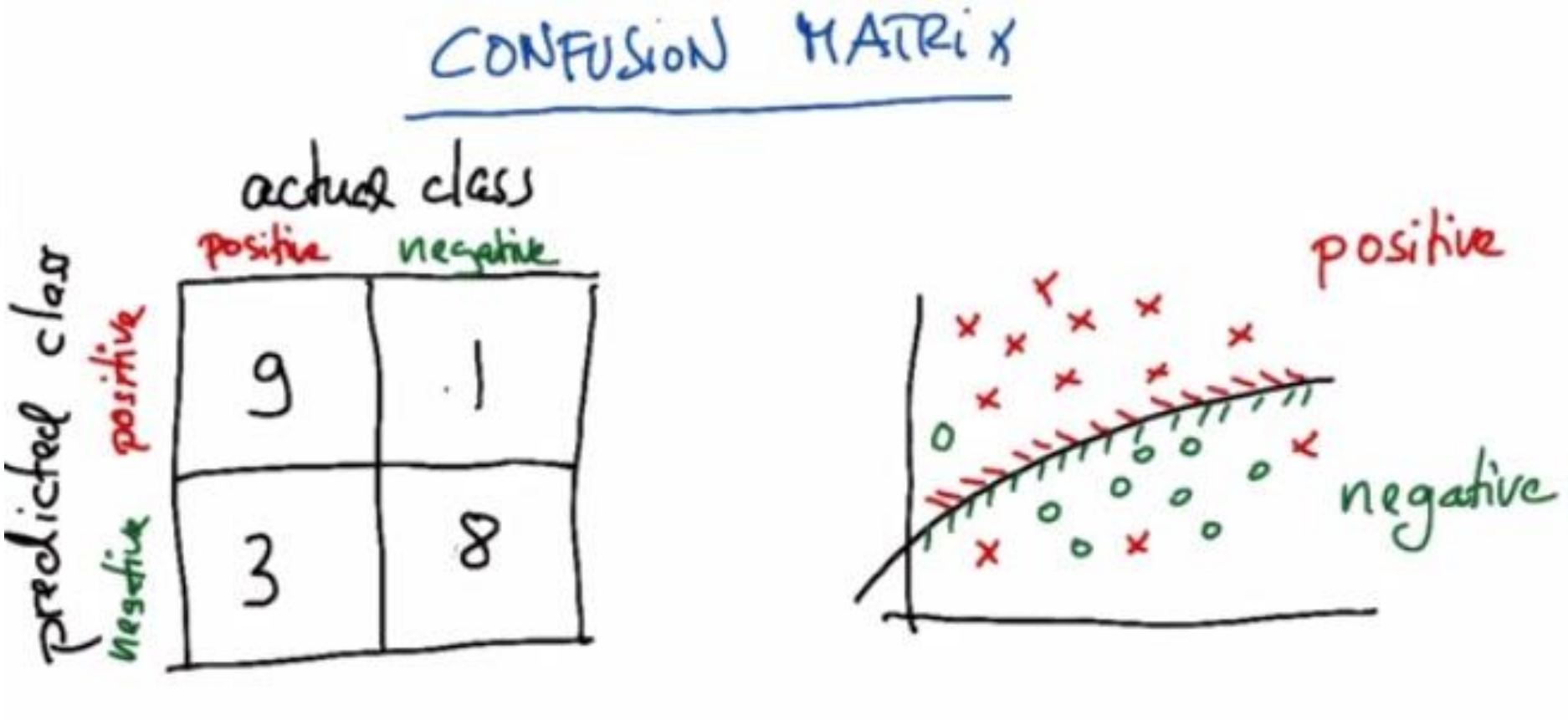
- Improved ability to train on categorical variables (using the `nbins_cats` parameter)
- Minor changes in histogramming logic for some corner cases

There was some code cleanup and refactoring to support the following features:

