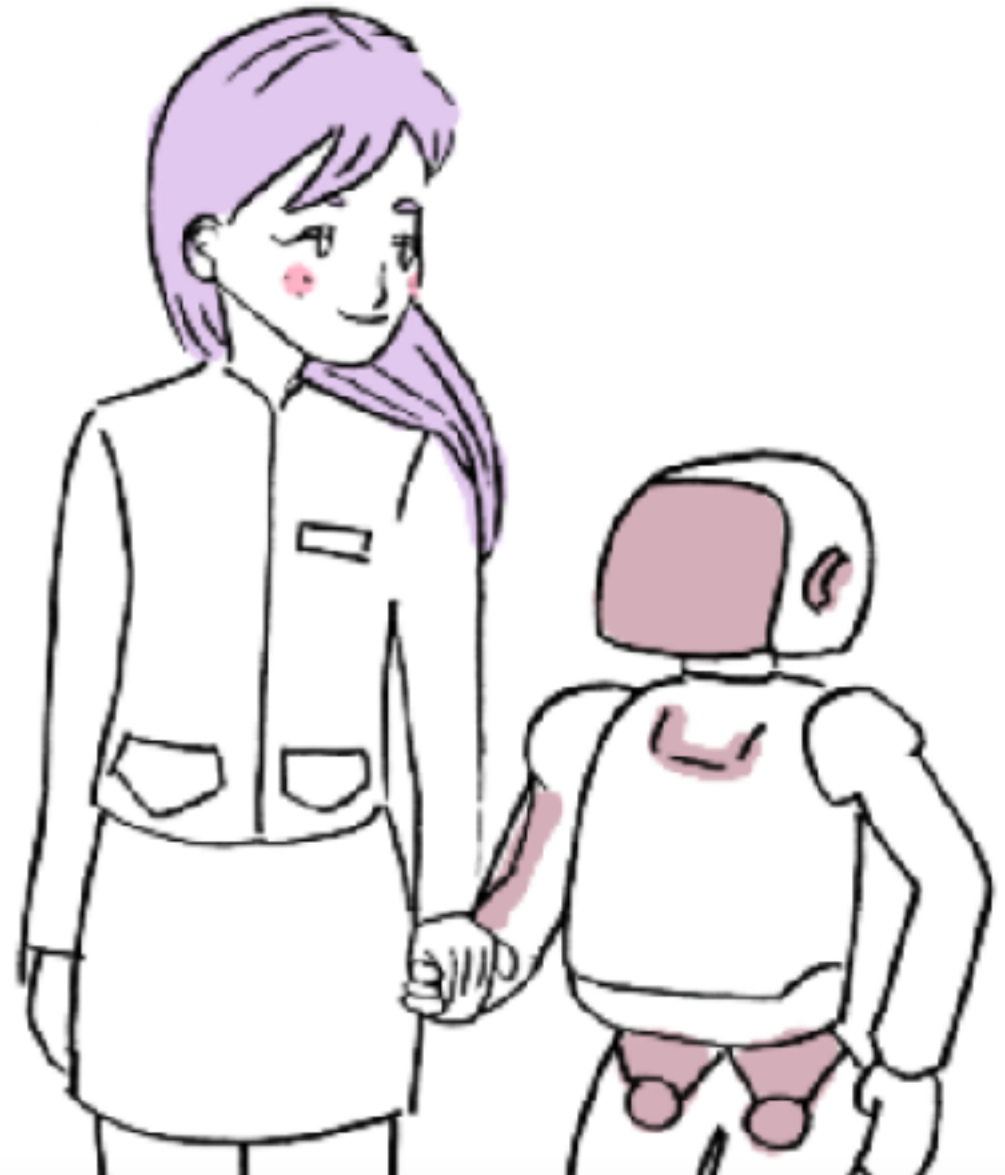
Intro to Machine Learning with H2O in R

R-Ladies Budapest Meetup
May 2017

Erin LeDell Ph.D.
@ledell

Introduction

- Chief Machine Learning Scientist at H2O.ai, in Mountain View, California, USA
- Ph.D. in Biostatistics from UC Berkeley (focus on ML)
- Co-organizer of R-Ladies San Francisco
- R-Ladies Global Leadership Team
- Founder of wimlds.org



Agenda

- Who/What is H2O?
- H2O Machine Learning Platform
- H2O in R
- H2O Tutorials

Slides ⬇️ https://tinyurl.com/rladies-erum-h2o

H2O.ai

H2O.ai, the Company

- Founded in 2012
- Stanford & Purdue Math & Systems Engineers
- Headquarters: Mountain View, California, USA

H2O, the Platform

- Open Source Software (Apache 2.0 Licensed)
- R, Python, Scala, Java and Web Interfaces
- Distributed Algorithms that Scale to Big Data

