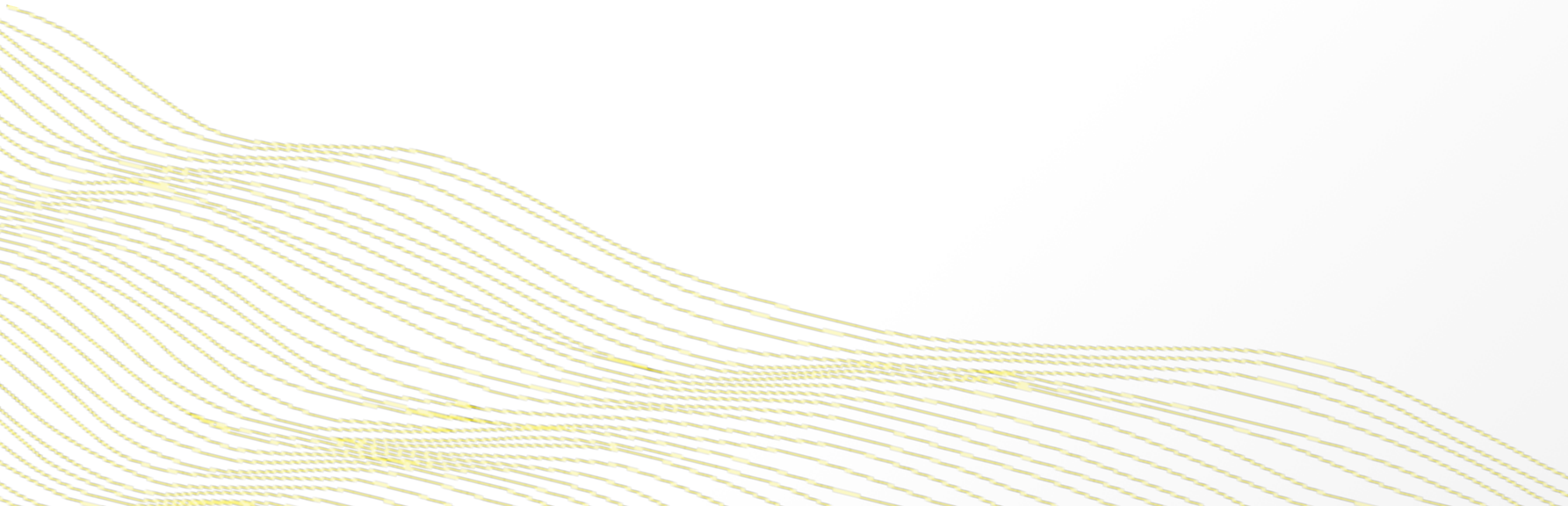
Hands on Introduction to H2O



Agenda

Today's Talk

H2O Installation

- Installation in Python
- Installation in R

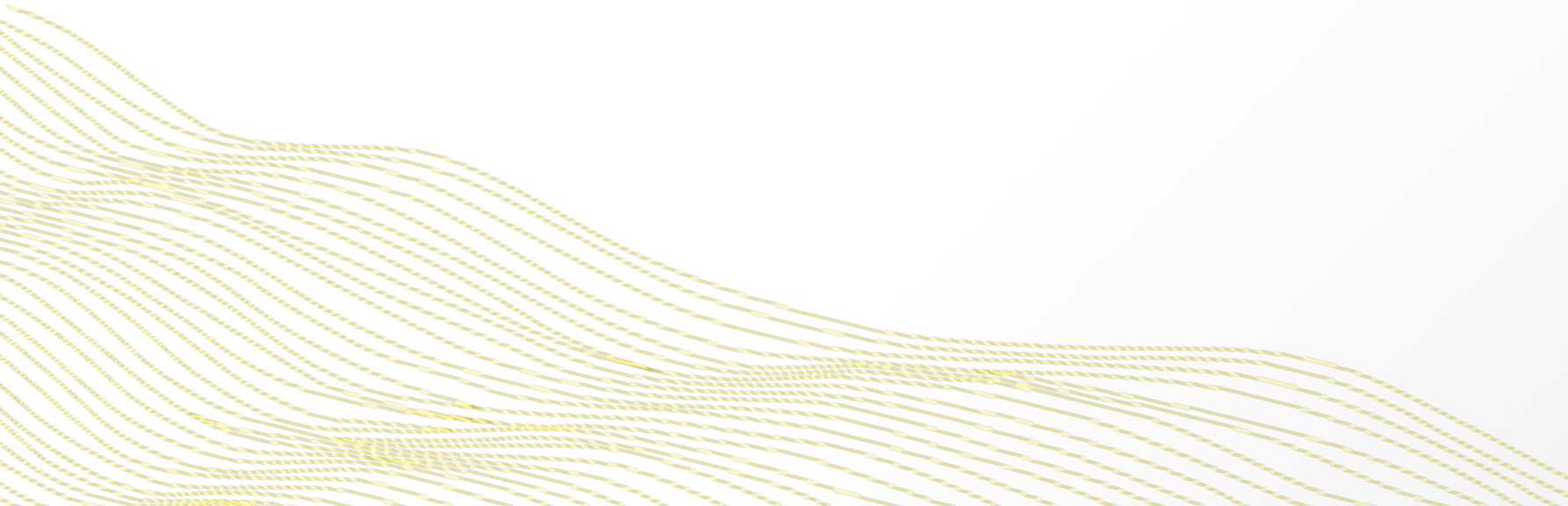
Introduction to H2O

- H2O in Python
- H2O in R

Hands on Demo in R and Python

- Our Use Case
- Importing data into H2O
- Data Cleaning and Feature Engineering
- Model Building
- Using Flow with R and Python

H2O Installation



H2O Prerequisites

To Launch H2O and Flow the only prerequisite is:

- 64 bit Java 6+

Installing H2O in Python

Prerequisite: Python 2.7+

Install Dependencies

```
pip install requests
pip install tabulate
pip install scikit-learn
```

Install H2O

```
# The following command removes the H2O module for Python.
pip uninstall h2o


# Next, use pip to install this version of the H2O Python module.
pip install http://h2o-release.s3.amazonaws.com/h2o/rel-ueno/2/Python/h2o-3.10.4.2-py2.py3-none-any.whl
```

Installing H2O in R

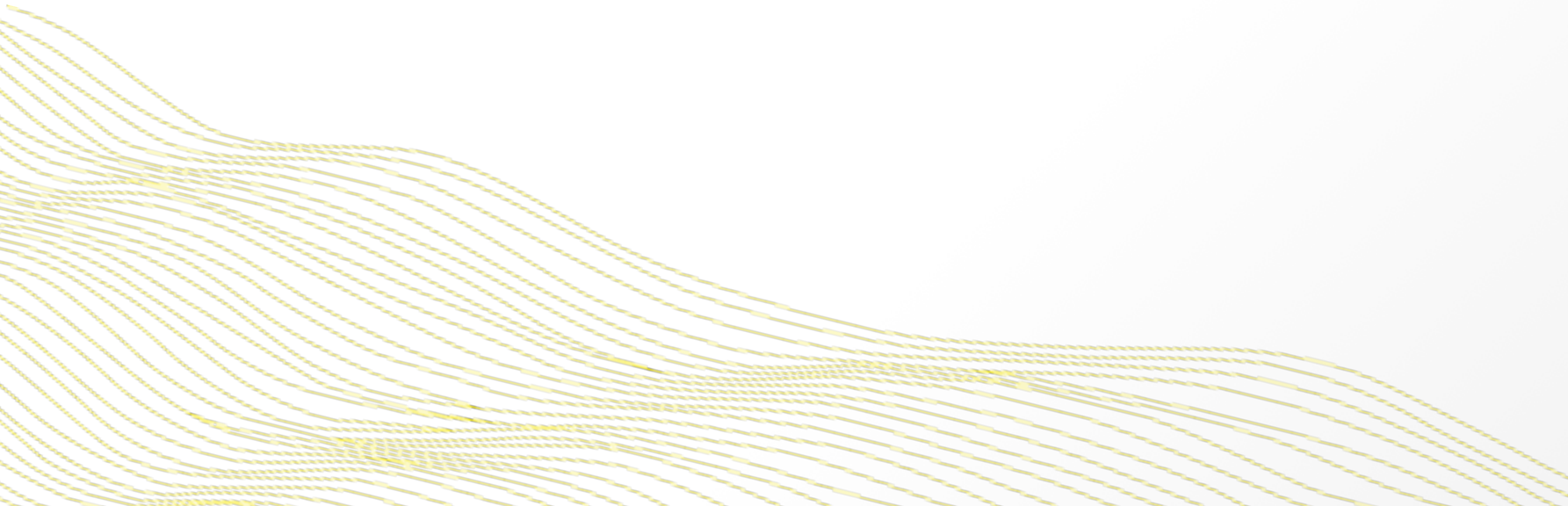
Prerequisite: R(>= 2.13.0)

```
# The following two commands remove any previously installed H2O packages for R.
if ("package:h2o" %in% search()) { detach("package:h2o", unload=TRUE) }
if ("h2o" %in% rownames(installed.packages())) { remove.packages("h2o") }

# Next, we download packages that H2O depends on.
if (! ("methods" %in% rownames(installed.packages()))) { install.packages("methods") }
if (! ("statmod" %in% rownames(installed.packages()))) { install.packages("statmod") }
if (! ("stats" %in% rownames(installed.packages()))) { install.packages("stats") }
if (! ("graphics" %in% rownames(installed.packages()))) { install.packages("graphics") }
if (! ("RCurl" %in% rownames(installed.packages()))) { install.packages("RCurl") }
if (! ("jsonlite" %in% rownames(installed.packages()))) { install.packages("jsonlite") }
if (! ("tools" %in% rownames(installed.packages()))) { install.packages("tools") }
if (! ("utils" %in% rownames(installed.packages()))) { install.packages("utils") }

# Now we download, install and initialize the H2O package for R.
install.packages("h2o", type="source", repos=(c("http://h2o-release.s3.amazonaws.com/h2o/rel-ueno/2/R")))
```

