Automatic and Interpretable Machine Learning in R with H_2O and LIME

#ODSC

Jo-fai (Joe) Chow

Data Science Evangelist / Community Manager

joe@h2o.ai

@matlabulous

Download → https://bit.ly/

odsc2018_h2o

About Me

London 12th Sep

Edinburgh 18th Sep

Paris 19th Sep

- **Before H_2O**
  - Water Engineer / EngD Researcher / Matlab Fan Boy
    (wonder why 🐦 @matlabulous?)
  - Discovered R, Python, H_2O … never look back again
  - Data Scientist at Virgin Media (UK), Domino Data Lab (US)
- **At H_2O …**
  - Data Scientist / Evangelist /
  - Sales Engineer / Solution Architect /
  - Community Manager

  … The harsh reality of startup life …

Reminder: #360Selfie

Agenda

Time	Topics / Tasks
9:30 – 9:35 am	Install **R packages** from CRAN slides/code: bit.ly/**odsc2018_h2o**
9:35 – 9:50 am	Introduction (H_2O, AutoML, LIME)
9:50 – 10:15 am	Worked Examples
10:15 – 10:25 pm	Real-World Use Case: Moneyball
10:25 – 10:30 am	Other H_2O News + Q & A

#ODSC

Why?

- Most users/organizations can benefit from **automatic machine learning pipelines**.
  - Eliminate time wasted on repetitive tasks, human errors, debugging etc.
- GDPR mandates a “right to explanation” from machine learning models.
  - **model interpretations** are crucial for those who must explain their models to regulators or customers.

You will learn …

- How to build high quality **H_2O** models (almost) automatically.
- How to explain predictions from complex H_2O models with **LIME**.
- **Bonus**: A real use case that led to a **multimillion-dollar** baseball decision earlier this year.

H2O.ai

H_2O AutoML + LIME for **REAL** !!!

led to the signing of a
Major League Baseball (MLB) player

$20M

multi-year contract

finalised two weeks
before the regular season

#ODSC

Time	Topics / Tasks
9:30 – 9:35 pm	Install **R packages** from CRAN slides/code: bit.ly/**odsc2018_h2o**

LIME

Reference: https://github.com/thomasp85/lime

```
# Install 'lime' from CRAN
install.packages('lime')
```

mlbench for datasets

```
install.packages('mlbench')
```

H2O

Reference: https://www.h2o.ai/download/

```
# Install 'h2o' from CRAN
install.packages('h2o')
```

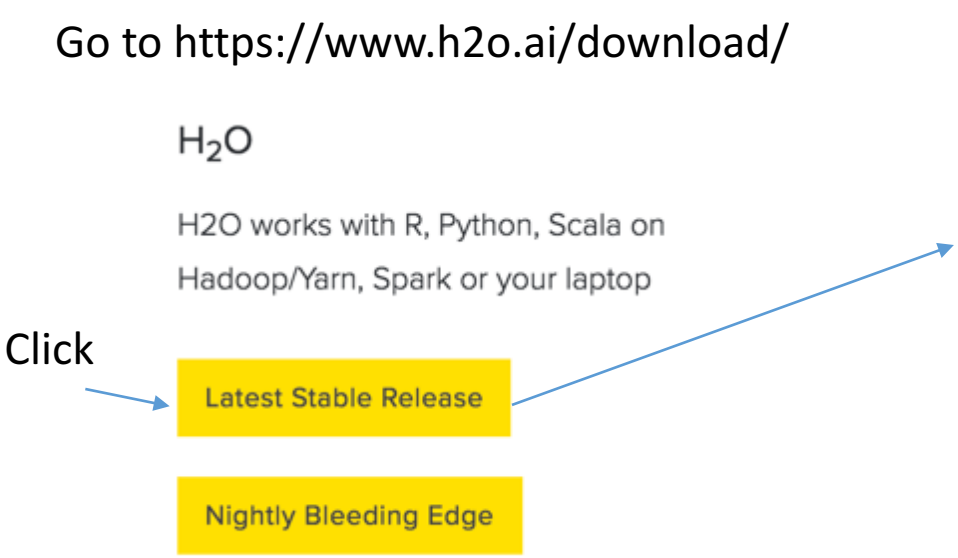
rmarkdown for rendering HTML

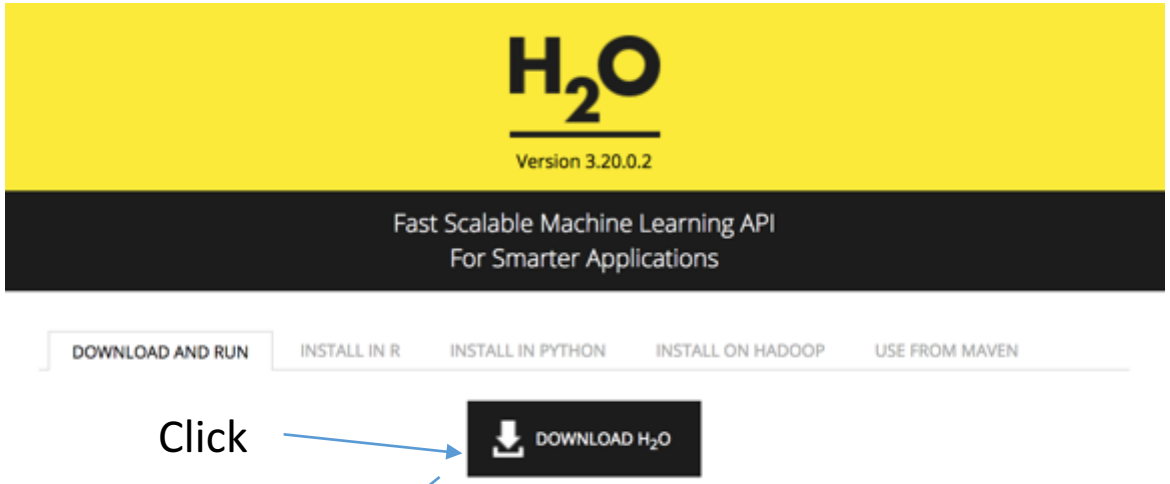
```
install.packages('rmarkdown')
```


H2O.ai

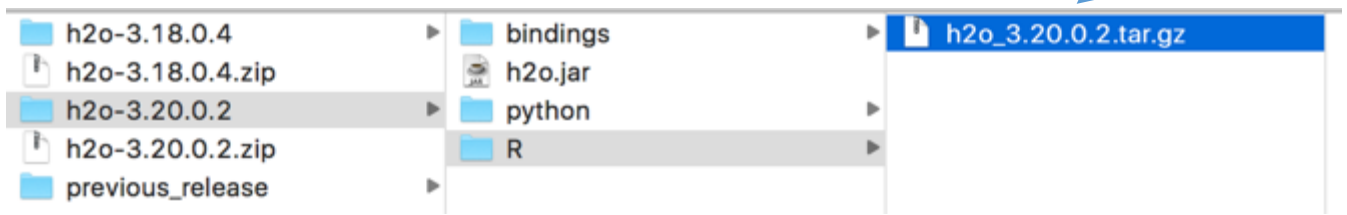
Install H_2O using tar.gz (if needed)

Go to https://www.h2o.ai/download/

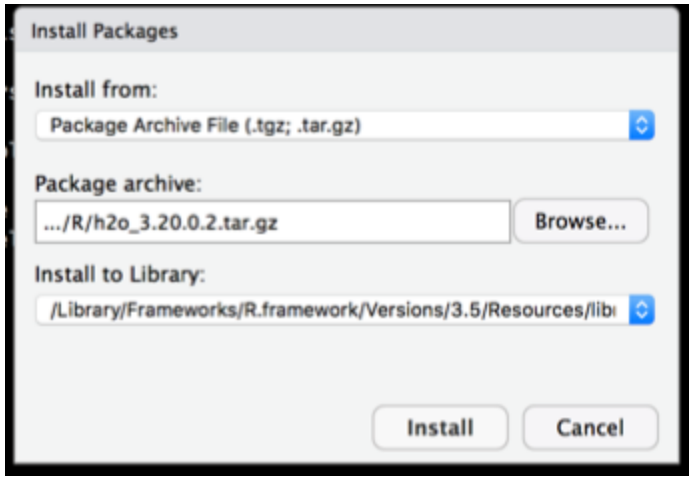




Unzip



Install



H2O.ai

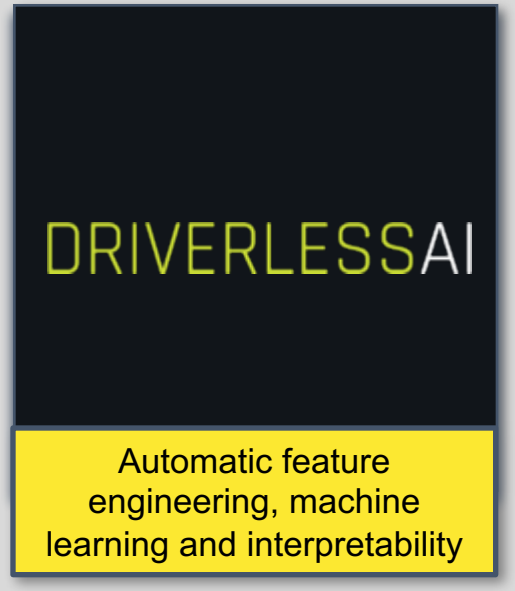
H2O.ai Overview

Company	Founded in Silicon Valley in 2012 Funded: $75M Investors: Wells Fargo, NVIDIA, Nexus Ventures, Paxion Ventures
Products	• H2O Open Source Machine Learning (14,000 organizations) • H2O Driverless AI – Automatic Machine Learning
Leadership	Leader in Gartner MQ Machine Learning and Data Science Platform
Team	120 AI experts (Kaggle Grandmasters, Distributed Computing, Visualization)
Global	Mountain View, London, Prague, India

H2O.ai Product Suite

H2O	Spark + H2O SPARKLING WATER	H2O4GPU
In-Memory, Distributed Machine Learning Algorithms with H2O Flow GUI	H2O AI Open Source Engine Integration with Spark	Lightning Fast machine learning on GPUs

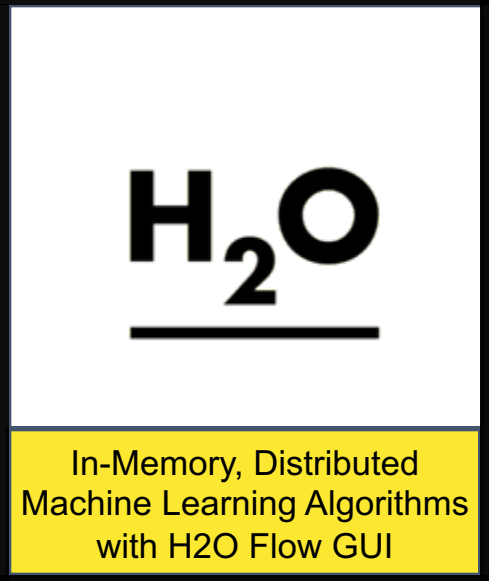
- 100% open source – Apache V2 licensed
- Built for data scientists – interface using R, Python on H2O Flow (interactive notebook interface)
- Enterprise Support subscriptions



Automatic feature engineering, machine learning and interpretability

- Enterprise software
- Built for domain users, analysts & data scientists – GUI based interface for end-to-end data science
- Fully automated machine learning from ingest to deployment
- User licenses on a per seat basis (annual subscription)