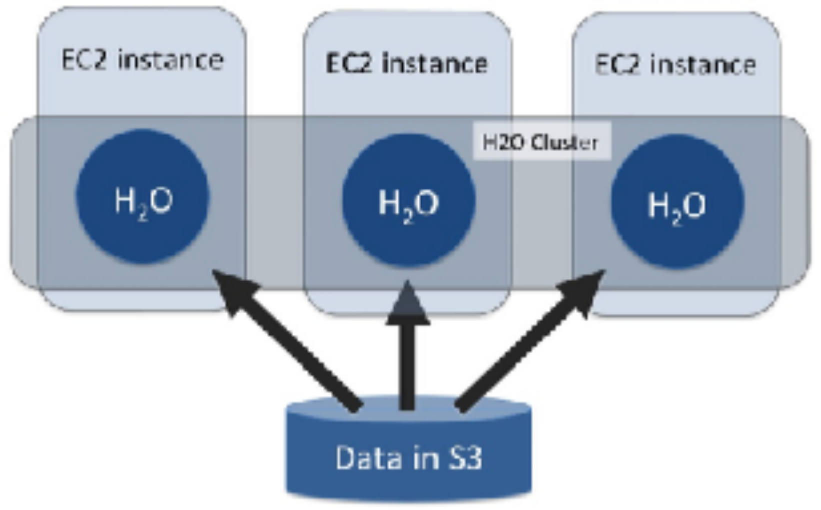
H2O Platform

H2O Platform Overview

- Distributed implementations of cutting edge ML algorithms.
- Core algorithms written in high performance Java.
- APIs available in R, Python, Scala, REST/JSON.
- Interactive Web GUI called H2O Flow.
- Easily deploy models to production with H2O Steam.

H2O Distributed Computing

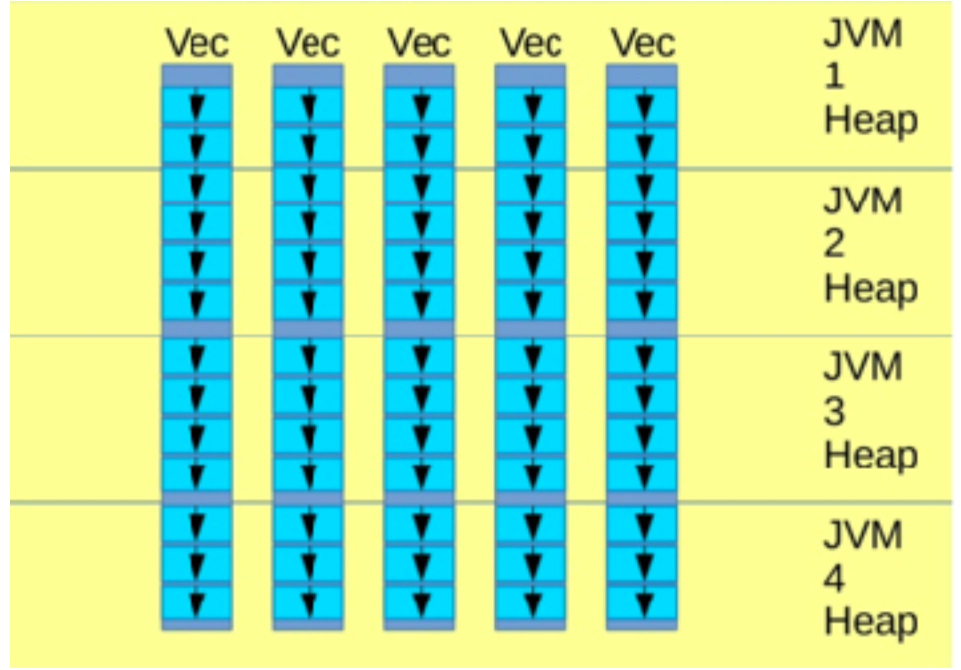
H2O Cluster



- Multi-node cluster with shared memory model.
- All computations in memory.
- Each node sees only some rows of the data.
- No limit on cluster size.

