The Making of a Real-World Moneyball

Finding Undervalued Players with H_2O, LIME and Shiny

Jo-fai (Joe) Chow

Data Science Evangelist / Community Manager

joe@h2o.ai

@matlabulous

More Info → https://bit.ly/

h2o_meetups

Berlin KNIME Users (KNIME + H_2O.ai Joint Meetup)

About Me

- **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 360 Selfie Guy**

    … The harsh reality of startup life …

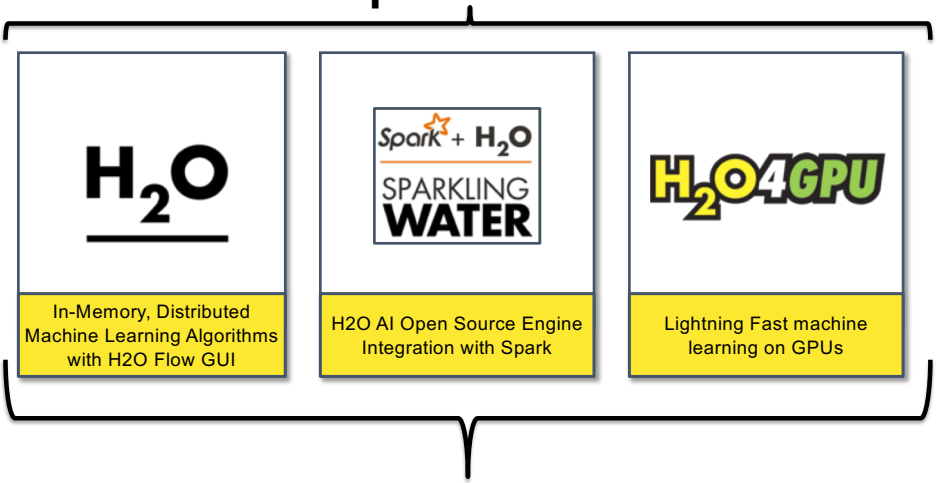
H2O.ai

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

Open Source



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

DRIVERLESSAI

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

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

Growing Worldwide Open Source Community

14,000 Companies using H2O

155,000 data scientists

H2O World
NYC, London, SF
Thousands attending live and online

116K Meet up Members

H_2O.ai Meetup Groups

Members	Groups	Countries
102,646	42	20

London Artificial Intelligence & Deep Learning

London, United Kingdom · 7,410 members · Public group

Organized by
Ian Gomez and **2 others**

Part of **H2O Artificial Intelligence and Machine Learning – 42 groups**

Share:

Contact Joe Chow
joe@h2o.ai

If you want to …

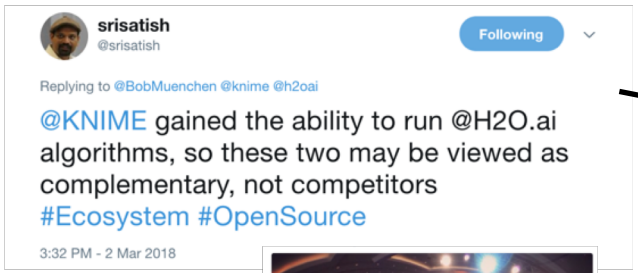
- Give a talk about AI / machine learning use case (it is a great opportunity to promote your work)
- Host a joint meetup with H_2O.ai

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

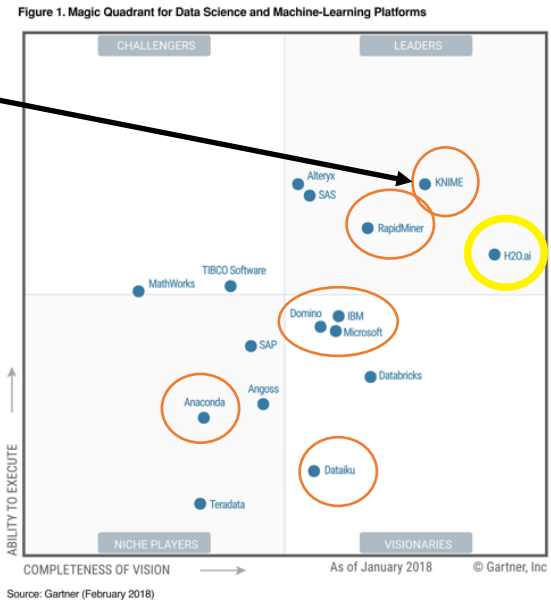
- **Technology leader with most completeness of vision**
- Recognized for the mindshare, partner network and status as a **quasi-industry standard** for machine learning and AI
- **H2O.ai customers gave the highest overall score** among all the vendors for sales relationship and account management, customer support (onboarding, troubleshooting, etc.) and overall service and support