Introduction to H2O



Reading Data into H2O with Python

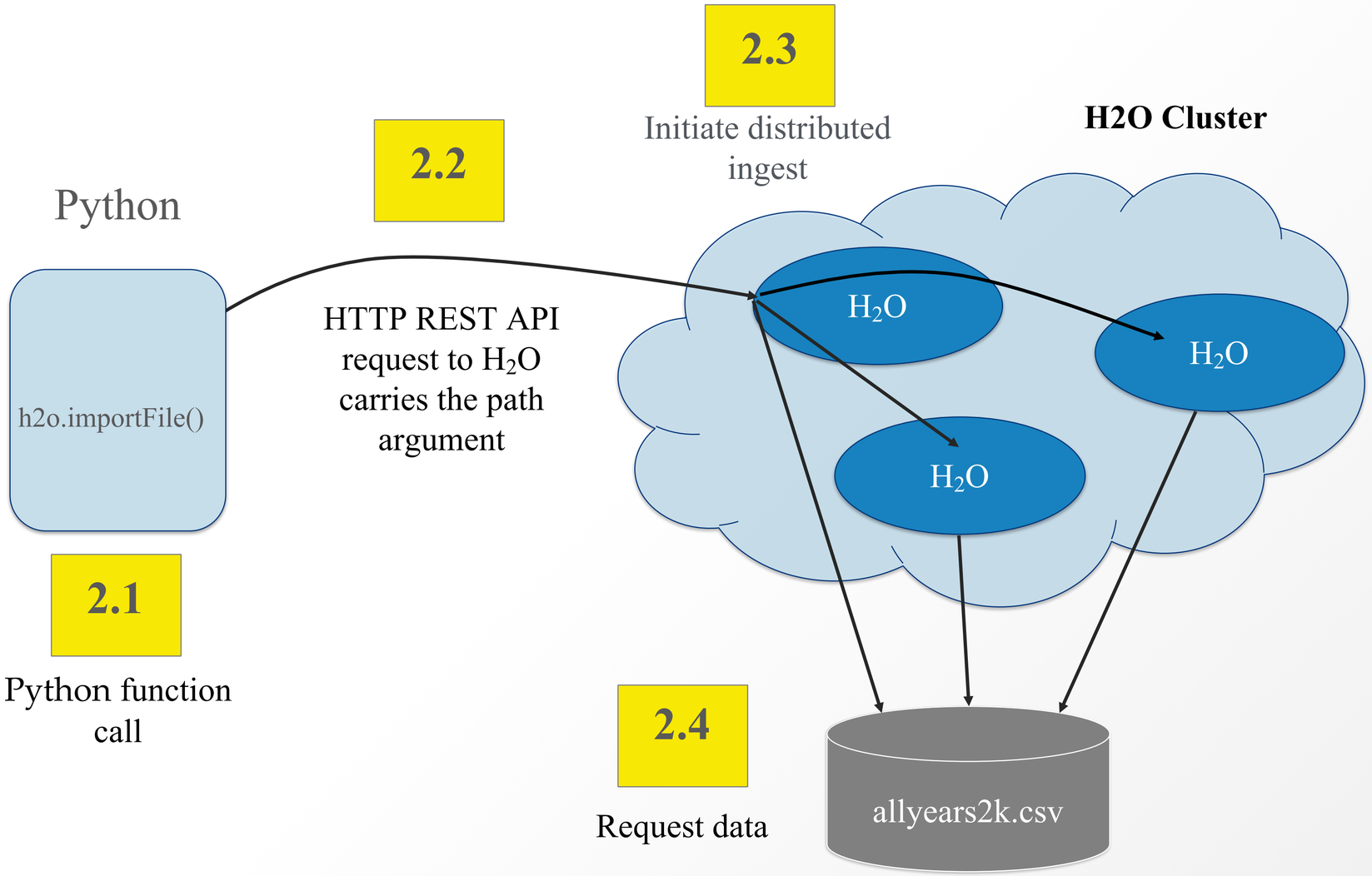
STEP 1

Python User

```
h2o_df = h2o.import_file("../data/allyears2k.csv")
```

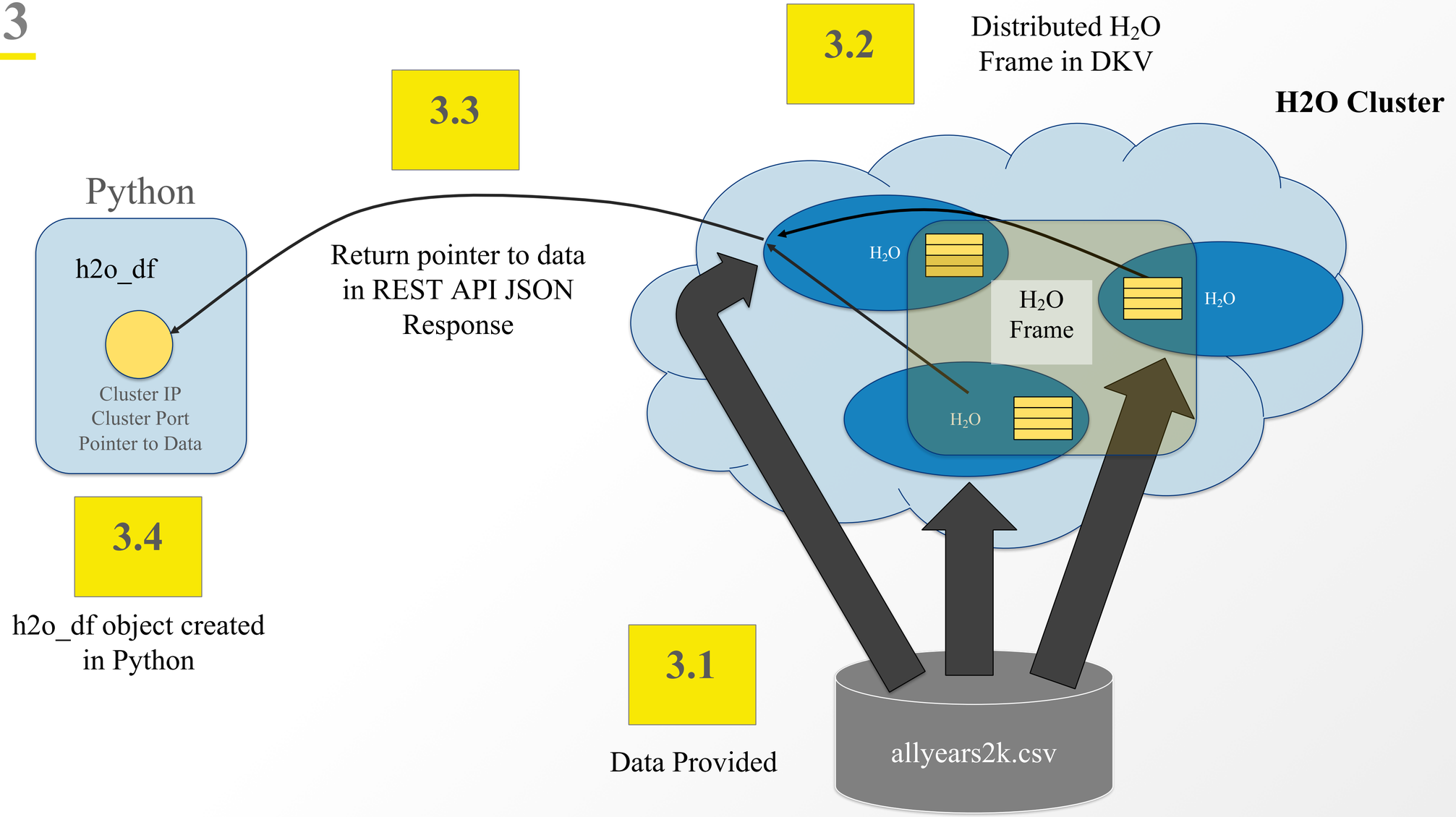
Reading Data into H2O with Python

STEP 2



Reading Data into H2O with Python

STEP 3



H2O.ai

Supported Python Functions

	Standard Python (Using module pandas and scikit-learn)	Python on H2O
Reading in data	pandas.read_csv(data_path)	h2o.import_file(data_path)
Summarizing data	pandas_frame.describe()	h2o_frame.describe()
Combining rows or columns	pandas.concat(list[frame1,frame2],axis = 1) pandas.concat(list[frame1,frame2])	h2o_frame.cbind(h2o_frame2) h2o_frame.rbind(h2o_frame2)
Unary or Binary Operations	+,-,*,/,^,%%, %/%,etc	+,-,*,/,^,%%, %/%,etc
Building Random Forest Model	model = RandomForestClassifier(n_estimators = 100) model = model.fit(x_frame, y_frame)	model = H2ORandomForestClassifier(n_trees = 100) model = model.train(x, y, train_frame)