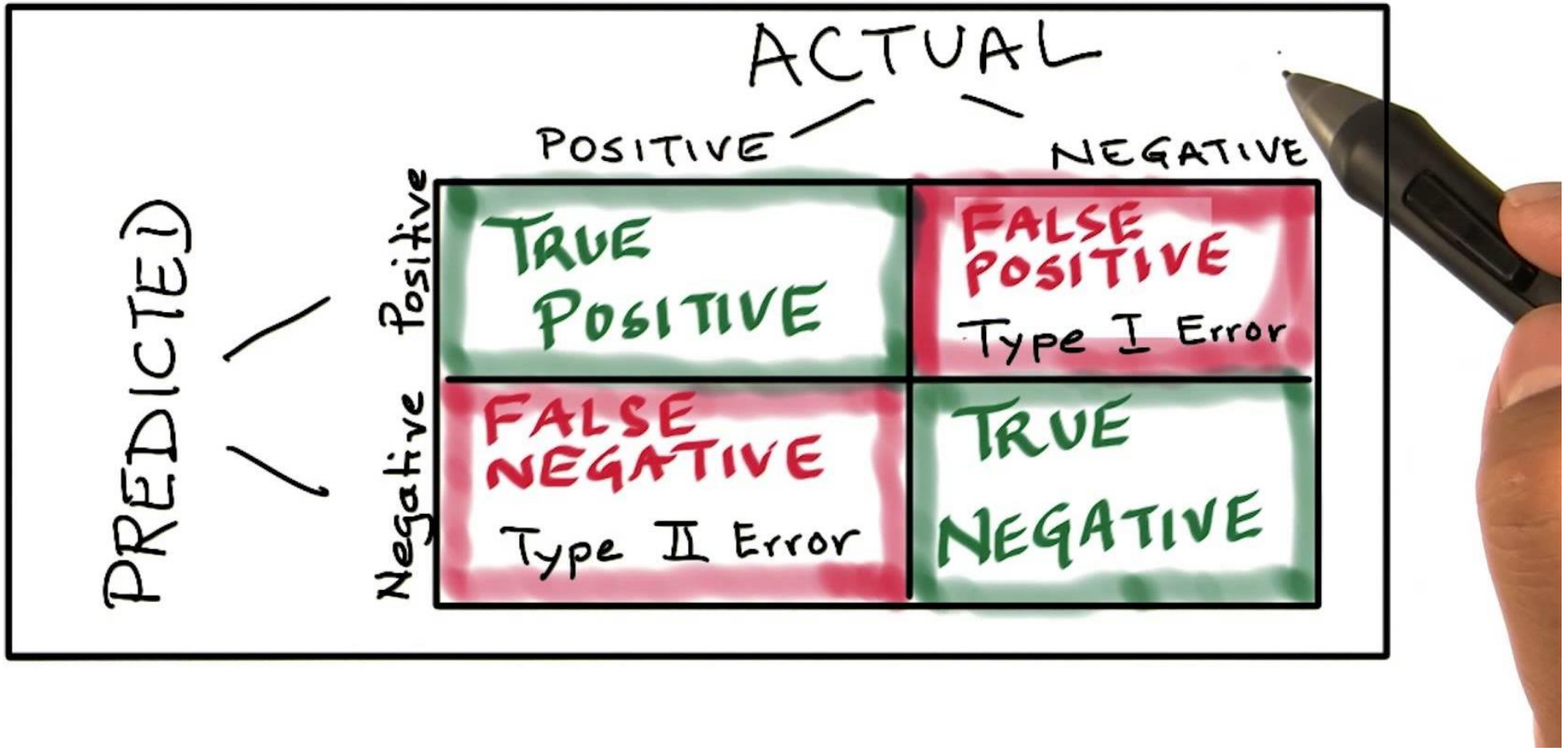
- Per-row observation weights
- Per-row offsets
- N-fold cross-validation
- Support for more distribution functions (such as Gamma, Poisson, and Tweedie)