Get the Gartner Magic Quadrant here

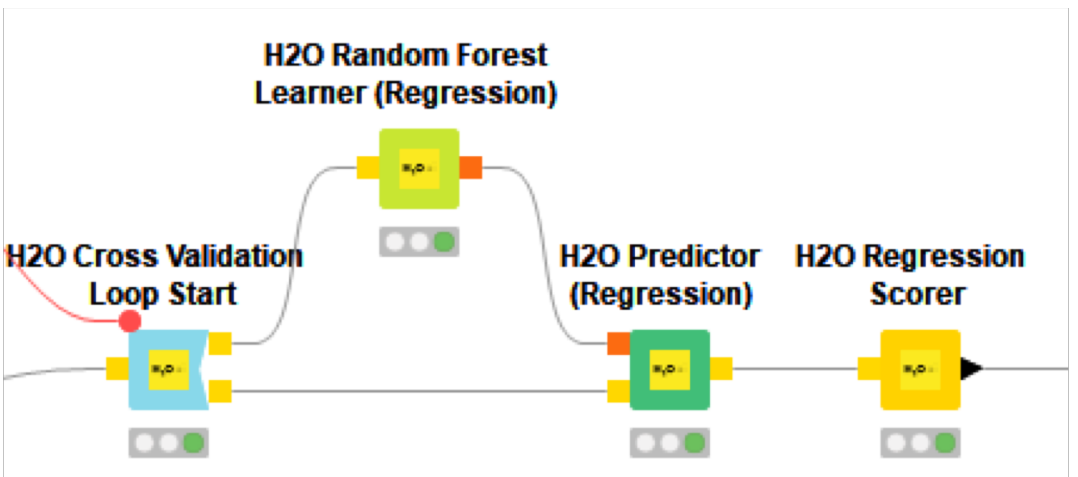
Platforms with H_2O Integration



H_2O + KNIME Talk at KNIME Summit March 2018



Open for Innovation
KNIME
+
H2O.ai



Solving a Kaggle Challenge using the combined power of KNIME Analytics Platform & H2O

Mon, 06/04/2018 - 14:04 — Marten Pfannen…

Some time ago, we set our mind to solving a popular Kaggle challenge offered by a Japanese restaurant chain: predict how many future visitors a restaurant will receive.

This is a classic demand prediction problem: how much energy will be required in the next N days, how many milk boxes will be in demand tomorrow, and how many customers will visit our restaurants tonight? We already know how to use KNIME Analytics Platform to solve this kind of time series analytics problems (see whitepaper on energy prediction). So, this time we decided to go for a different approach: a mixed approach.

Thanks to the open architecture of KNIME Analytics Platform, we can practically plug in almost any open source analytics tool, such as Python, R, Weka, to name just three very prominent examples - and, more recently also H2O.

We already developed a cross-platform ensemble model to predict flight delays (another popular challenge). Here, cross-platform means that we trained a model with KNIME, a model with Python, and a model with R. These models from different platforms were then blended together as an ensemble model in a KNIME workflow. Indeed, one of KNIME Analytics Platform's many qualities consists of its capability to blend data sources, data, models, and, yes, also tools.

For this restaurant demand prediction challenge we decided to raise the bar and develop a solution using the combined power of KNIME Analytics Platform and H2O.

The KNIME H2O Extension

In order to use H2O within KNIME Analytics Platform, all you need to do is install the H2O extension and you're ready to go. Check this video, if you do not know how to install a KNIME extension.