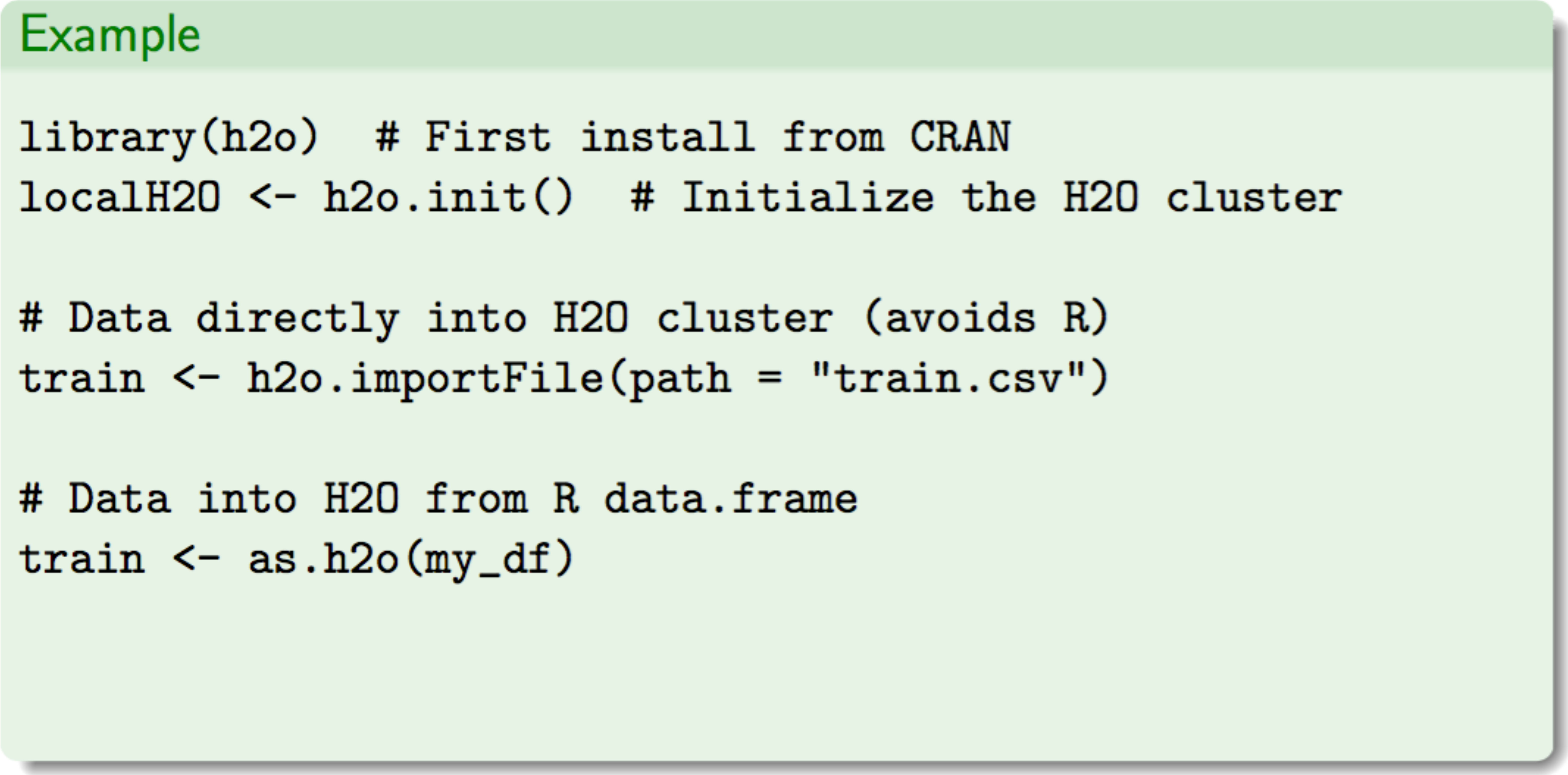
H2O Frame

- Distributed data frames (collection of vectors).
- Columns are distributed (across nodes) arrays.
- Works just like R’s data.frame or Python Pandas DataFrame



H2O in R

H2O Startup & Load Data



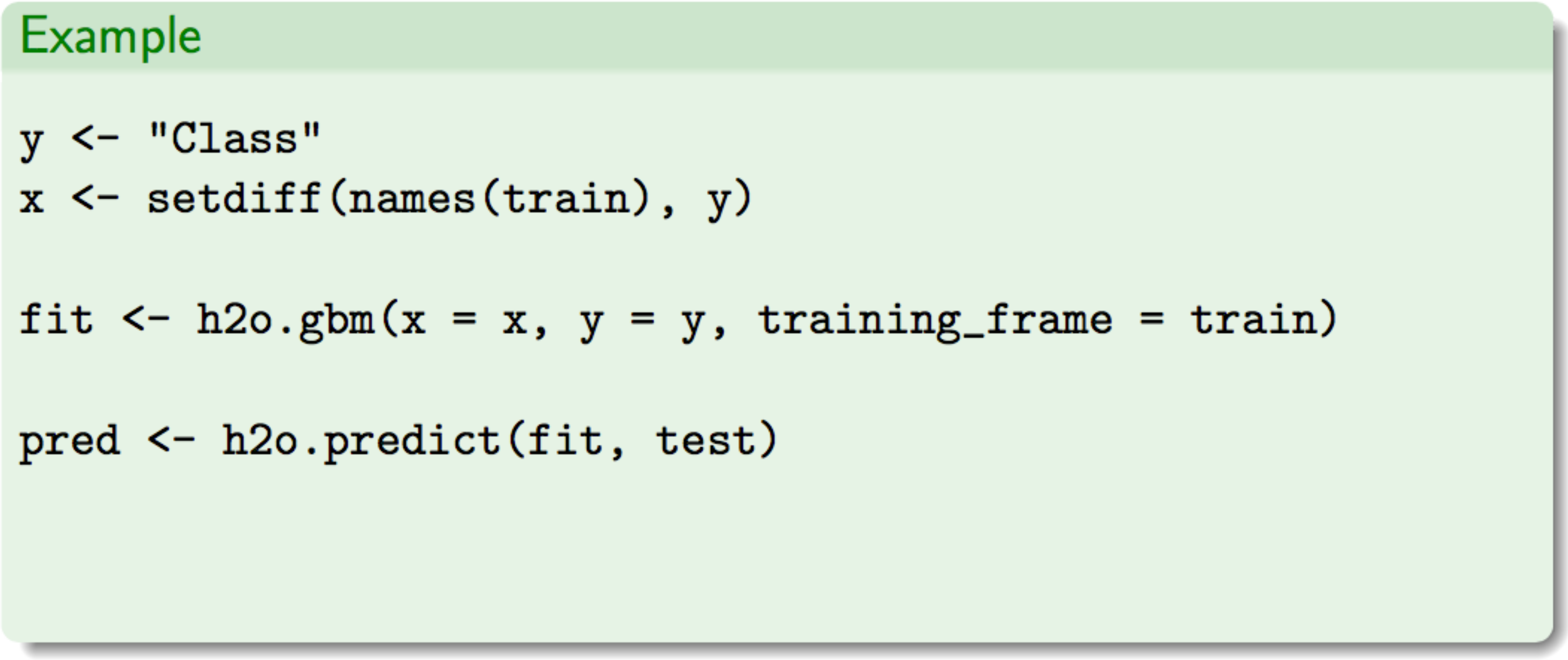
Example

```
library(h2o)  # First install from CRAN
localH2O <- h2o.init()  # Initialize the H2O cluster

# Data directly into H2O cluster (avoids R)
train <- h2o.importFile(path = "train.csv")

# Data into H2O from R data.frame
train <- as.h2o(my_df)
```