H2O.ai Product Suite



In-Memory, Distributed Machine Learning Algorithms with H2O Flow GUI

H2O AI Open Source Engine Integration with Spark

machine learning on GPUs

DRIVERLESS AI

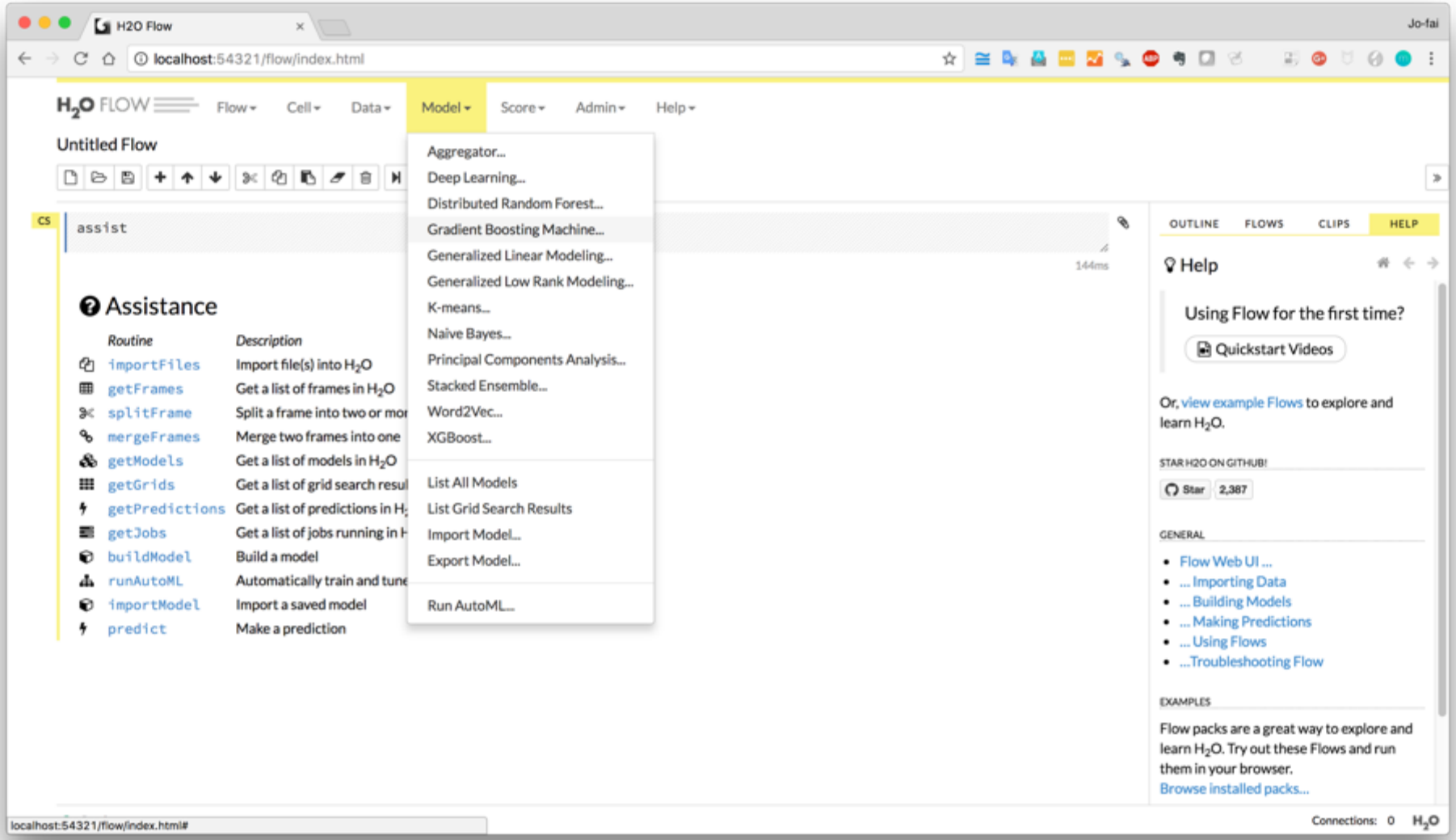
Automatic feature engineering, machine learning and interpretability

This Workshop: Open Source H_2O Core

- 100% open source – Apache V2 licensed
- Built for data scientists – interface using R, Python on H2O Flow (interactive notebook interface)
- Enterprise Support subscriptions

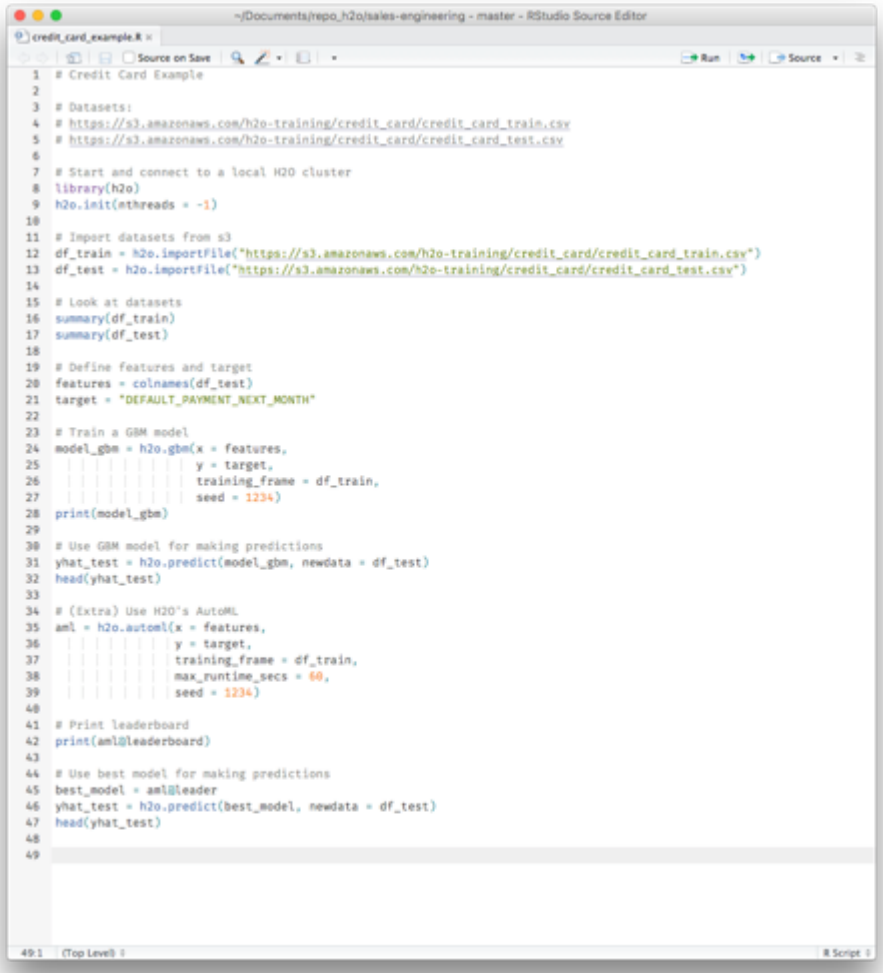
- Enterprise software
- Built for domain users, analysts & data scientists – GUI based interface for end-to-end data science
- Fully automated machine learning from ingest to deployment
- User licenses on a per seat basis (annual subscription)

H_2O Flow (Web)

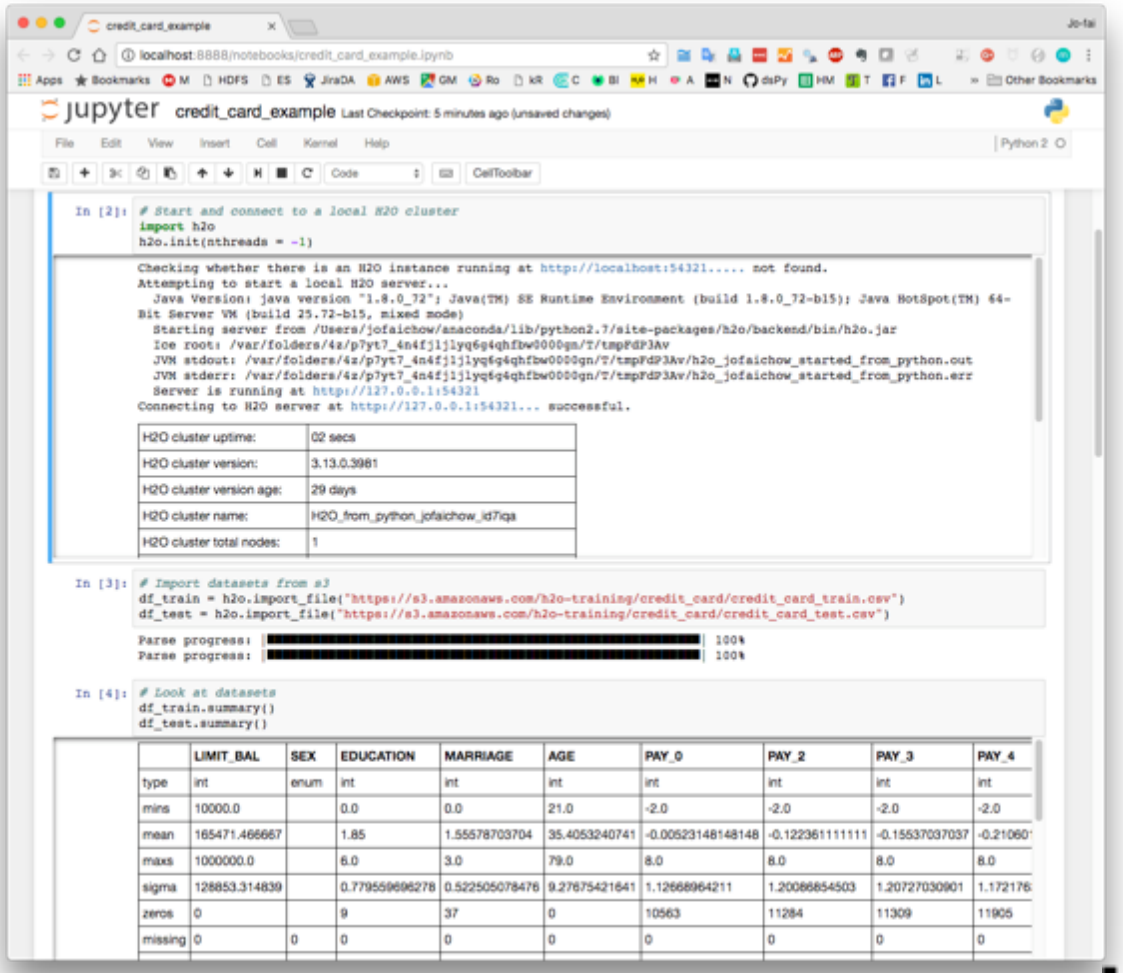


H2O.ai

Using H_2O with R and Python



```r
# Credit Card Example

# Datasets:
# https://s3.amazonaws.com/h2o-training/credit_card/credit_card_train.csv
# https://s3.amazonaws.com/h2o-training/credit_card/credit_card_test.csv

# Start and connect to a local H2O cluster
library(h2o)
h2o.init(nthreads = -1)

# Import datasets from s3
df_train = h2o.importFile("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_train.csv")
df_test = h2o.importFile("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_test.csv")

# Look at datasets
summary(df_train)
summary(df_test)

# Define features and target
features = colnames(df_test)
target = "DEFAULT_PAYMENT_NEXT_MONTH"

# Train a GBM model
model_gbm = h2o.gbm(x = features,
                    y = target,
                    training_frame = df_train,
                    seed = 1234)
print(model_gbm)

# Use GBM model for making predictions
yhat_test = h2o.predict(model_gbm, newdata = df_test)
head(yhat_test)

# (Extra) Use H2O's AutoML
aml = h2o.automl(x = features,
                 y = target,
                 training_frame = df_train,
                 max_runtime_secs = 60,
                 seed = 1234)

# Print leaderboard
print(aml@leaderboard)

# Use best model for making predictions
best_model = aml@leader
yhat_test = h2o.predict(best_model, newdata = df_test)
head(yhat_test)
```



```
In [2]: # 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_72"; Java(TM) SE Runtime Environment (build 1.8.0_72-b15); Java HotSpot(TM) 64-Bit Server VM (build 25.72-b15, mixed mode)
  Starting server from /Users/jofaichow/anaconda/lib/python2.7/site-packages/h2o/backend/bin/h2o.jar
  Ice root: /var/folders/4z/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T/tmpFdP3Av
  JVM stdout: /var/folders/4z/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T/tmpFdP3Av/h2o_jofaichow_started_from_python.out
  JVM stderr: /var/folders/4z/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T/tmpFdP3Av/h2o_jofaichow_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:	02 secs
H2O cluster version:	3.13.0.3981
H2O cluster version age:	29 days
H2O cluster name:	H2O_from_python_jofaichow_id7iqa
H2O cluster total nodes:	1

```
In [3]: # Import datasets from s3
        df_train = h2o.import_file("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_train.csv")
        df_test = h2o.import_file("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_test.csv")
```

```
Parse progress: |████████████████████████████████████████████████| 100%
Parse progress: |████████████████████████████████████████████████| 100%
```