H2O.ai

Confusion Matrix



Confusion Matrix



H2O.ai

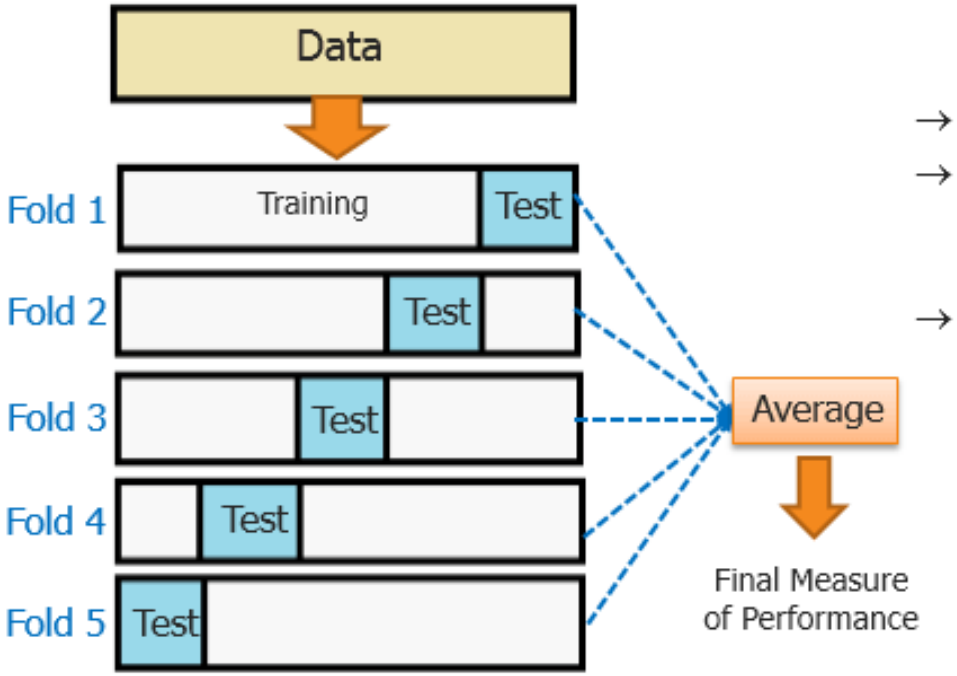
Regression Models (Grid Search)

…_03b_regression_grid_search.ipynb

Improving Model Performance (Step-by-Step)

Model Settings	MSE (CV)	MSE (Test)
GBM with default settings	N/A	0.4551
GBM with manual settings	N/A	0.4433
Manual settings + cross-validation	0.4502	0.4433
Manual + CV + early stopping	0.4429	0.4287
CV + early stopping + full grid search	0.4378	0.4196
CV + early stopping + random grid search	0.4227	0.4047
Stacking best two from random grid search	N/A	0.3997

Cross-Validation



→ Technique to validate models/classifiers

→ Method to estimate how accurately the model generalizes to unseen data i.e., how well it performs/predicts