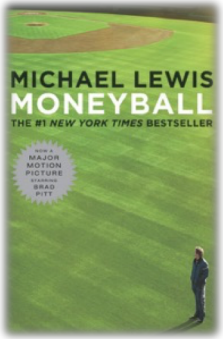
The integration of H2O in KNIME offers an extensive number of nodes and encapsulating functionalities of the H2O open source machine learning libraries, making it easy to use H2O algorithms from a KNIME workflow without touching any code - each of the H2O nodes looks and feels just like a normal KNIME node - but the workflow reaches out to the high performance libraries of H2O during execution.

***Figure 1**. All available nodes in the KNIME H2O extension*

Moneyball:
The Multimillion-Dollar Business Problem

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

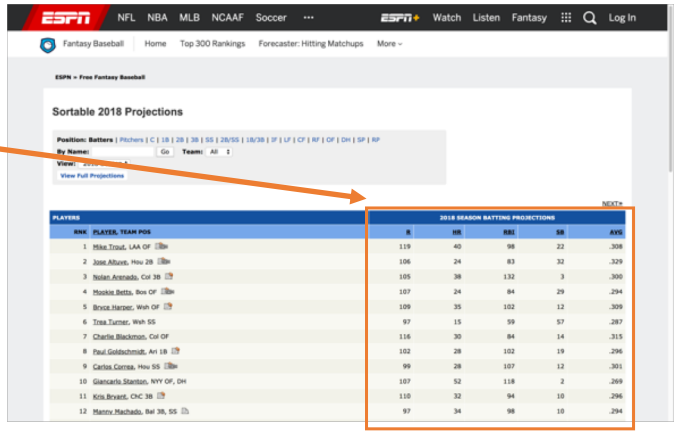
Source: Moneyball, 2011 Columbia Pictures



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

In case you're wondering … final project result

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

$20M

multi-year contract

finalised two weeks
before the regular season

Framing the Business Problem for Machine Learning

Code on GitHub (without Ari's proprietary data)

https://github.com/woobe/moneyball

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.

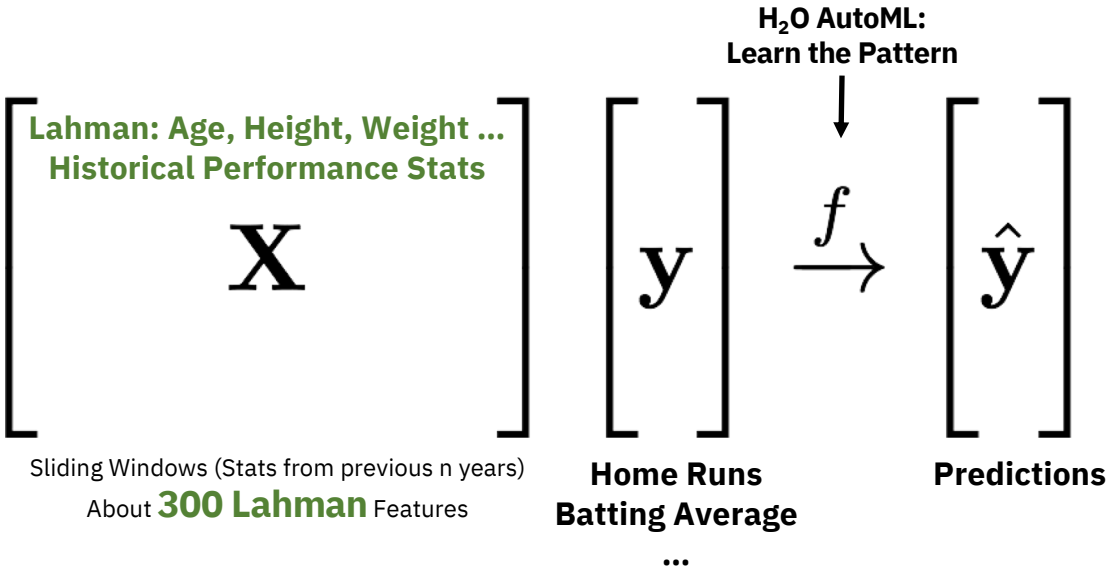
Attribute	Description
Pitch_Type	Two - character code of type of pitch. FF=fastball, CU=curveball, SL=slider, etc.
Spin_rate	Spin of the pitch in rotations per minute. One of the top fields for a feature...the theory is the more spin the harder it is to hit.
Start_speed	The velocity of the pitch in mph (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).