H2O Machine Learning (e.g. GBM)



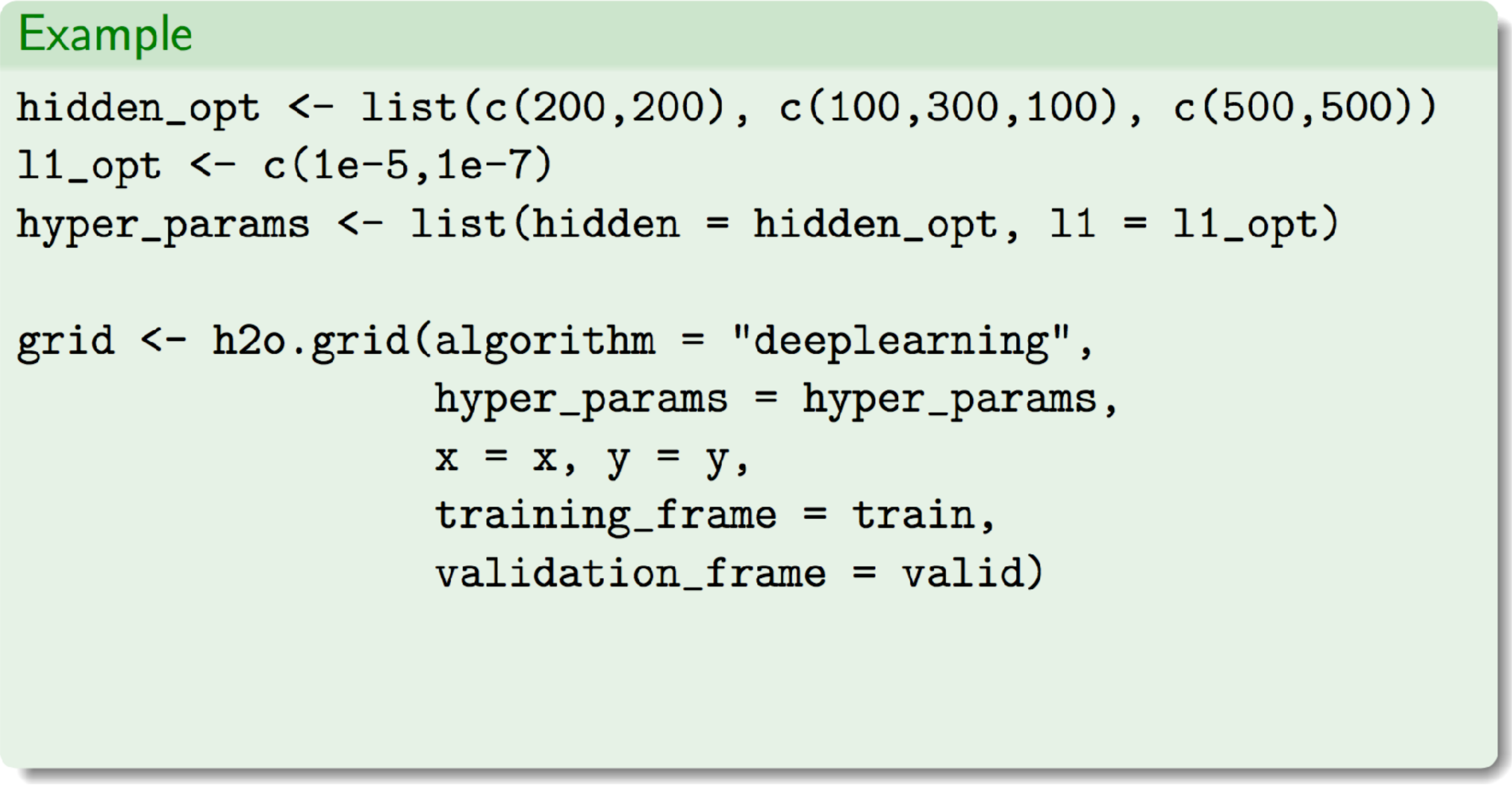
Example

```
y <- "Class"
x <- setdiff(names(train), y)

fit <- h2o.gbm(x = x, y = y, training_frame = train)

pred <- h2o.predict(fit, test)
```