```
In [4]: # Look at datasets
        df_train.summary()
        df_test.summary()
```

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4
type	int	enum	int	int	int	int	int	int	int
mins	10000.0		0.0	0.0	21.0	-2.0	-2.0	-2.0	-2.0
mean	165471.466667		1.85	1.55578703704	35.4053240741	-0.00523148148148	-0.122361111111	-0.15537037037	-0.21060
maxs	1000000.0		6.0	3.0	79.0	8.0	8.0	8.0	8.0
sigma	128853.314839		0.779559696278	0.522505078476	9.27675421641	1.12668964211	1.20086854503	1.20727030901	1.17217
zeros	0		9	37	0	10563	11284	11309	11905
missing	0	0	0	0	0	0	0	0	0

Scientific Advisory Council

Dr. Trevor Hastie

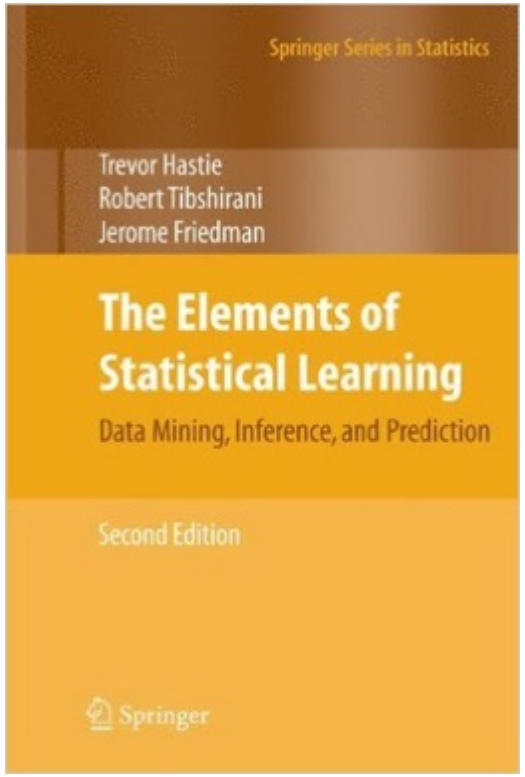
- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*

Dr. Robert Tibshirani

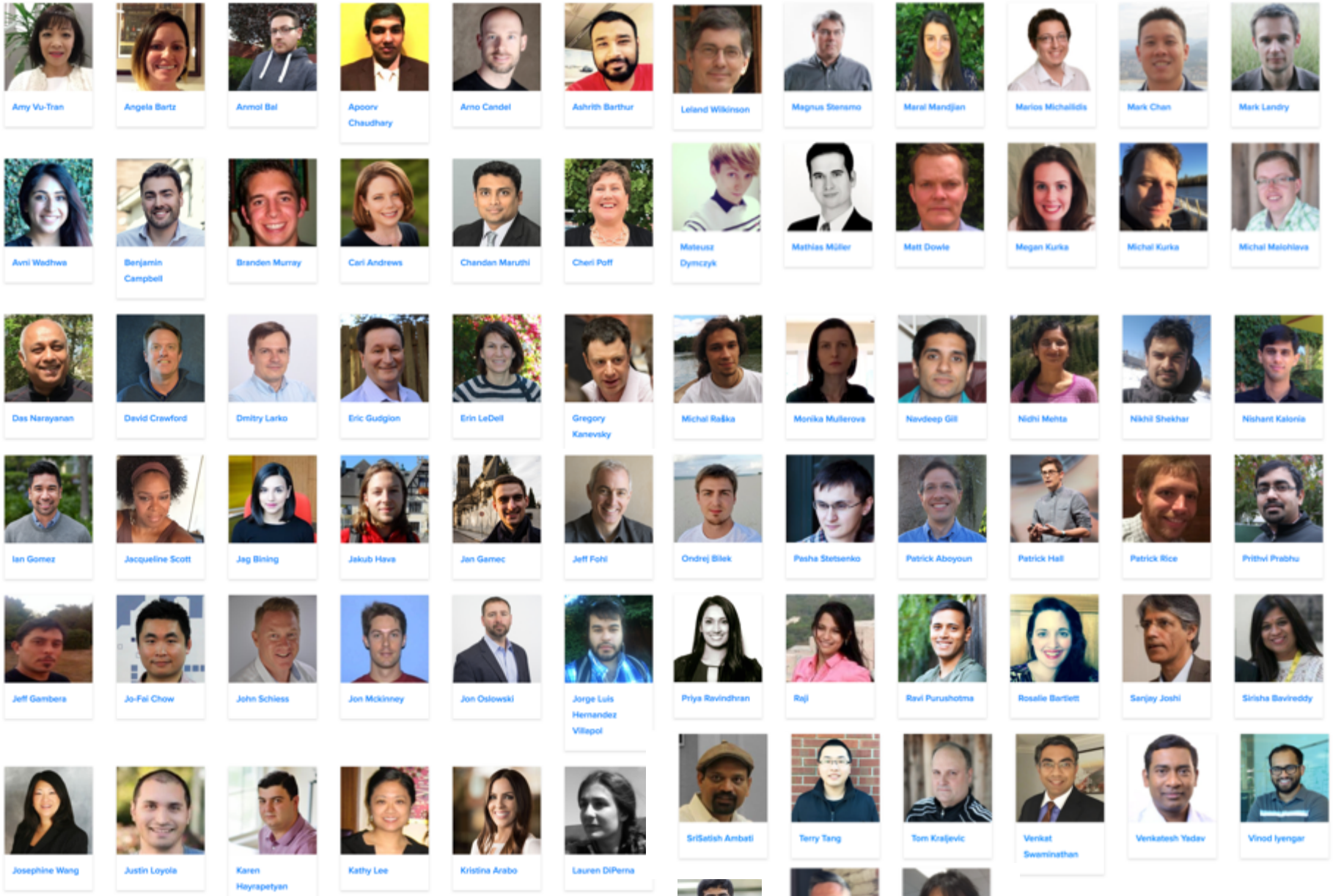
- Professor of Statistics and Health Research and Policy, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Author, *Regression Shrinkage and Selection via the Lasso*
- Co-author, *An Introduction to the Bootstrap*

Dr. Steven Boyd

- Professor of Electrical Engineering and Computer Science, Stanford University
- PhD in Electrical Engineering and Computer Science, UC Berkeley
- Co-author, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*
- Co-author, *Linear Matrix Inequalities in System and Control Theory*
- Co-author, *Convex Optimization*

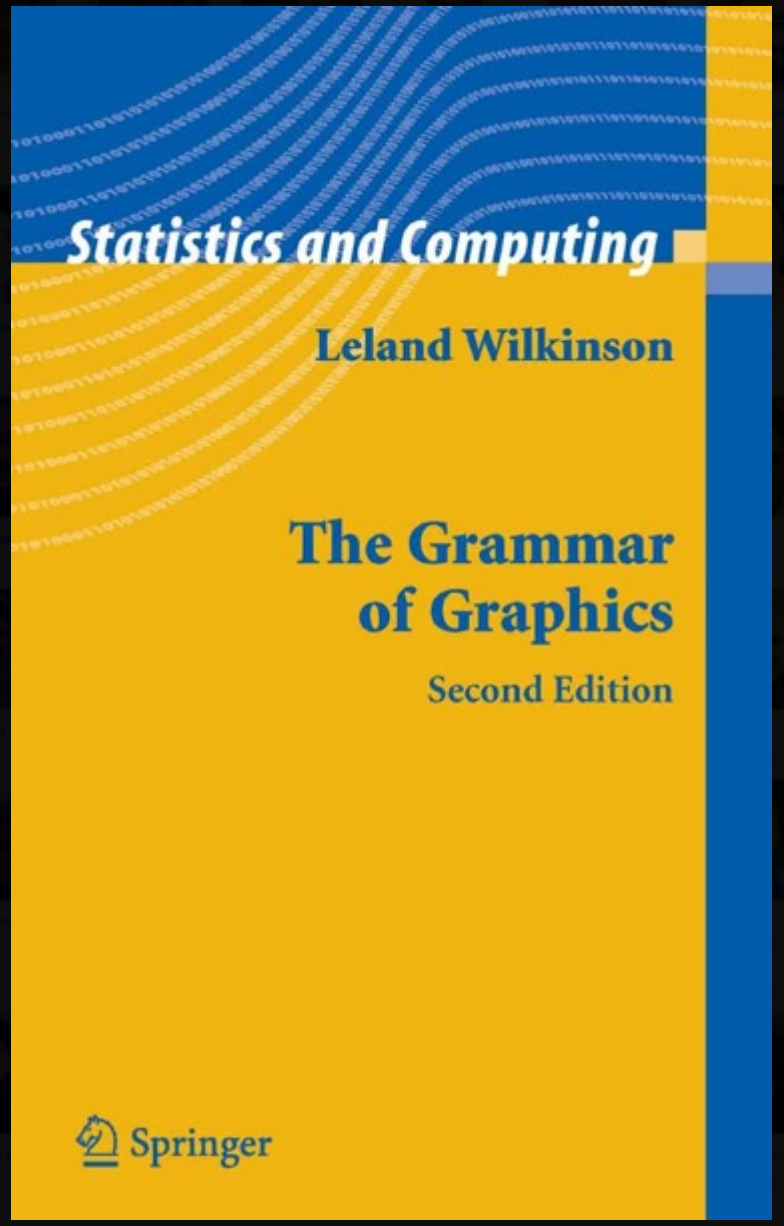


H2O.ai



H_2O Team

H_2O.ai

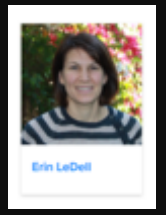


Prof. Leland Wilkinson
Origin of R Package `ggplot2`

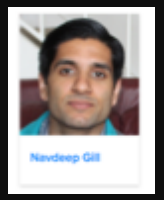
Matt Dowle

H_2O Team

H_2O AutoML



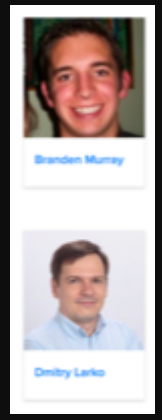
Erin LeDell



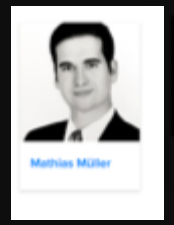
Navdeep Gill

H_2O Team

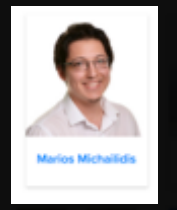
48th



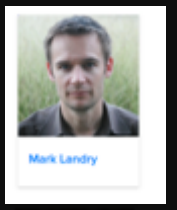
25th



4th



1st



33rd

Kaggle Grandmasters
(and their Highest Rank)

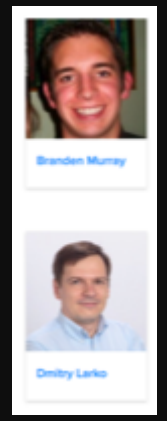
About 80,000 Kagglers

13th

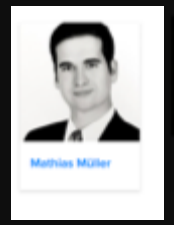
H2O Team

H2O.ai

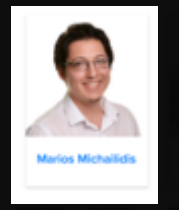
48th



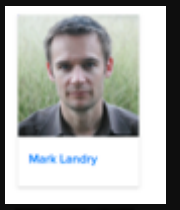
25th



4th



1st



33rd

181st

Hoping to get closer to them at some point …

13th

H2O Team

Worldwide Recognition in the H2O.ai Community

Open source community

222 OF FORTUNE 500 ♥ H_2O

8 OF TOP 10 BANKS

7 OF TOP 10 INSURANCE COMPANIES

4 OF TOP 10 HEALTHCARE COMPANIES

Paying Customers

"H2O.ai's reference customers gave it the highest overall score for sales relationship and overall service and support" - Gartner MQ 2018

H2O.ai

H2O.ai is a Leader in the 2018 Gartner Data Science and Machine Learning Platforms Magic Quadrant

- **Technology leader with most completeness of vision**

- Recognized for the mindshare, partner network and status as a **quasi-industry standard** for machine learning and AI

- **H2O.ai customers gave the highest overall score** among all the vendors for sales relationship and account management, customer support (onboarding, troubleshooting, etc.) and overall service and support

Figure 1. Magic Quadrant for Data Science and Machine-Learning Platforms

Source: Gartner (February 2018)

Get the Gartner Magic Quadrant here

Platforms with H_2O integration

H_2O + KNIME Talk at KNIME Summit Mar 2017

Figure 1. Magic Quadrant for Data Science and Machine-Learning Platforms

Source: Gartner (February 2018)

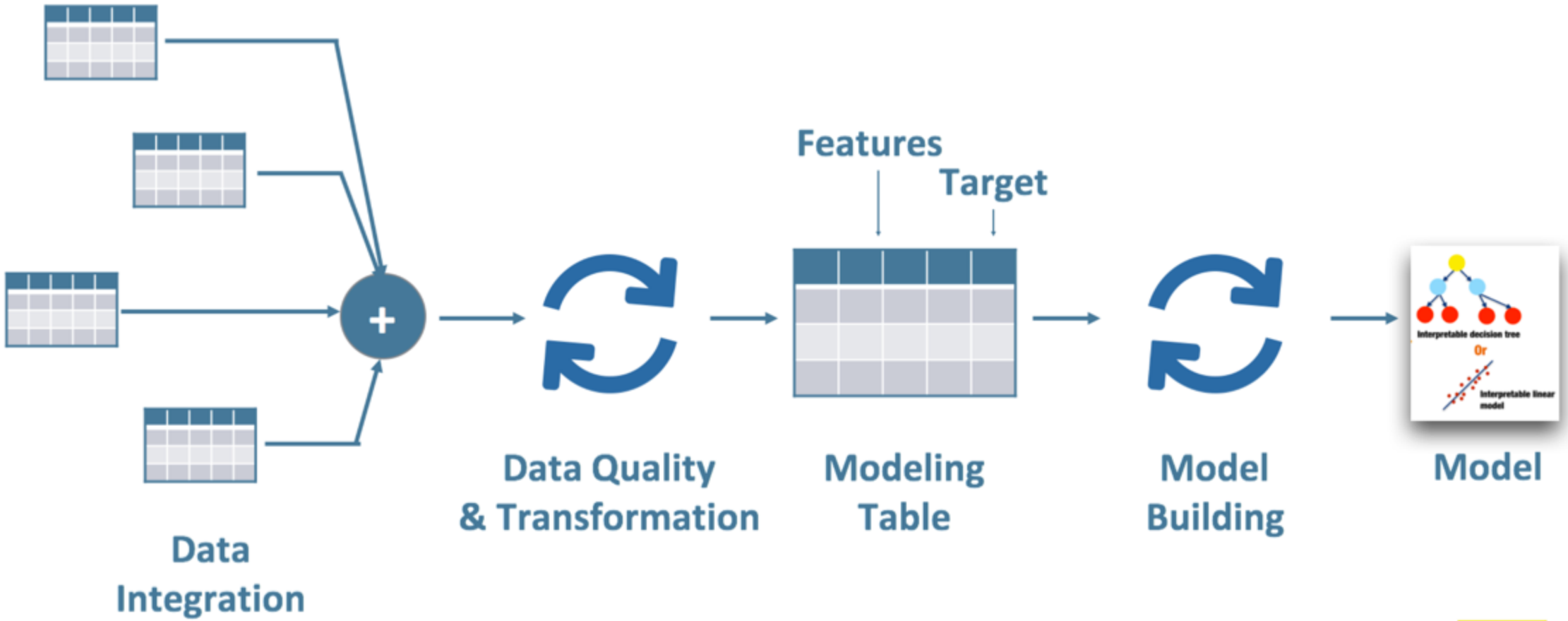
H2O.ai

H_2O AutoML

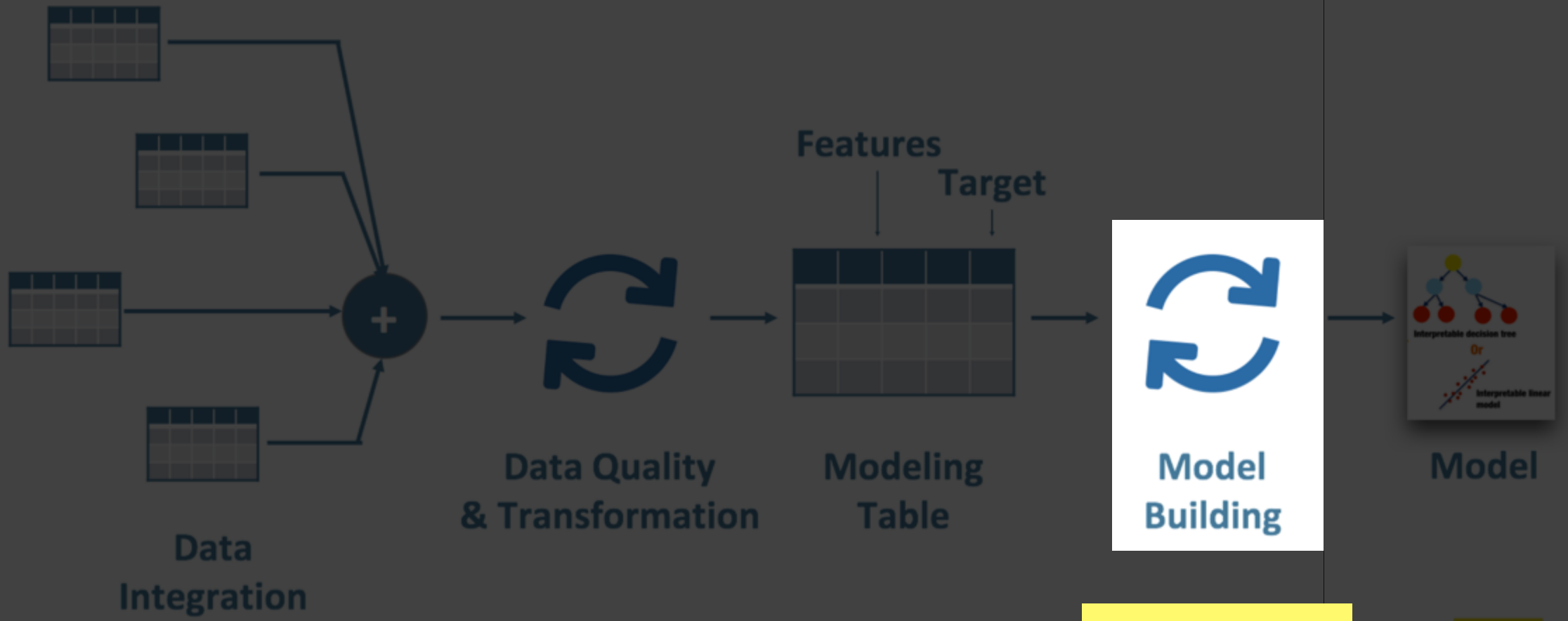
Automatic Machine Learning with H_2O

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html

Typical Enterprise Machine Learning Workflow



Typical Enterprise Machine Learning Workflow



AutoML

H_2O-3 Supervised Algorithms in AutoML

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

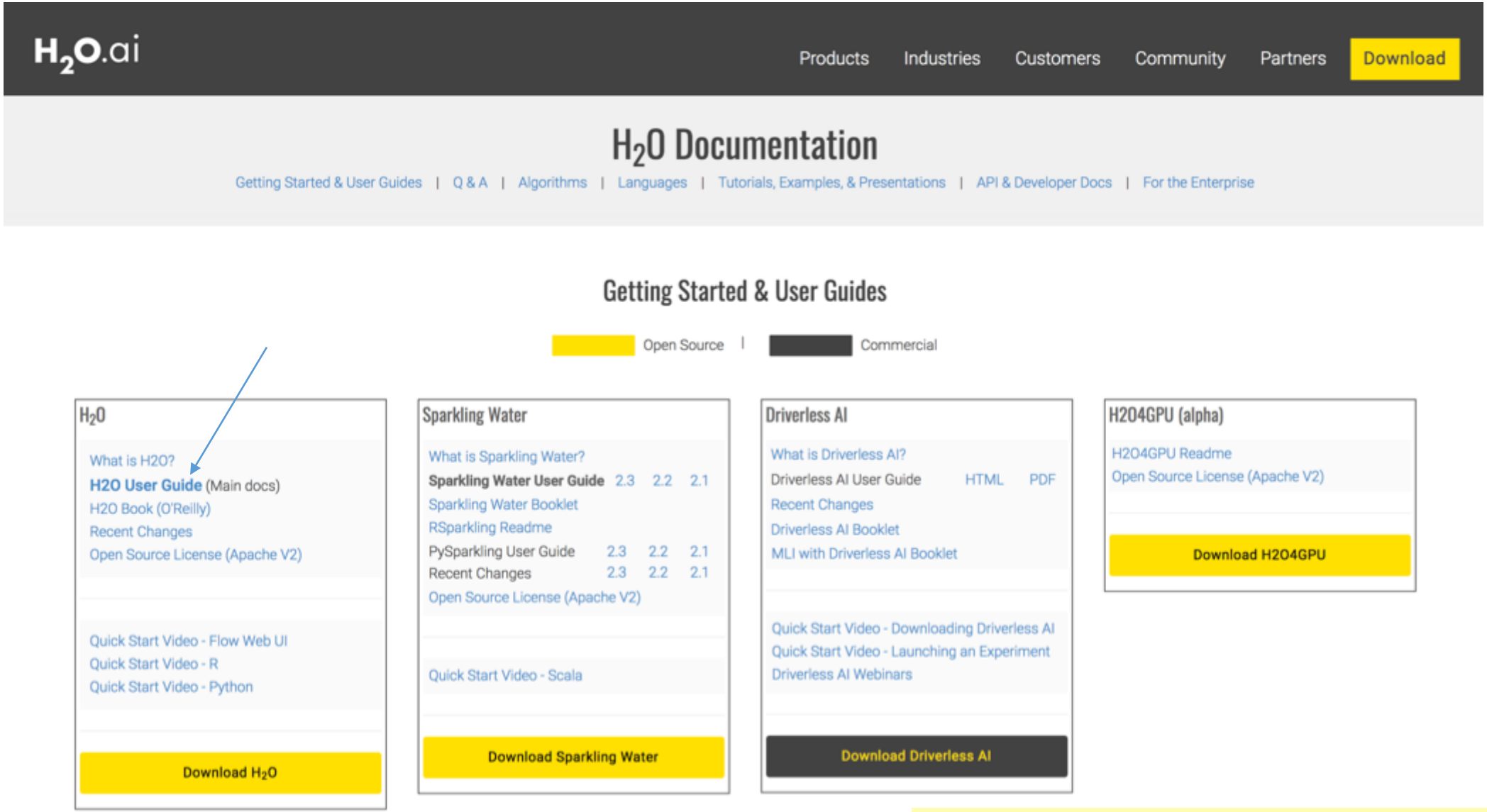
3.3 H2O AutoML: Multiple H2O Models + Stacked Ensemble