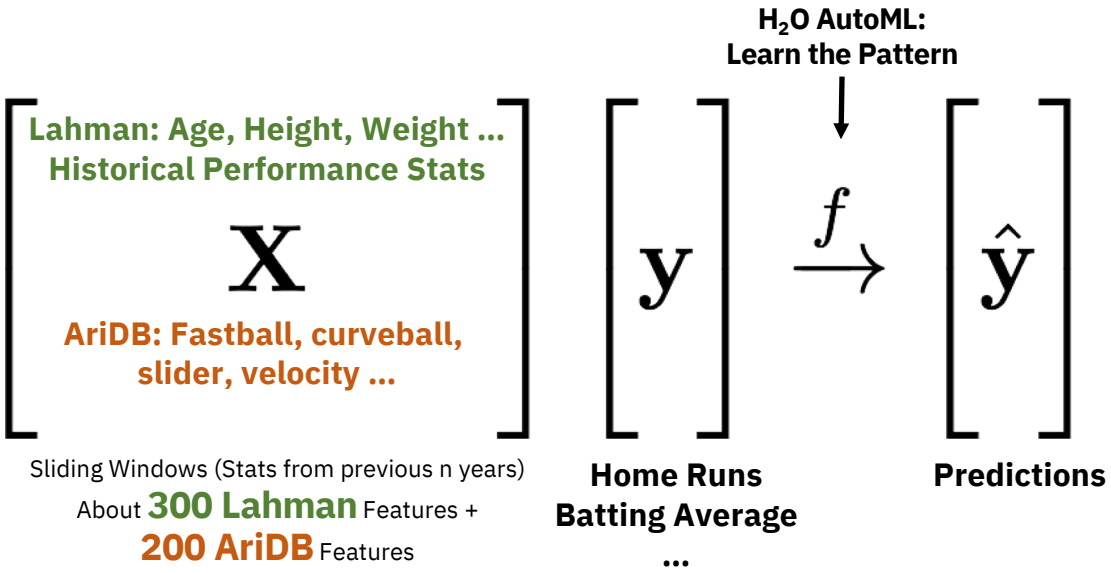

```
# Install 'h2o' from CRAN
install.packages('h2o')
```

Approach One: Learning from **Lahman** only



H2O.ai

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



H2O.ai

Lahman Data

Player's information

birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity
1991	8	7	USA	NJ	Vineland

nameFirst	nameLast	nameGiven	weight	height	bats	throws	debut	finalGame	retroID	bbrefID
Mike	Trout	Michael Nelson	235	74	R	R	2011-07-08	2017-10-01	troum001	troutmi01

Player's past performance (batting in this case)

d_bat_raw

Filter

	playerID	yearID	stint	teamID	lgID	G	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	SO	IBB	HBP	SH	SF	GIDP
95484	troutmi01	2011	1	LAA	AL	40	123	20	27	6	0	5	16	4	0	9	30	0	2	0	1	2
96904	troutmi01	2012	1	LAA	AL	139	559	129	182	27	8	30	83	49	5	67	139	4	6	0	7	7
98308	troutmi01	2013	1	LAA	AL	157	589	109	190	39	9	27	97	33	7	110	136	10	9	0	8	8
99744	troutmi01	2014	1	LAA	AL	157	602	115	173	39	9	36	111	16	2	83	184	6	10	0	10	6
101226	troutmi01	2015	1	LAA	AL	159	575	104	172	32	6	41	90	11	7	92	158	14	10	0	5	11
102712	troutmi01	2016	1	LAA	AL	159	549	123	173	32	5	29	100	30	7	116	137	12	11	0	5	5
104195	troutmi01	2017	1	LAA	AL	114	402	92	123	25	3	33	72	22	4	94	90	15	7	0	4	8

Lahman Data Framed as a ML problem

Player Attributes

yearID	teamID	lgID	weight	height	bats	throws	birthYear	birthCountry	birthState	birthCity	age	career_year
2011	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	20	1
2012	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	21	2
2013	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	22	3
2014	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	23	4
2015	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	24	5
2016	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	25	6
2017	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	26	7
2018	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	27	8
2019	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	28	9
2020	LAA	AL	235	74	R	R	1991	USA	NJ	Vineland	29	10