H2O Cartesian Grid Search

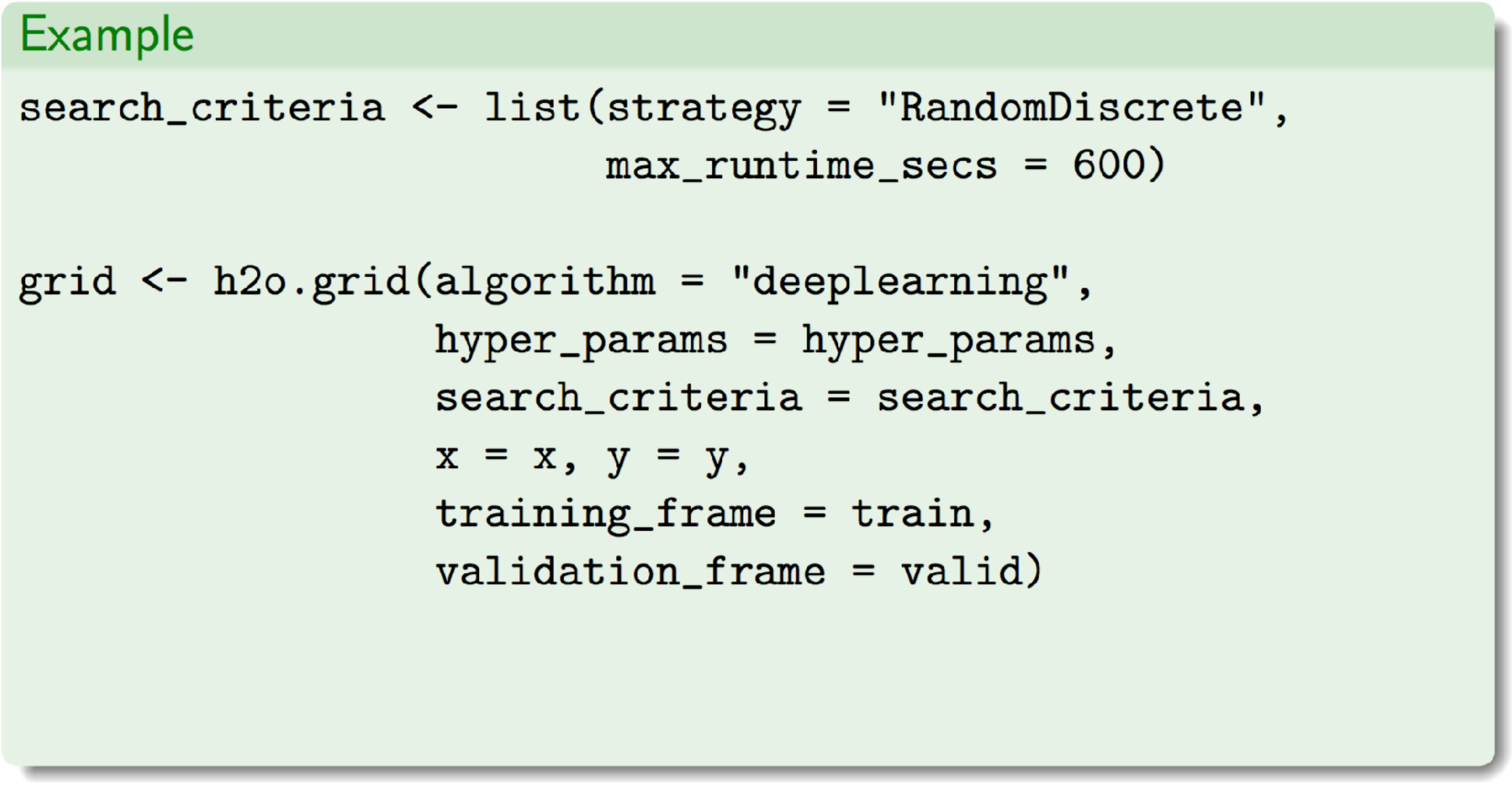


Example

```
hidden_opt <- list(c(200,200), c(100,300,100), c(500,500))
l1_opt <- c(1e-5,1e-7)
hyper_params <- list(hidden = hidden_opt, l1 = l1_opt)

grid <- h2o.grid(algorithm = "deeplearning",
                 hyper_params = hyper_params,
                 x = x, y = y,
                 training_frame = train,
                 validation_frame = valid)
```