```
# Train multiple H2O models with H2O AutoML
# Stacked Ensembles will be created from those H2O models
# You tell H2O ...
#     1) how much time you have and/or
#     2) how many models do you want
# Note: H2O deep learning algo on multi-core is stochastic
model_automl = h2o.automl(x = features,
                          y = target,
                          training_frame = h_train,
                          nfolds = 5,                # Cross-Validation
                          max_runtime_secs = 120,    # Max time
                          max_models = 100,          # Max no. of models
                          stopping_metric = "RMSE", # Metric to optimize
                          project_name = "my_automl",
                          exclude_algos = NULL,      # If you want to exclude any algo
                          seed = n_seed)
```

3.4 AutoML Leaderboard

```
model_automl@leaderboard
```

```
##                                                  model_id
## 1 StackedEnsemble_BestOfFamily_0_AutoML_20180514_100853
## 2    DeepLearning_grid_0_AutoML_20180514_100853_model_1
## 3    StackedEnsemble_AllModels_0_AutoML_20180514_100853
## 4             GBM_grid_0_AutoML_20180514_100853_model_1
## 5             GBM_grid_0_AutoML_20180514_100853_model_3
## 6             GBM_grid_0_AutoML_20180514_100853_model_2
##   mean_residual_deviance     rmse      mae     rmsle
## 1               8.850089 2.974910 2.017611 0.1422685
## 2               9.119150 3.019793 2.124464 0.1572535
## 3               9.537272 3.088247 2.030157 0.1397577
## 4              11.299723 3.361506 2.176395 0.1501403
## 5              11.535687 3.396423 2.181420 0.1510191
## 6              11.737661 3.426027 2.208042 0.1524836
##
## [20 rows x 5 columns]
```



URL: **docs.h2o.ai**

H2O.ai

LIME
Local Interpretable Model-Agnostic Explanations

Acknowledgement

- **Marco Tulio Ribeiro**: Original LIME Framework and Python package

- **Thomas Lin Pedersen**: LIME R package

- **Matt Dancho**: LIME + H2O AutoML example + LIME R package improvement

- **Kasia Kulma**: LIME + H2O AutoML example

Why Should I Trust Your Model?

System that performs behaviour but you don't know how it works

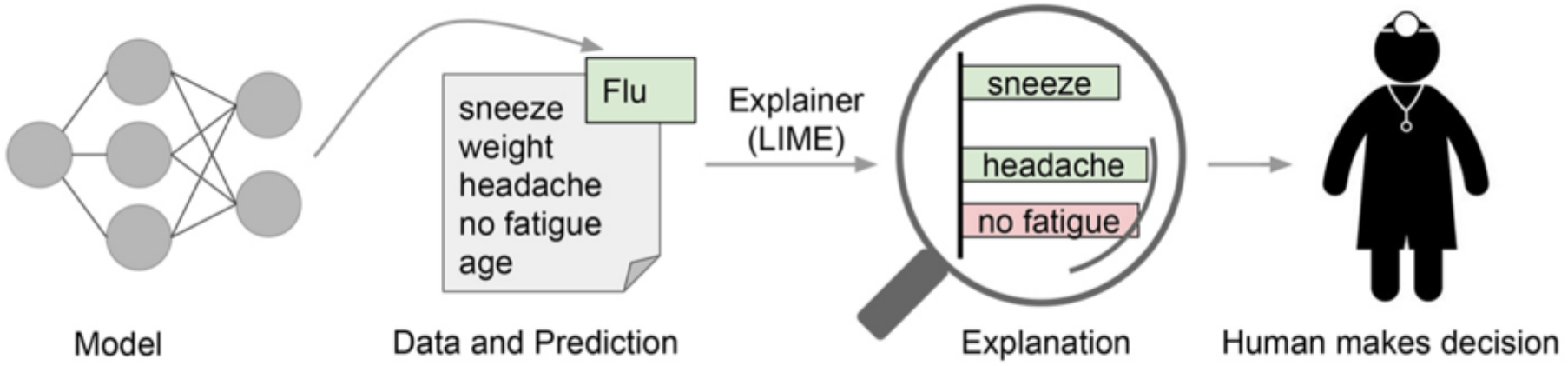


Figure 1. Explaining individual predictions to a human decision-maker. Source: Marco Tulio Ribeiro.

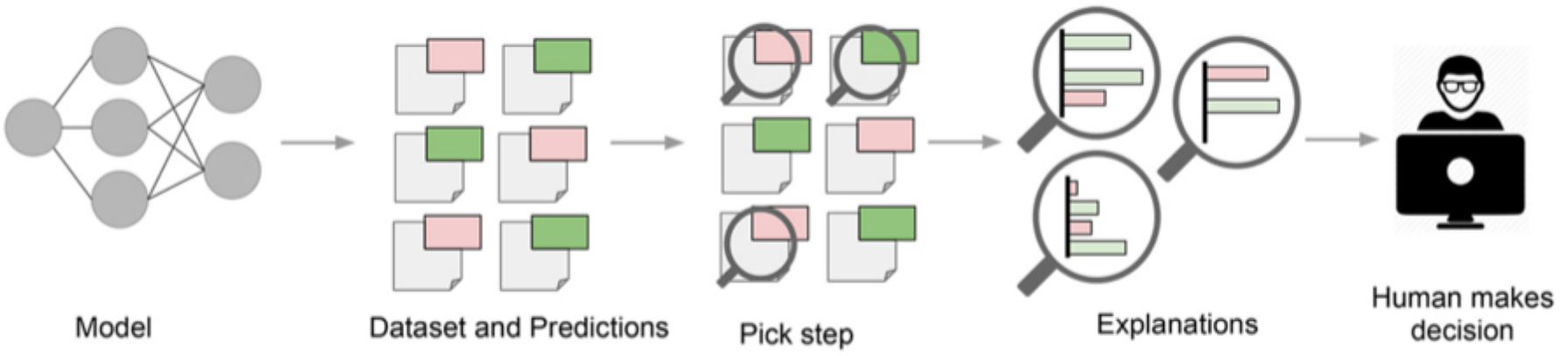
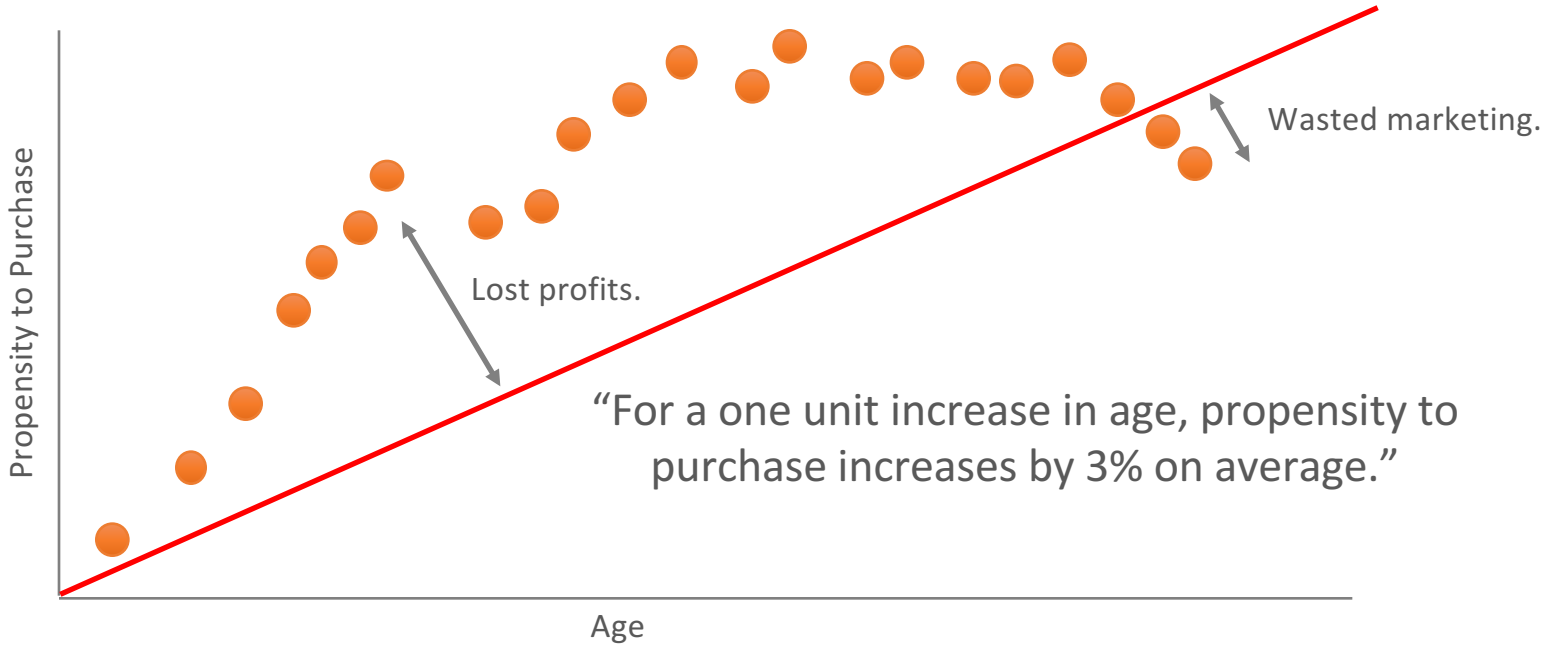


Figure 2. Explaining a model to a human decision-maker. Source: Marco Tulio Ribeiro.

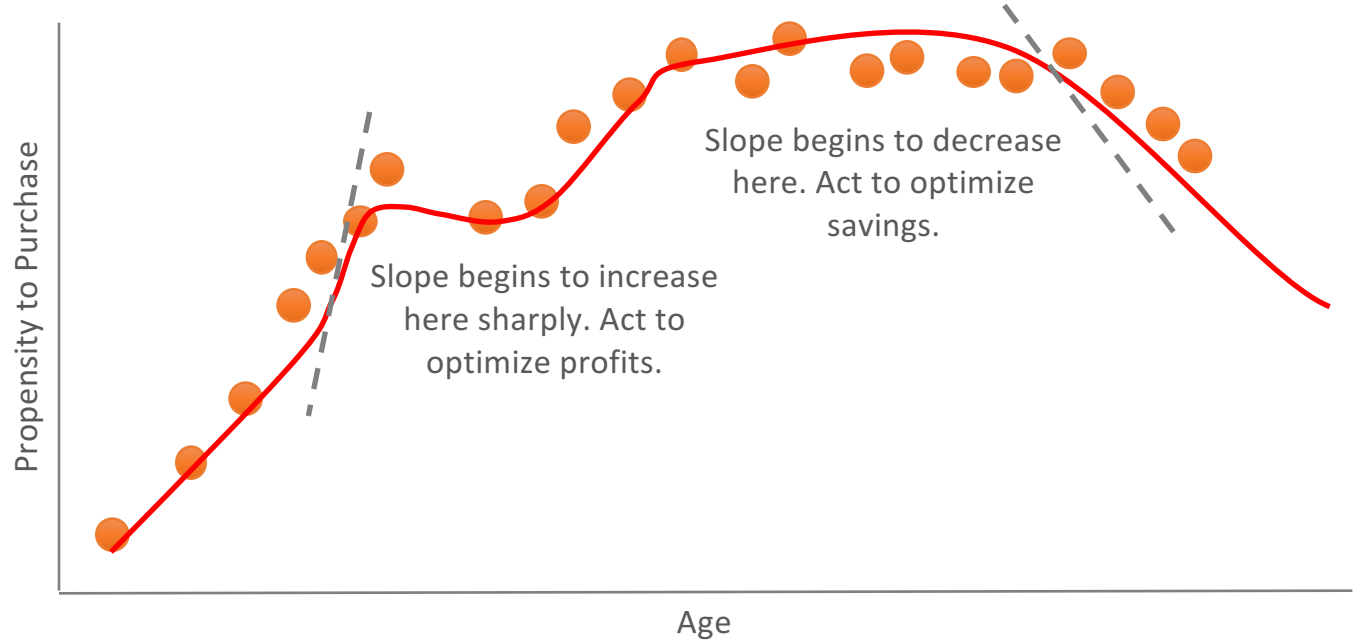
Linear Models

Exact explanations for approximate models.



Machine Learning

Approximate explanations for exact models.



H2O.ai

Local Interpretable Model-Agnostic Explanations
LIME - How does it work?

Theory

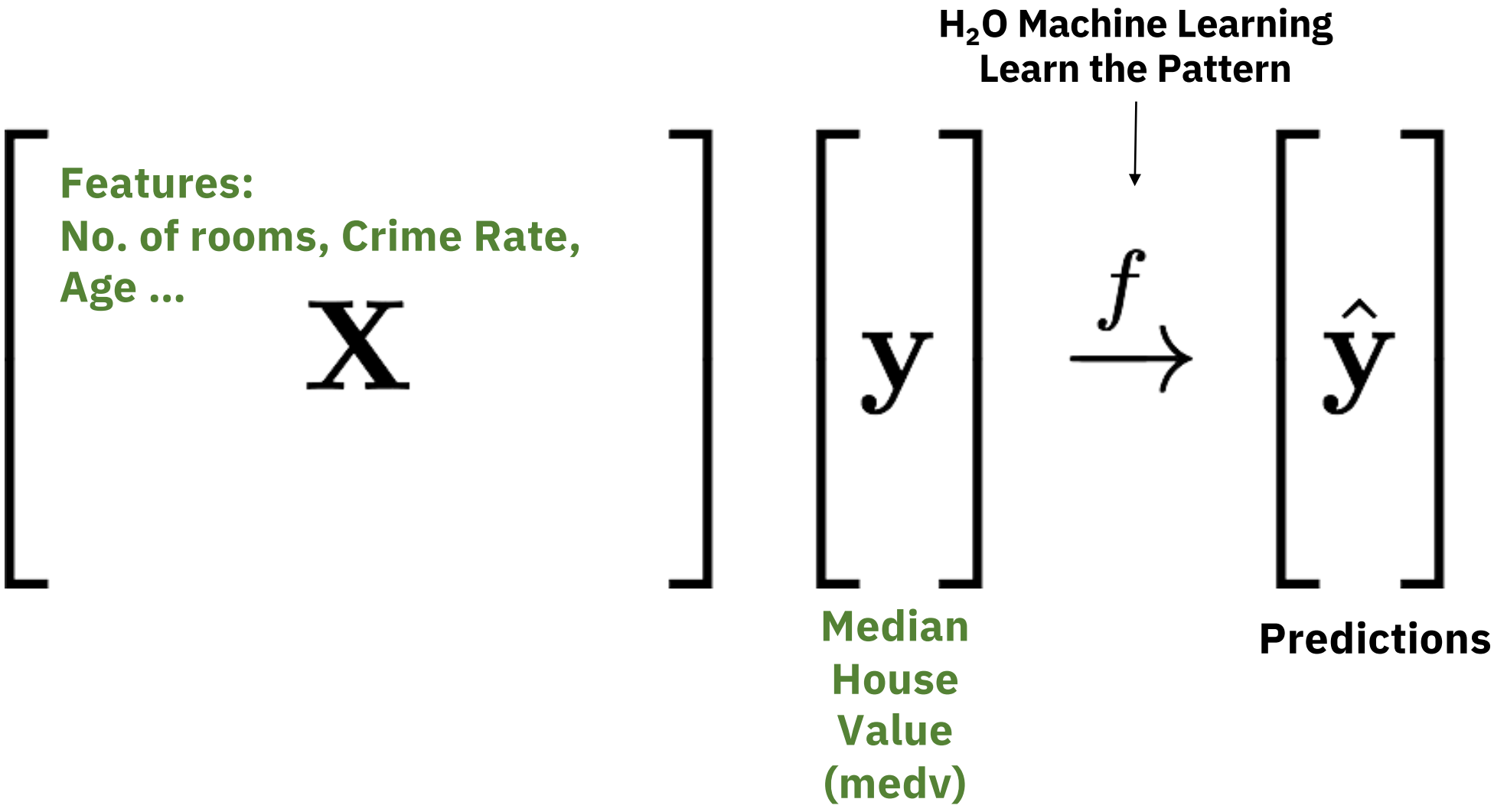
- LIME approximates model locally as logistic or linear model
- Repeats process many times
- Outputs features that are most important to local models

Outcome

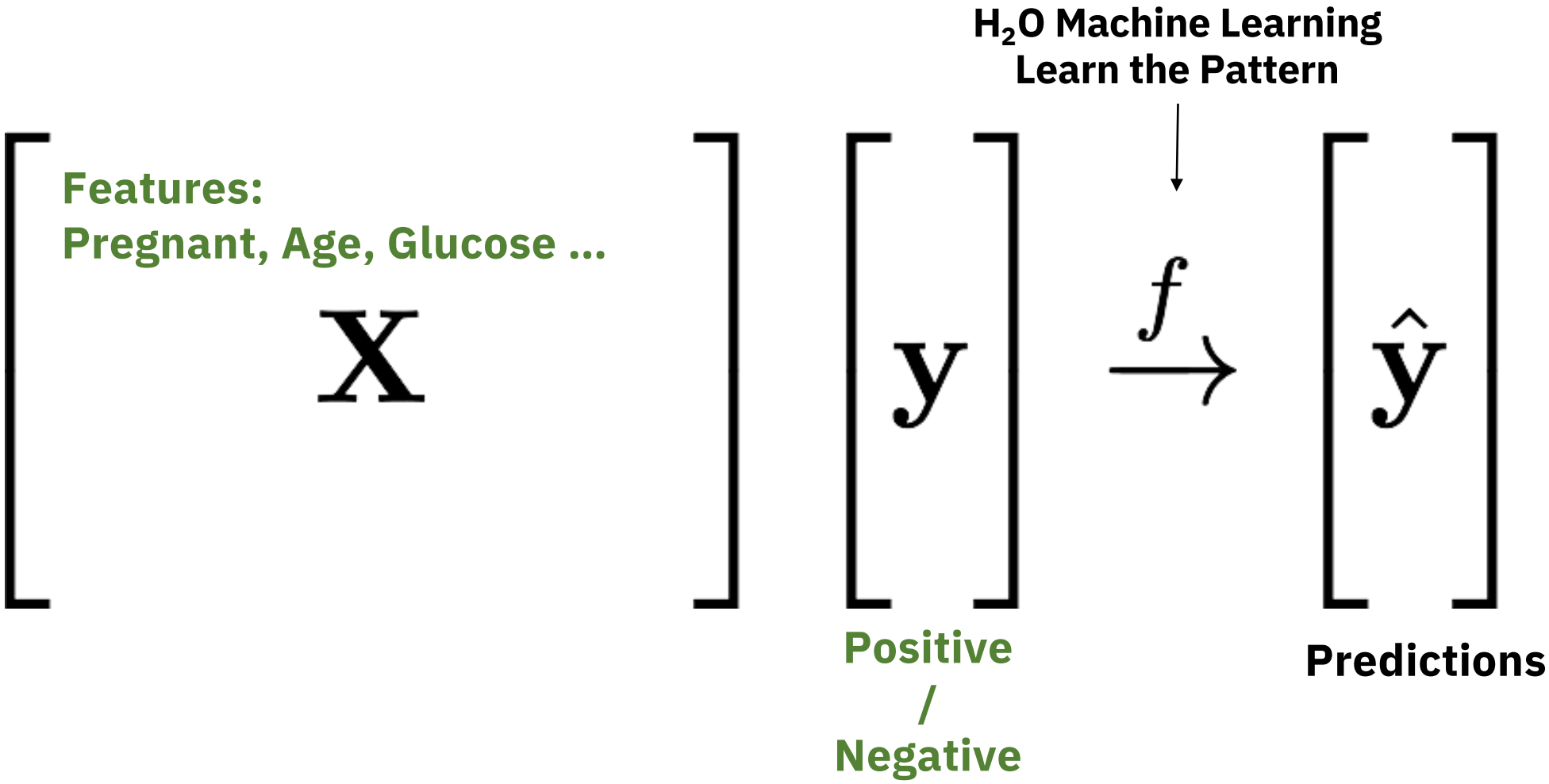
- Approximate reasoning
- Complex models can be interpreted
  - Neural nets, Random Forest, Ensembles etc.

Worked Examples

Learning from **Boston Housing** Data