Past Performance Sliding Windows + Other Stats

last1_HR	last2_HR	last3_HR	last4_HR	last5_HR	avg_last2_HR	avg_last3_HR	avg_last4_HR	avg_last5_HR
NA	NA	NA	NA	NA	NaN	NaN	NaN	NaN
5	NA	NA	NA	NA	5.0	5.00000	5.00000	5.00000
30	5	NA	NA	NA	17.5	17.50000	17.50000	17.50000
27	30	5	NA	NA	28.5	20.66667	20.66667	20.66667
36	27	30	5	NA	31.5	31.00000	24.50000	24.50000
41	36	27	30	5	38.5	34.66667	33.50000	27.80000
29	41	36	27	30	35.0	35.33333	33.25000	32.60000
33	29	41	36	27	31.0	34.33333	34.75000	33.20000
33	33	29	41	36	33.0	31.66667	34.00000	34.40000
33	33	33	29	41	33.0	33.00000	32.00000	33.80000

One of the Targets

yearID	HR
2011	5
2012	30
2013	27
2014	36
2015	41
2016	29
2017	33
2018	NA
2019	NA
2020	NA

Training

Validation

Forecast

No data. Used 2017 value. Not perfect (a quick hack).

H2O.ai

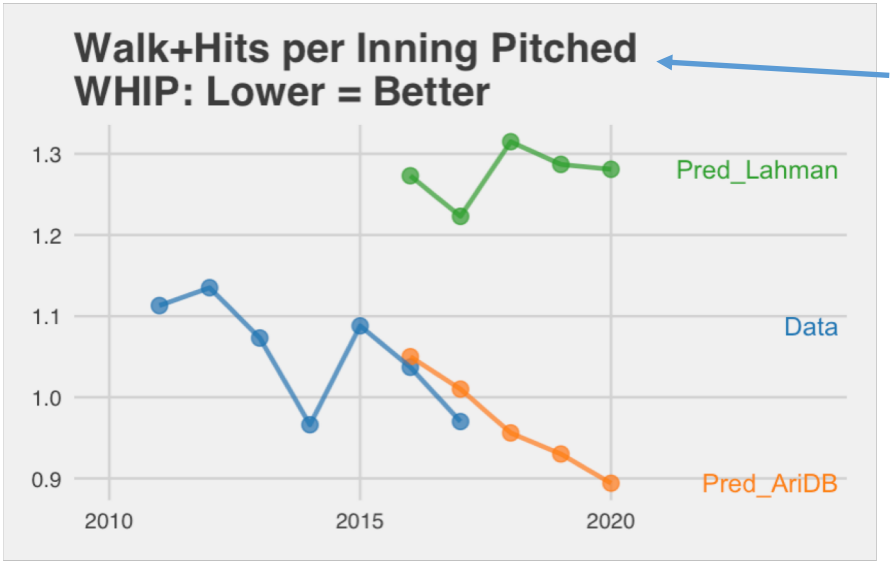
H_2O AutoML Code


```
# H2O AutoML with Lahman only
automl_lahman = h2o.automl(x = features,
                           y = targets[n_target],
                           training_frame = h_train,
                           validation_frame = h_valid,
                           max_models = 10, # increase this to allow more models
                           max_runtime_secs = 120, # increase this to allow more time
                           stopping_metric = "RMSE",
                           stopping_rounds = 3,
                           seed = n_seed,
                           exclude_algos = c("DeepLearning"), # you can exclude any algo
                           project_name = paste0("AutoML_Lahman", targets[n_target]))
```

H_2O AutoML Results


```
H2ORegressionMetrics: stackedensemble
** Reported on cross-validation data. **
** 5-fold cross-validation on training data (Metrics computed for combined holdout predictions) **

MSE:  0.00246453
RMSE:  0.04964404
MAE:  0.03335875
RMSLE:  0.04124294
Mean Residual Deviance :  0.00246453



Slot "leaderboard":
                                             model_id mean_residual_deviance     rmse      mae     rmsle
1 StackedEnsemble_BestOfFamily_0_AutoML_20180615_040834                0.002465 0.049644 0.033359 0.041243
2    StackedEnsemble_AllModels_0_AutoML_20180615_040834                0.002467 0.049669 0.033367 0.041265
3             GLM_grid_0_AutoML_20180615_040834_model_0                0.002480 0.049802 0.033560 0.041401
4             GBM_grid_0_AutoML_20180615_040834_model_4                0.002486 0.049856 0.033707 0.041373
5             GBM_grid_0_AutoML_20180615_040834_model_2                0.002564 0.050638 0.034346 0.042008
6             GBM_grid_0_AutoML_20180615_040834_model_1                0.002569 0.050684 0.034261 0.042022

[12 rows x 5 columns]
```