H2O Random Grid Search

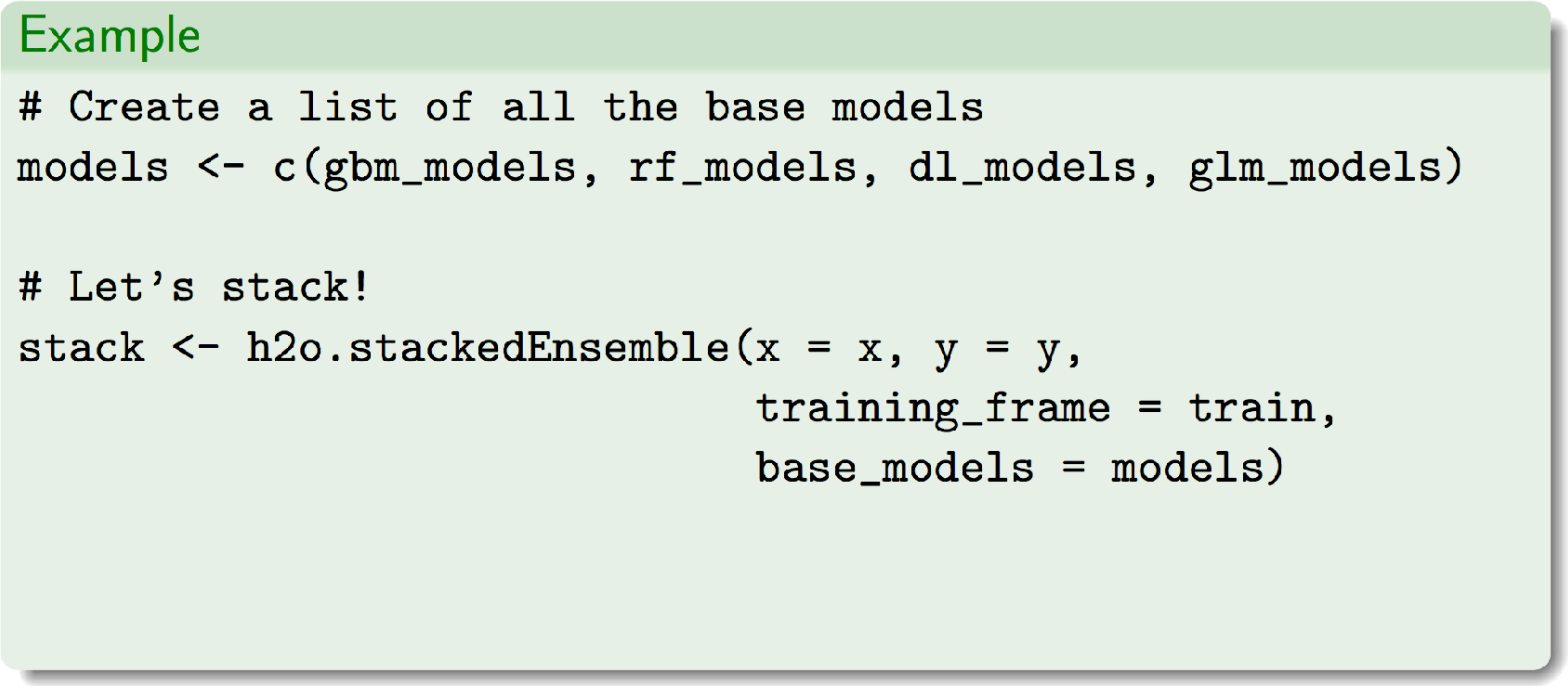


Example

```
search_criteria <- list(strategy = "RandomDiscrete",
                        max_runtime_secs = 600)

grid <- h2o.grid(algorithm = "deeplearning",
                 hyper_params = hyper_params,
                 search_criteria = search_criteria,
                 x = x, y = y,
                 training_frame = train,
                 validation_frame = valid)
```

Stacked Ensembles



Example

```
# Create a list of all the base models
models <- c(gbm_models, rf_models, dl_models, glm_models)

# Let's stack!
stack <- h2o.stackedEnsemble(x = x, y = y,
                             training_frame = train,
                             base_models = models)
```

H2O AutoML

Example

```r
library(h2o)
h2o.init()

train <- h2o.importFile("train.csv")


aml <- h2o.automl(y = "response_colname",
                  training_frame = train,
                  max_runtime_secs = 600)

lb <- aml@leaderboard
```