Predict with Model	model.predict	model.predict
Obtaining Metrics	metrics.auc	metrics = model.model_performance(frame) metrics.auc()
Unsupported Functions	sklearn.neighbors	UNSUPPORTED

Supported R Functions

	Standard R	R on H2O
Reading in data	read.csv, read.table	h2o.importFile(data_path)
Summarizing data	summary	h2o.summary
Combining rows or columns	cbind, rbind	h2o.cbind, h2o.rbind
Unary or Binary Operations	+,-,*,/,^,%%, %/%,etc	+,-,*,/,^,%%, %/%,etc
Building Random Forest Model	randomForest(ntree = 100)	H2o.randomForest(n_trees = 100)
Predict with Model	predict	h2o.predict
Obtaining Metrics	auc	h2o.auc
Unsupported Functions	smooth	UNSUPPORTED

Demo

Use Case

End Goal

Build a model that can predict whether a loan will default

Why?

We can use this to identify which loans should be approved.

How?

Use information about the loan and loan applicant to train a model that will predict whether the loan ended up defaulting.

The Data

Columns	Description	Units
loan_amnt	Requested loan amount	Dollars
term	Loan term length	Months
int_rate	Interest rate of the loan	%
emp_length	Borrower's length of employment	Years
home_ownership	Borrower's home ownership	Categorical
annual_inc	Borrower's annual income	Dollars
purpose	Purpose of the loan	Categorical
addr_state	State of residence	Categorical
dti	Debt to income ratio	%
deling_years	Number of delinquencies in the past 2 years	Integer
issue_d	The month which the loan was funded	Date
loan_status	The status of the loan – can be current	Categorical

Resources

- Data: http://h2o-public-test-data.s3.amazonaws.com/bigdata/laptop/lending-club/LoanStats3a.csv
- R Script: lending_club_solutions.R
- Python Script: lending_club_solutions.ipynb

Questions?