Learning from **Diabetes** Data



Quick Recap

#ODSC

Why?

- Most users/organizations can benefit from **automatic machine learning pipelines**.
  - Eliminate time wasted on repetitive tasks, human errors, debugging etc.
- GDPR mandates a “right to explanation” from machine learning models.
  - **model interpretations** are crucial for those who must explain their models to regulators or customers.

You will learn …

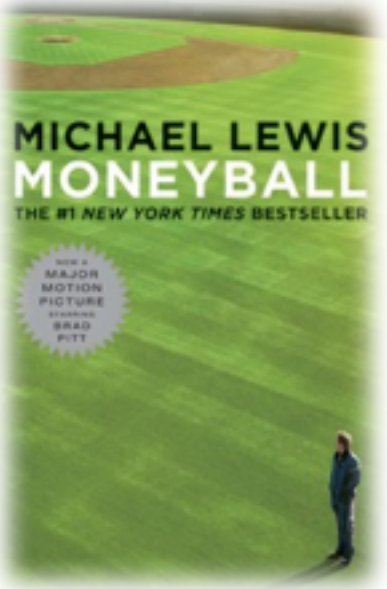
- How to build high quality **H_2O** models (almost) automatically.
- How to explain predictions from complex H_2O models with **LIME**.
- **Bonus**: A real use case that led to a **multimillion-dollar** baseball decision earlier this year.

Quick Recap

Moneyball: The Multimillion-Dollar Business Problem

The quest to find the most undervalued players (before other teams notice them)

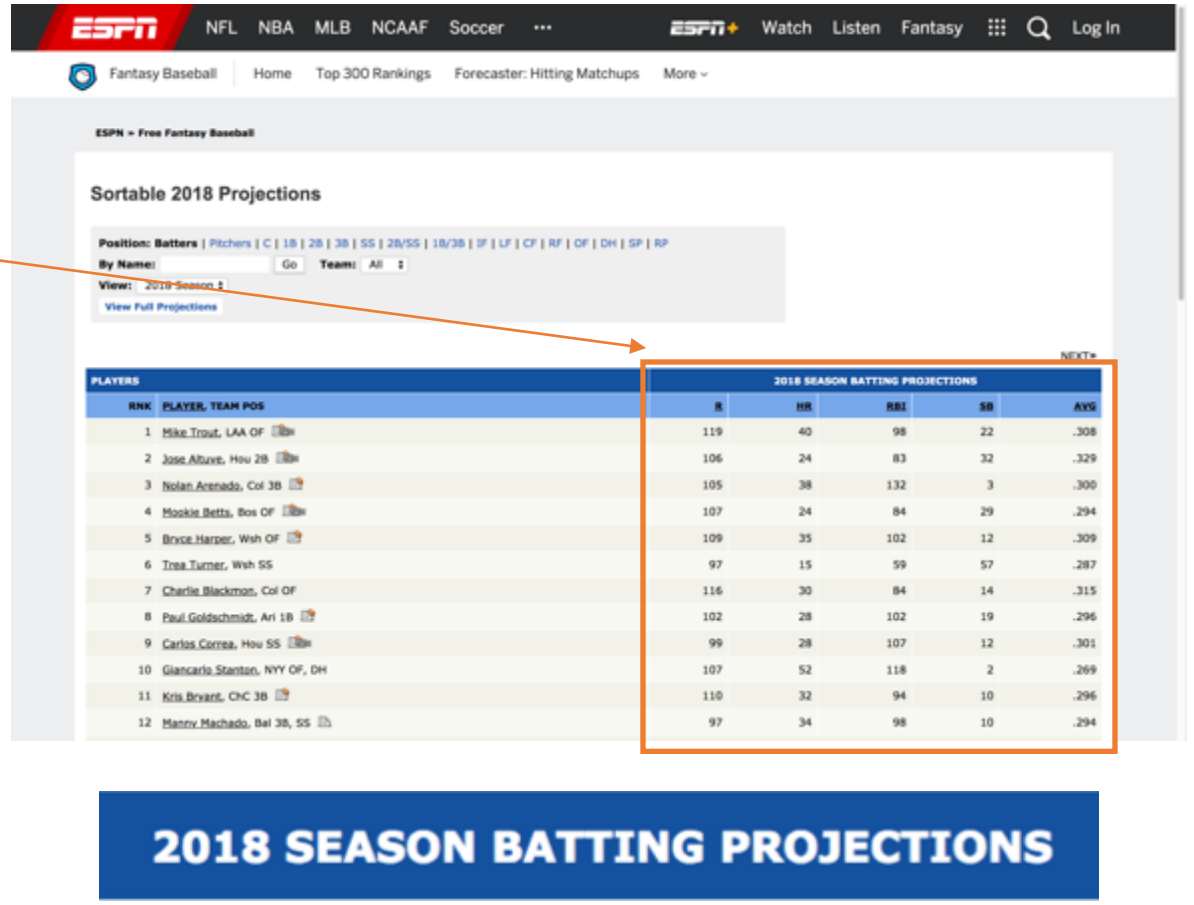
Source: Moneyball, 2011 Columbia Pictures



H2O.ai

The Real Business Problem in Major League Baseball (MLB)

- Existing Forecasts (e.g. ESPN) are usually projections for the **next year only**.
- MLB players usually consider terms for 3 to 5 years when they sign a new contract.
- MLB teams need to consider players' **long-term performance** (i.e. > 1 year).



H2O.ai

The Moneyball Team

David Kearns

PM @ IBM Data Science

Jo-Fai Chow

Data Scientist @ H_2O.ai

Ari Kaplan

Mr. Moneyball @ Aginity

IBM + aginity + H_2O.ai

Baseball Player Performance Data

- Open data – **Lahman** Database.
- Proprietary data (**AriDB**) from Ari Kaplan – our real Moneyball guy.
- Enriched Lahman data with Ari's Data – Final dataset for predictive modelling

Lahman Database

http://www.seanlahman.com/baseball-archive/statistics/

Attribute	Description
playerID	Player ID code
yearID	Year player was born
G	Games
AB	At Bats
R	Runs
H	Hits
2B	Doubles
3B	Tripples
HR	Homeruns
SO	Strike Outs
IBB	Intentional Walks
SF	Sacrifice flies

Ari's Database

- Private database containing 5 years of data
- Pitch-by-pitch play for each MLB game:
  - Pitch type, top speed, end speed, spin rate, x, y, z coordinates, batter result etc.

Attribute	Description
Pitch_Type	Two - character code of type of pitch. FF=fastball, CU=curveball, SL=slider, etc.
Spin_rate	Spin of the pitch in rotations per minute. One of the top fields for a feature...the theory is the more spin the harder it is to hit.
Start_speed	The velocity of the pitch in mph (when it leaves the hand, which is the measure used for tv).
End_speed	The velocity of the pitch when it arrives at the plate
Z0	Feet off the ground when the pitch is released.
Spray_x	When ball is hit into play, this is the x - coordinate of where it is hit/picked up by a fielder
Spray_y	When ball is hit into play, this is the y - coordinate of where it is hit/picked up by a fielder
Spray_des	Classification of type of hit: pop out, flyout, groundout, hit, error

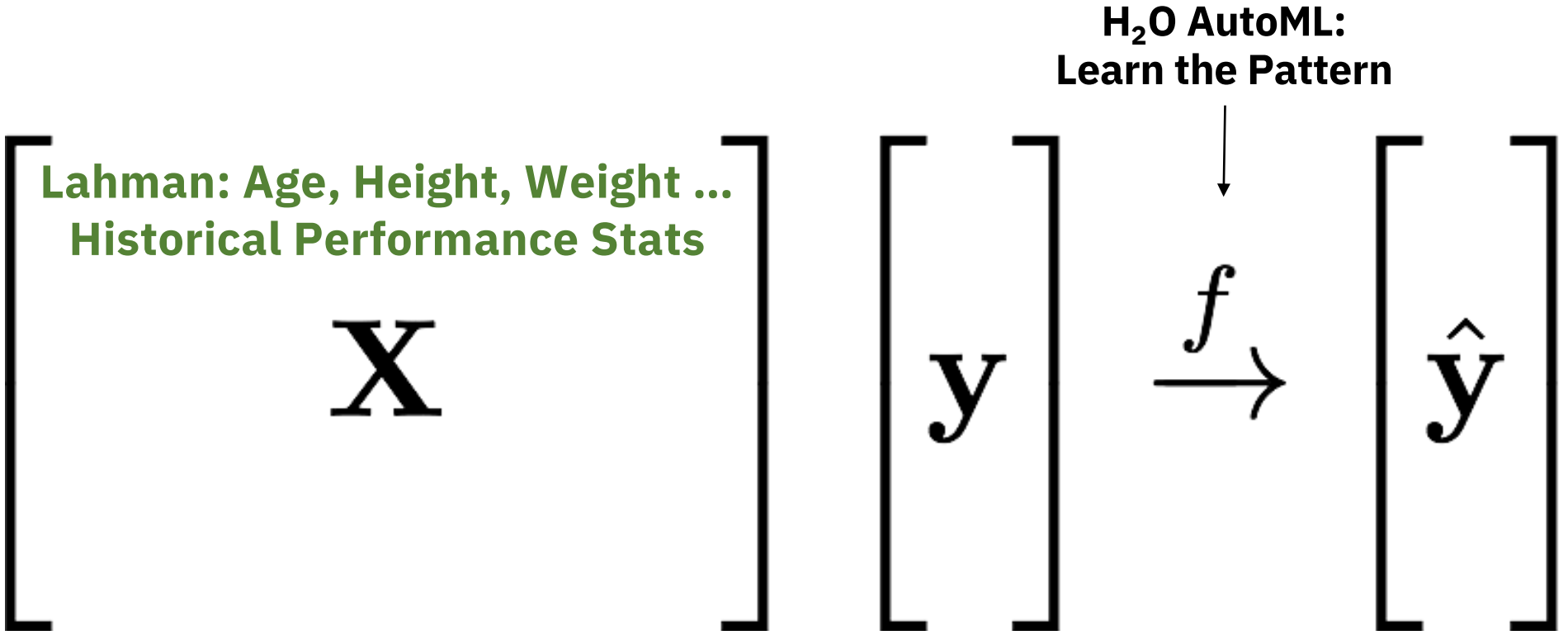
Predictive Modelling – H_2O AutoML

- Framed data as regression problems for performance prediction.
- Historical player performance as features.
- Used H_2O AutoML to build ensembles (linear model, random forests, gradient boosting, and deep neural networks).