H2O R Tutorials

https://github.com/h2oai/h2o-tutorials

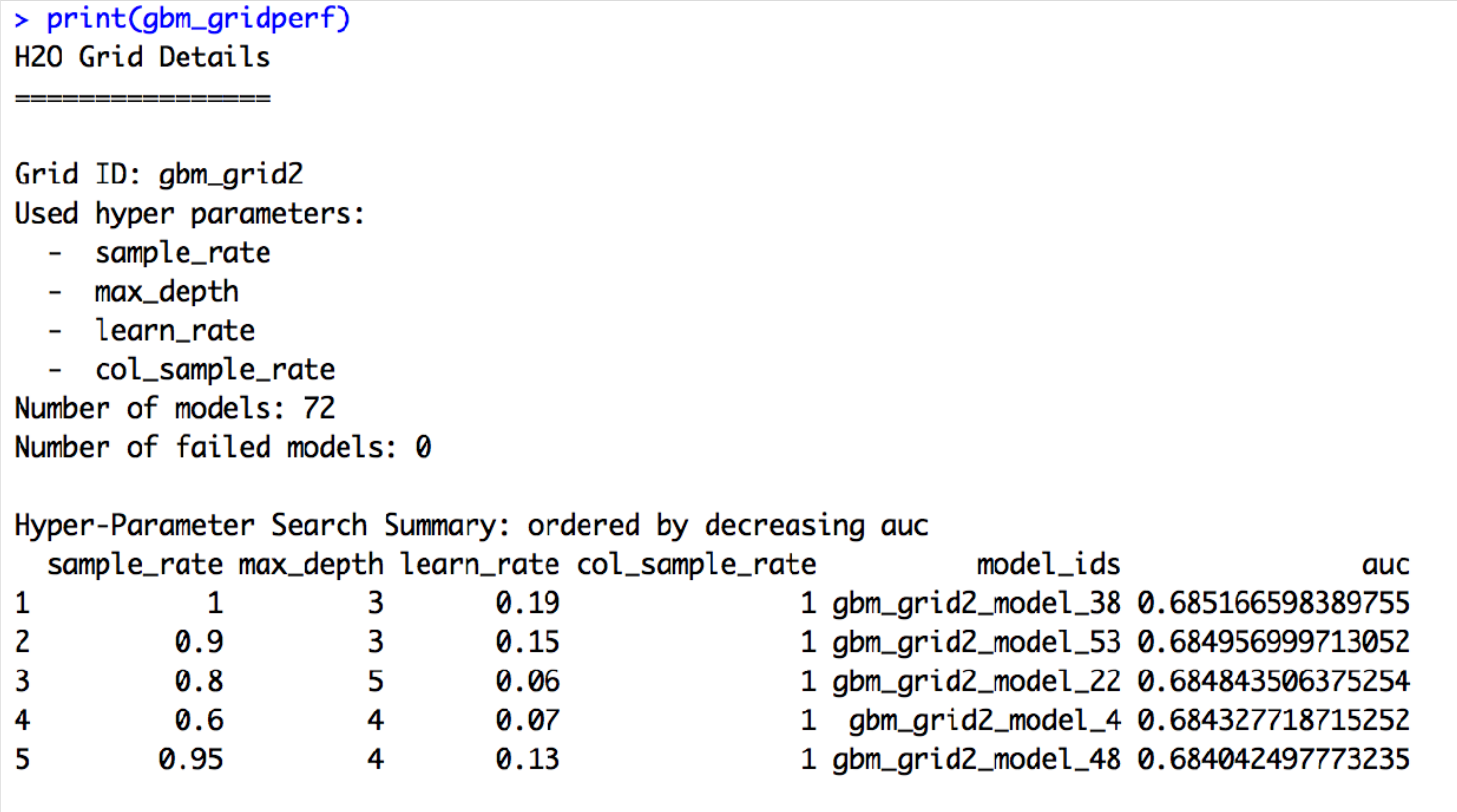
R Tutorial: Intro to H2O Algorithms

The “Intro to H2O” tutorial introduces five popular supervised machine learning algorithms in the context of a binary classification problem.

The training module demonstrates how to train models and evaluate model performance on a test set.

- Generalized Linear Model (GLM)
- Random Forest (RF)
- Gradient Boosting Machine (GBM)
- Deep Learning (DL)
- Naive Bayes (NB)