Predictive Modelling – H_2O AutoML

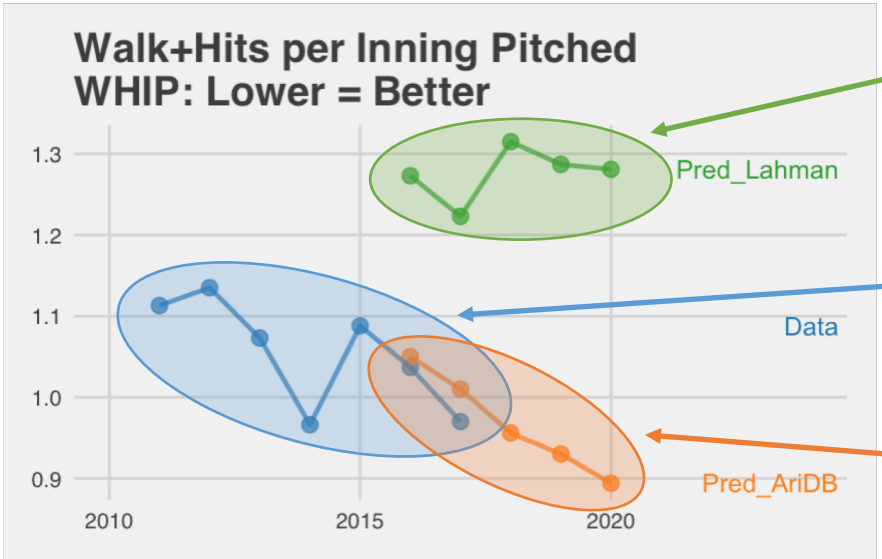


One of Many Targets
(e.g. Home Runs, Batting Average)


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


H2O.ai

Predictive Modelling – H_2O AutoML



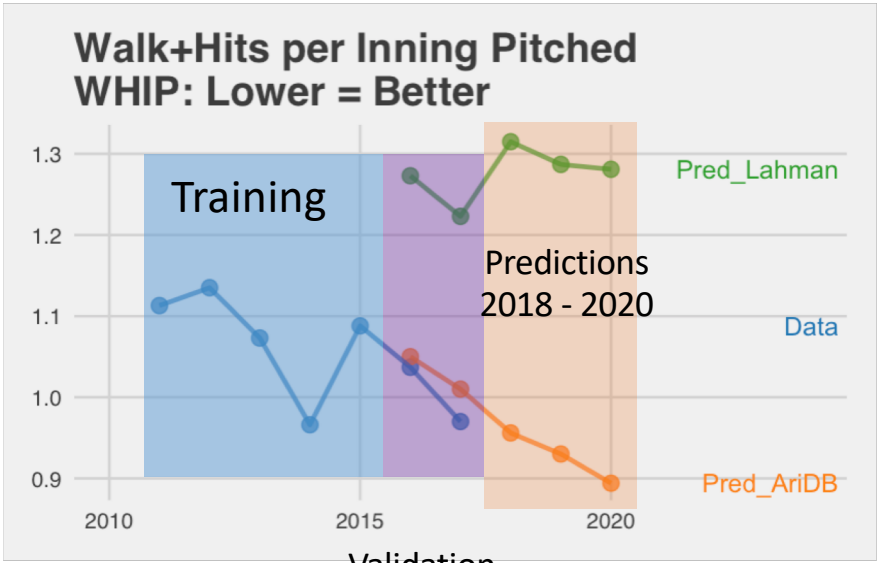
Results from models based on Lahman data only