```
# Install 'h2o' from CRAN
install.packages('h2o')
```


H2O.ai

Approach One: Learning from **Lahman** only



Sliding Windows (Stats from previous n years)

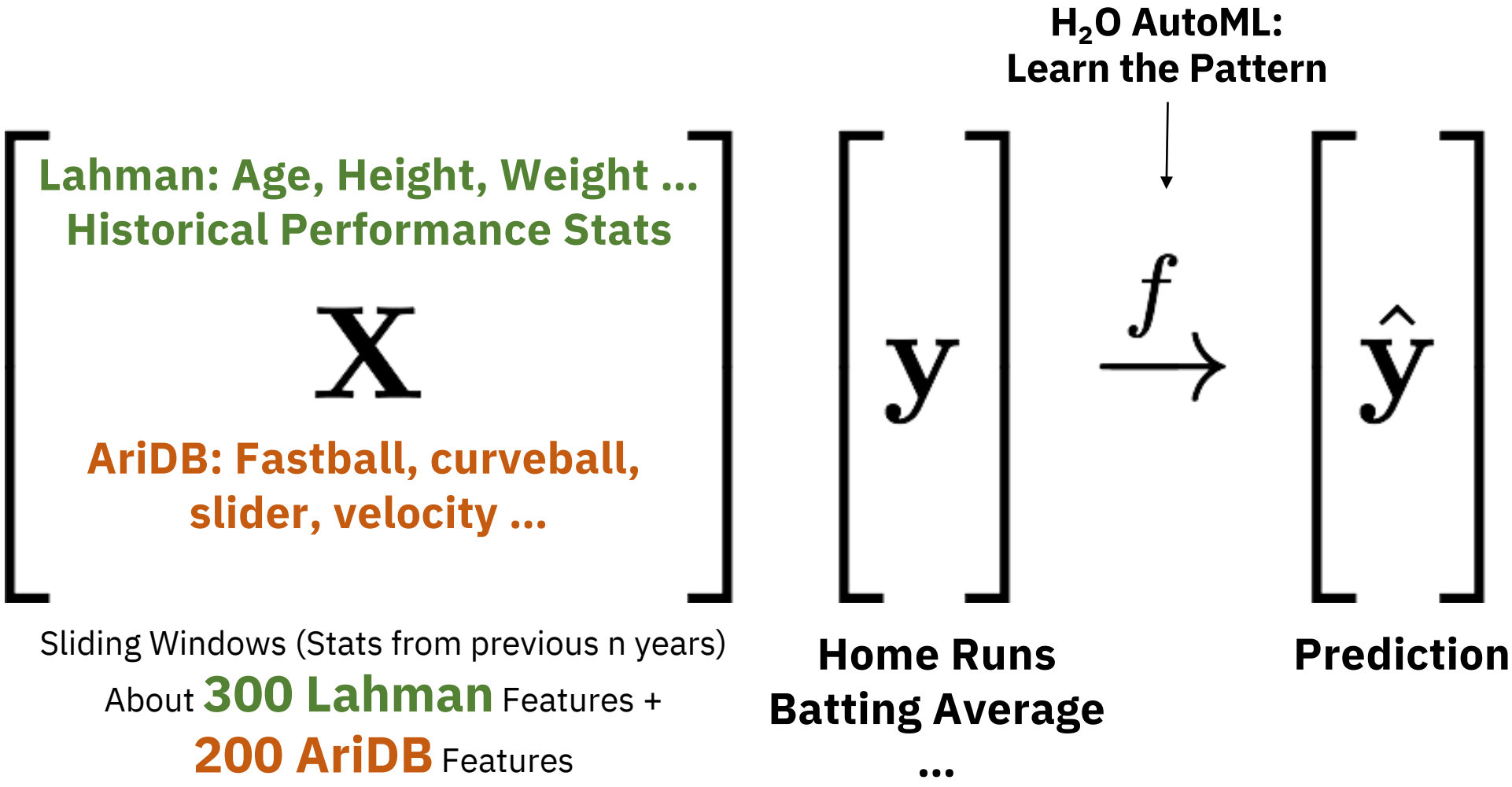
About **300 Lahman** Features

Home Runs
Batting Average
…

Predictions

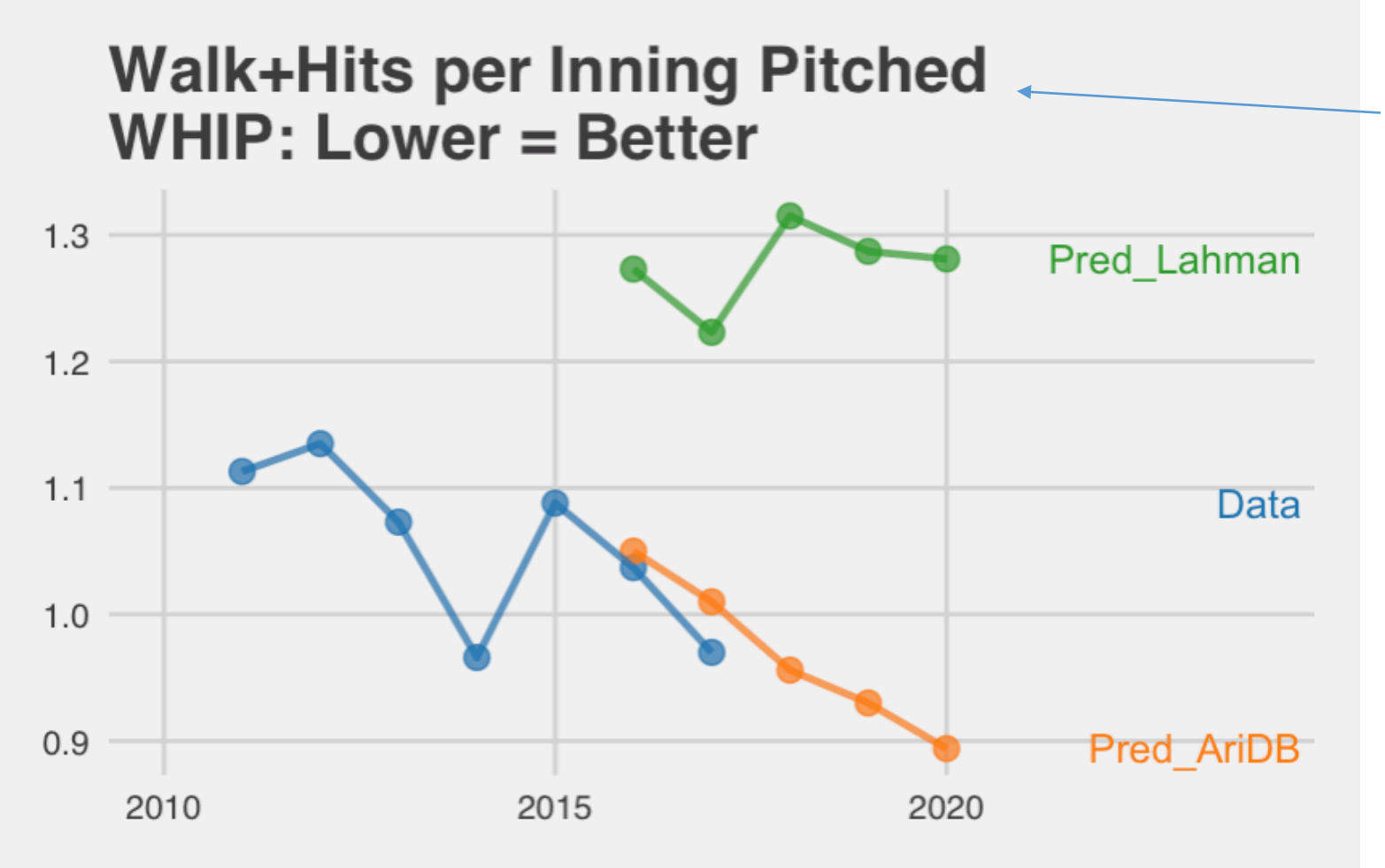
H2O.ai

Approach Two: Learning from **Lahman** & **AriDB**



H2O.ai

Predictive Modelling – H_2O AutoML

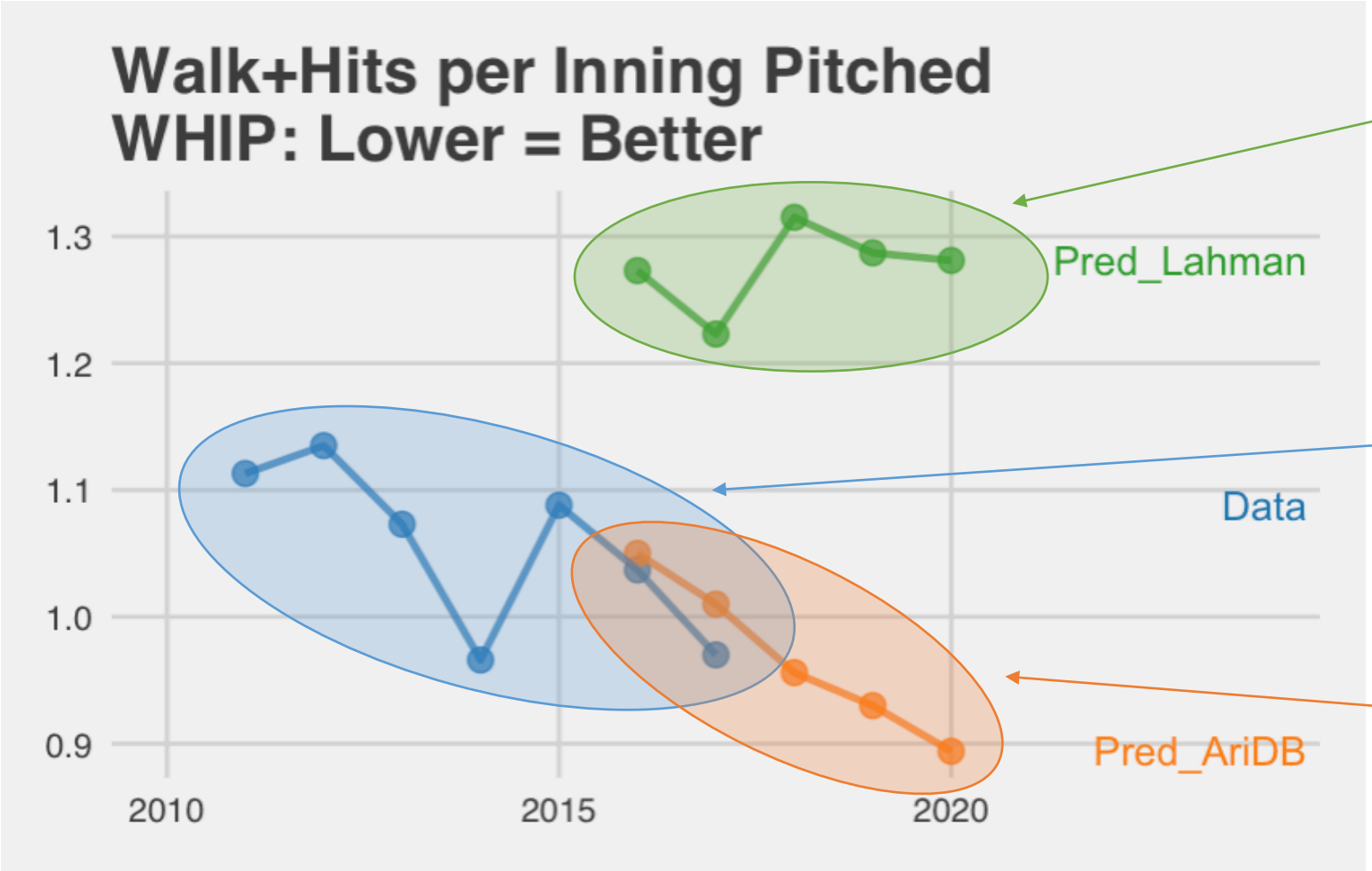


One of Many Targets
(e.g. Home Runs, Batting Average)