R Tutorial: Grid Search for Model Selection



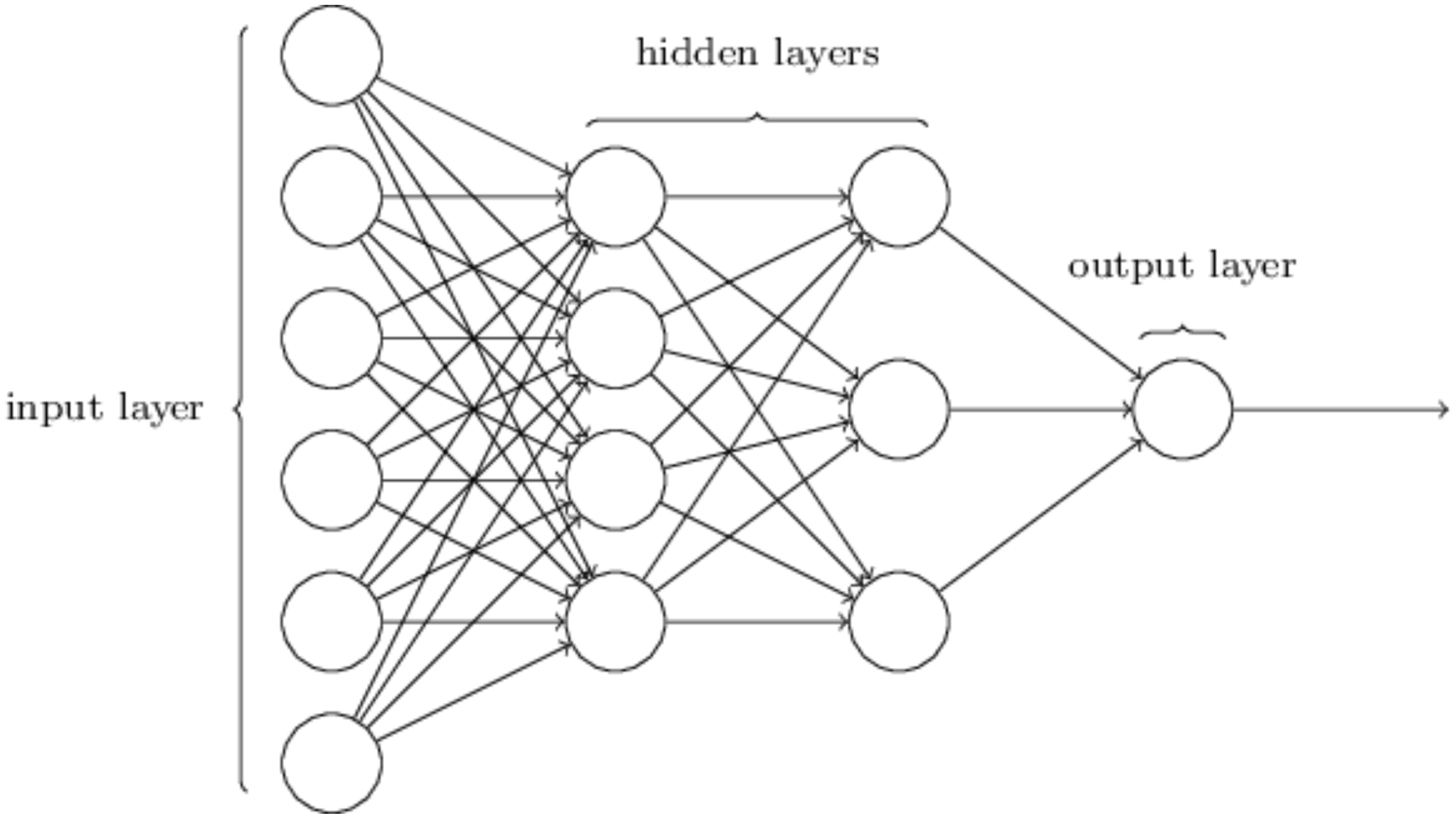
```
> print(gbm_gridperf)
H2O Grid Details
================

Grid ID: gbm_grid2
Used hyper parameters:
  -  sample_rate
  -  max_depth
  -  learn_rate
  -  col_sample_rate
Number of models: 72
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by decreasing auc
  sample_rate max_depth learn_rate col_sample_rate          model_ids               auc
1           1         3       0.19               1 gbm_grid2_model_38 0.685166598389755
2         0.9         3       0.15               1 gbm_grid2_model_53 0.684956999713052
3         0.8         5       0.06               1 gbm_grid2_model_22 0.684843506375254
4         0.6         4       0.07               1  gbm_grid2_model_4 0.684327718715252
5        0.95         4       0.13               1 gbm_grid2_model_48 0.684042497773235
```

The second training module demonstrates how to find the best set of model parameters for each model using Grid Search.

R Tutorial: Deep Learning



The "Deep Learning in R" tutorial gives an overview of how to train H2O deep neural networks in R.

- Deep Learning via Multilayer Perceptrons (MLPs)
  - Early Stopping
  - Random Grid Search
- Deep Learning Autoencoders
  - Unsupervised Pretraining
  - Deep Features
  - Anomaly Detection

H2O Resources

- Documentation: http://docs.h2o.ai
- Tutorials: https://github.com/h2oai/h2o-tutorials
- Slidedecks: https://github.com/h2oai/h2o-meetups
- Videos: https://www.youtube.com/user/0xdata
- Stack Overflow: https://stackoverflow.com/tags/h2o
- Google Group: https://tinyurl.com/h2ostream
- Gitter: http://gitter.im/h2oai/h2o-3
- Events & Meetups: http://h2o.ai/events

Contribute to H2O!

open source™

Get in touch over email, Gitter or JIRA.

https://tinyurl.com/h2o-contribute

Thank you!

@ledell on Github, Twitter

erin@rladies.org