Historical player performance data

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

H2O.ai

Predictive Modelling – H_2O AutoML



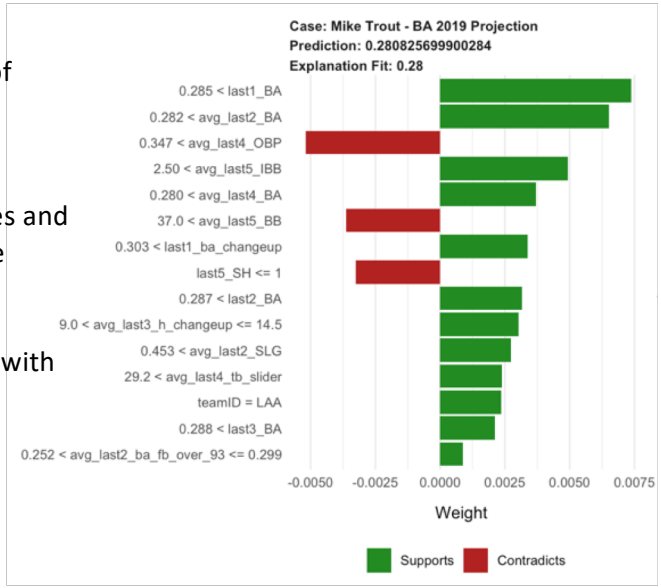
Validation
(2016 & 2017)


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


H2O.ai

Explaining the Predictions
LIME – Local Interpretable Model-agnostic Explanations

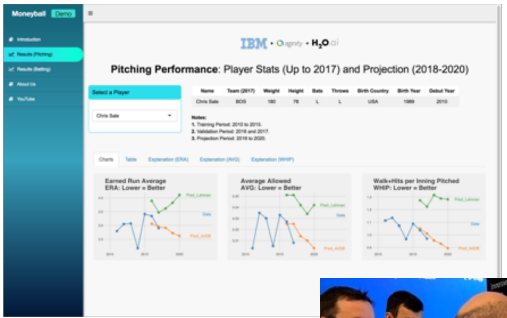
- Approximate reasoning of complex ML models (ensembles).
- Most important attributes and their contributions to the predictions.
- Ari validated the models with his 30+ years of baseball domain knowledge.
- He trusted the models.

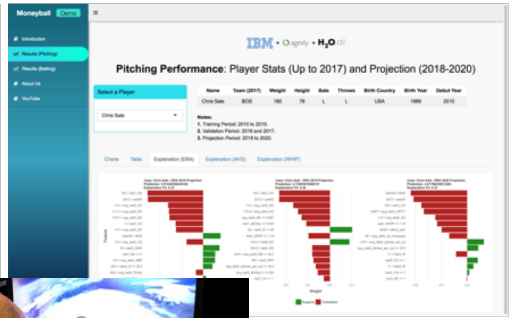



```
# Install 'lime' from CRAN
install.packages('lime')
```

H2O.ai

Putting Everything Together – Moneyball Shiny App



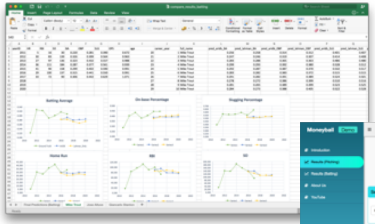


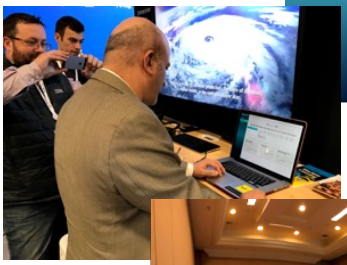
Live Demo
on my laptop

H2O.ai

From Toy Demo to Real Moneyball

- **January to March** – Data munging.
- **March 19** – Joe travelled to Vegas for IBM Think
- **March 19** – AutoML predictions finalized. Initial presentation in Excel.
- **March 20** – Version 1 of Shiny app. Ari used the app to validate some players he had in mind and recommended one player to his team.
- **March 21** – Multimillion-dollar contract finalized.
- **March 22** – Moneyball presentation at IBM Think