```
# Install 'h2o' from CRAN
install.packages('h2o')
```


H2O.ai

Predictive Modelling – H_2O AutoML



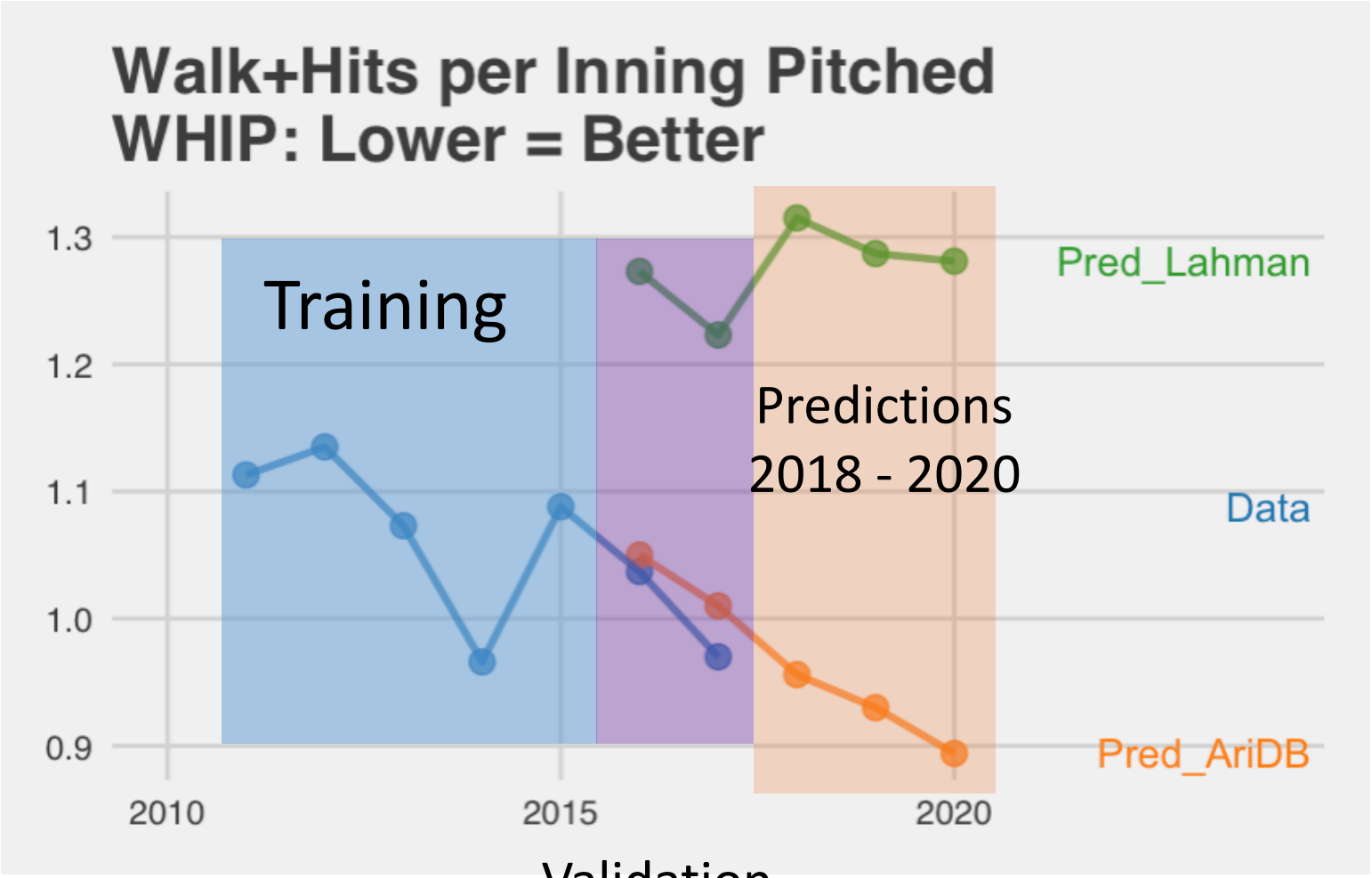
Results from models based on Lahman data only

Historical player performance data

Results from models based on final dataset (Lahman + AriDB)

H2O.ai

Predictive Modelling – H_2O AutoML



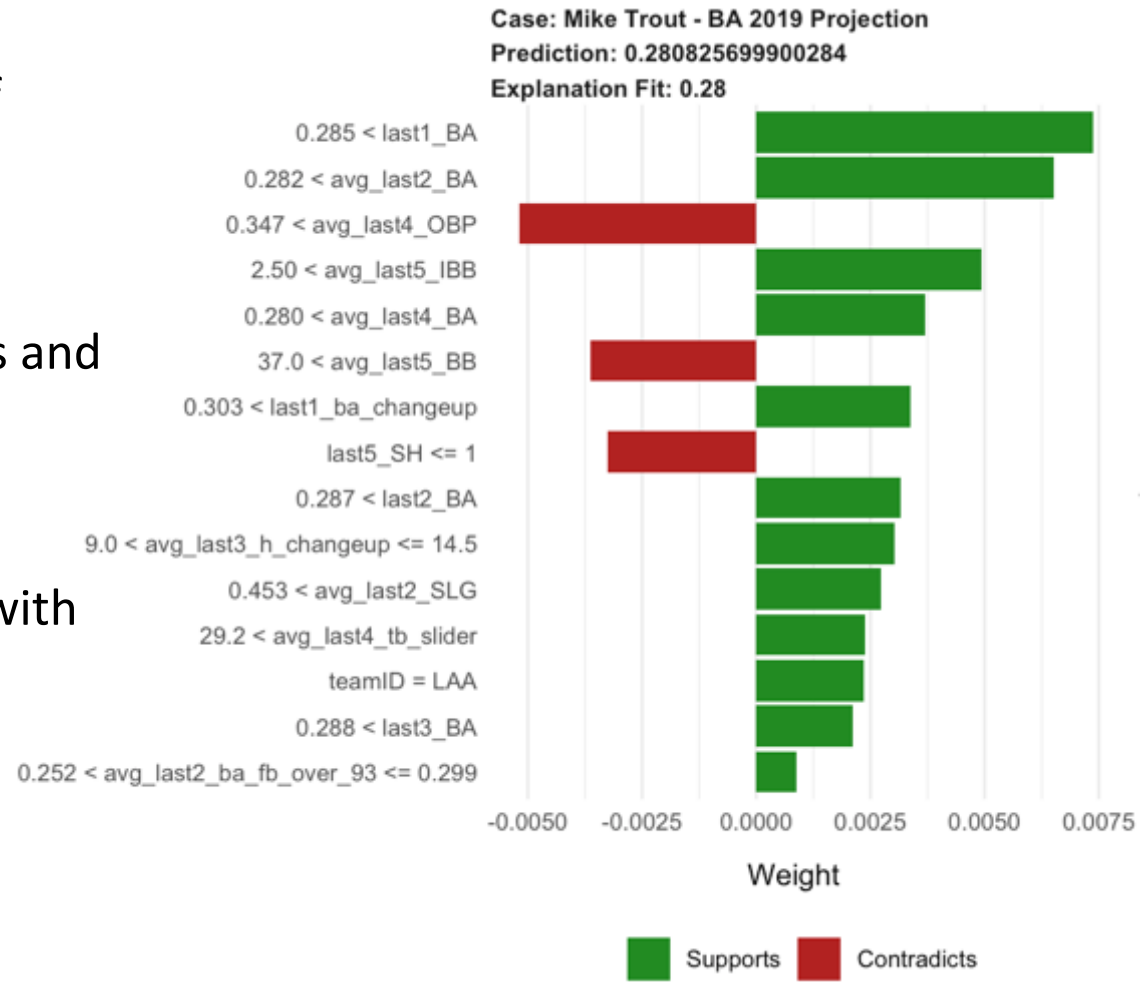
Validation
(2016 & 2017)