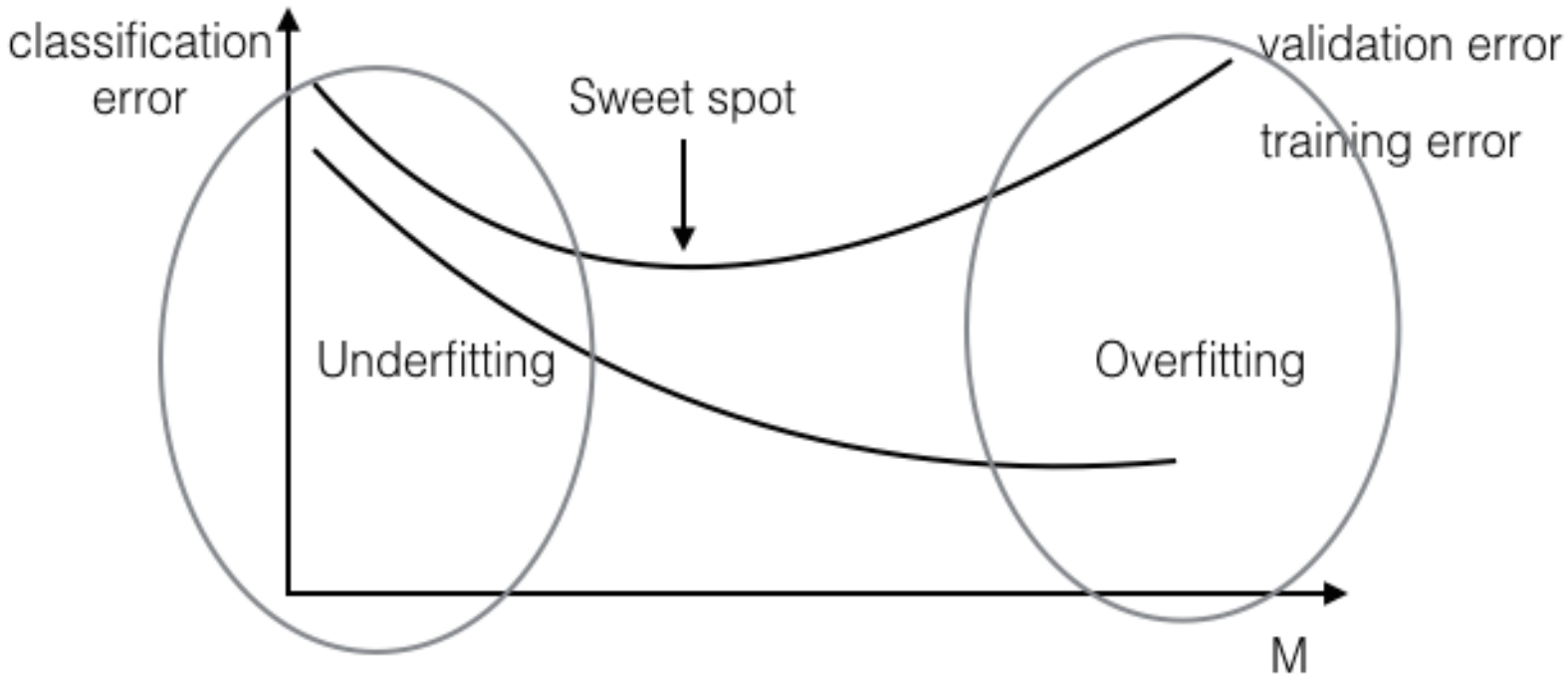
→ K-fold CV

- » Most popular
- » k is typically set to 10
- » Every sample/record is used both in training and test sets

Slide 46

www.edureka.in/data-science

Early Stopping



Regression Models (Ensembles)

…_03c_regression_ensembles.ipynb

Improving Model Performance (Step-by-Step)

Model Settings	MSE (CV)	MSE (Test)
GBM with default settings	N/A	0.4551
GBM with manual settings	N/A	0.4433
Manual settings + cross-validation	0.4502	0.4433
Manual + CV + early stopping	0.4429	0.4287
CV + early stopping + full grid search	0.4378	0.4196
CV + early stopping + random grid search	0.4227	0.4047
Stacking best two from random grid search	N/A	0.3997

Recap

Learning Objectives

- Start and connect to a local H_2O cluster from R/Python.
- Import data from R/Python data frames, local files or web.
- Perform basic data transformation and exploration.
- Train classification and regression models using H_2O machine learning algorithms.
- Evaluate models and make predictions.
- Improve performance by tuning and stacking.

Thanks!

- Slides
  - bit.ly/h2o_meetups
- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe

H2O.ai

Making Machine Learning Accessible to Everyone

Photo credit: Virgin Media