H2O.ai

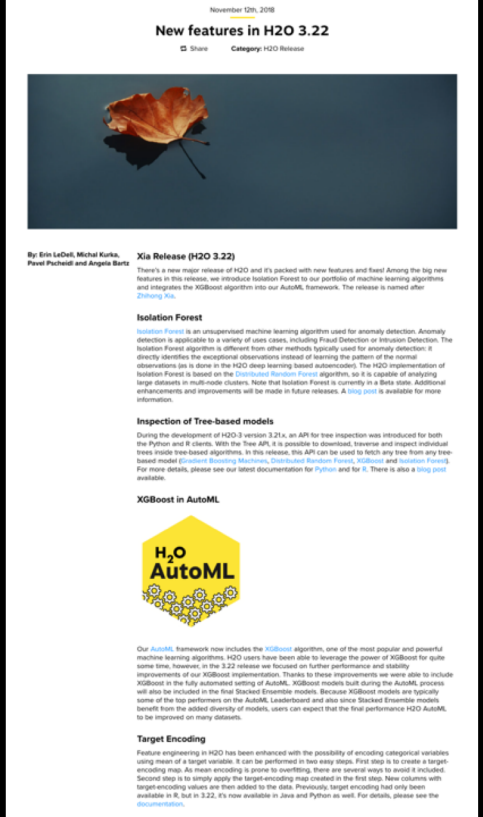
The Moneyball App – Final Project Result

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

$20M

multi-year contract

finalised two weeks
before the regular season



November 12th, 2018

New features in H2O 3.22

Share Category: H2O Release

By: Erin LeDell, Michal Kurka, Pavel Pscheidl and Angela Bartz

Xia Release (H2O 3.22)

There's a new major release of H2O and it's packed with new features and fixes! Among the big new features in this release, we introduce Isolation Forest to our portfolio of machine learning algorithms and integrates the XGBoost algorithm into our AutoML framework. The release is named after Zhihong Xia.

Isolation Forest

Isolation Forest is an unsupervised machine learning algorithm used for anomaly detection. Anomaly detection is applicable to a variety of uses cases, including Fraud Detection or Intrusion Detection. The Isolation Forest algorithm is different from other methods typically used for anomaly detection: it directly identifies the exceptional observations instead of learning the pattern of the normal observations (as is done in the H2O deep learning based autoencoder). The H2O implementation of Isolation Forest is based on the Distributed Random Forest algorithm, so it is capable of analyzing large datasets in multi-node clusters. Note that Isolation Forest is currently in a Beta state. Additional enhancements and improvements will be made in future releases. A blog post is available for more information.

Inspection of Tree-based models

During the development of H2O-3 version 3.21.x, an API for tree inspection was introduced for both the Python and R clients. With the Tree API, it is possible to download, traverse and inspect individual trees inside tree-based algorithms. In this release, this API can be used to fetch any tree from any tree-based model (Gradient Boosting Machines, Distributed Random Forest, XGBoost and Isolation Forest). For more details, please see our latest documentation for Python and for R. There is also a blog post available.

XGBoost in AutoML

H2O AutoML

Our AutoML framework now includes the XGBoost algorithm, one of the most popular and powerful machine learning algorithms. H2O users have been able to leverage the power of XGBoost for quite some time, however, in the 3.22 release we focused on further performance and stability improvements of our XGBoost implementation. Thanks to these improvements we were able to include XGBoost in the fully automated setting of AutoML. XGBoost models built during the AutoML process will also be included in the final Stacked Ensemble models. Because XGBoost models are typically some of the top performers on the AutoML Leaderboard and also since Stacked Ensemble models benefit from the added diversity of models, users can expect that the final performance H2O AutoML to be improved on many datasets.

Target Encoding

Feature engineering in H2O has been enhanced with the possibility of encoding categorical variables using mean of a target variable. It can be performed in two easy steps. First step is to create a target-encoding map. As mean encoding is prone to overfitting, there are several ways to avoid it included. Second step is to simply apply the target-encoding map created in the first step. New columns with target-encoding values are then added to the data. Previously, target encoding had only been available in R, but in 3.22, it's now available in Java and Python as well. For details, please see the documentation.

New Features in H2O

- Isolation Forest
- Tree Inspection
- XGBoost in AutoML
- Target Encoding
- More details -> **www.h2o.ai/blog**

Thanks!

- More Info, Code, and Slides
  - bit.ly/h2o_meetups
  - www.h2o.ai/blog
- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe

H_2O Training Day in Amsterdam

4th December (Tuesday)

Ask Joe for free tickets **joe@h2o.ai**