```
# Install 'h2o' from CRAN
install.packages('h2o')
```


H2O.ai

Explaining the Predictions
LIME – Local Interpretable Model-agnostic Explanations

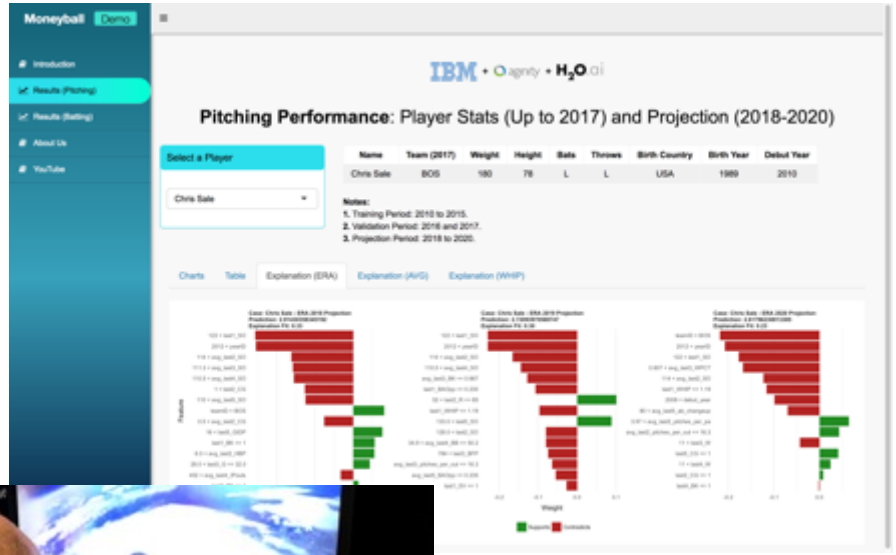
- Approximate reasoning of complex ML models (ensembles).
- Most important attributes and their contributions to the predictions.
- Ari validated the models with his 30+ years of baseball domain knowledge.
- He trusted the models.




```
# Install 'lime' from CRAN
install.packages('lime')
```

H2O.ai

Putting Everything Together – Moneyball Shiny App

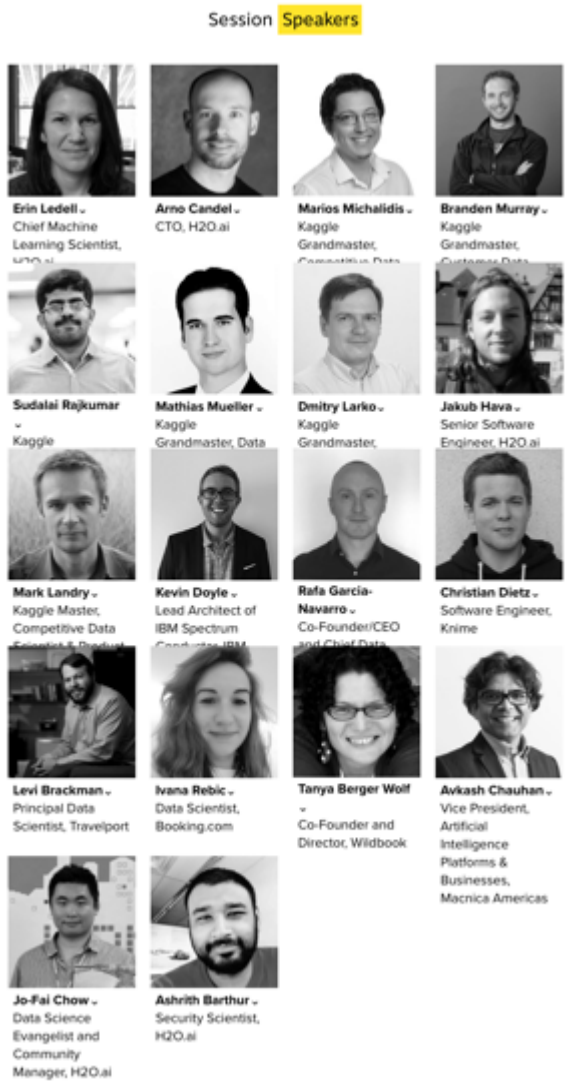


Live Demo

H2O.ai

If you want to hear the Moneyball story from Ari and David …

29th & 30th Oct, London



More real-world use cases

+

All H_2O Kaggle Grandmasters

+

Hands-on Training

H_2O.ai

Thanks!

#ODSC

- More Info, Code, and Slides
  - bit.ly/ odsc2018_h2o
- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe