

# H2O-3 Tutorial



# Contents

H2O.ai

H2O-3とは

H2O-3で利用可能なアルゴリズム

H2O-3 AutoML

Driverless AI と H2O-3

Sparkling Water

Flowによる実施

- インストール
- Flowによる操作

Python Clientによる実施

- インストール
- Pythonでの実行例

R Clientによる実施

- インストール
- Rでの実行例

詳細

- データの前処理
- k分割交差検証 (k Cross-Validation)
- モデルの解釈
- モデルファイル

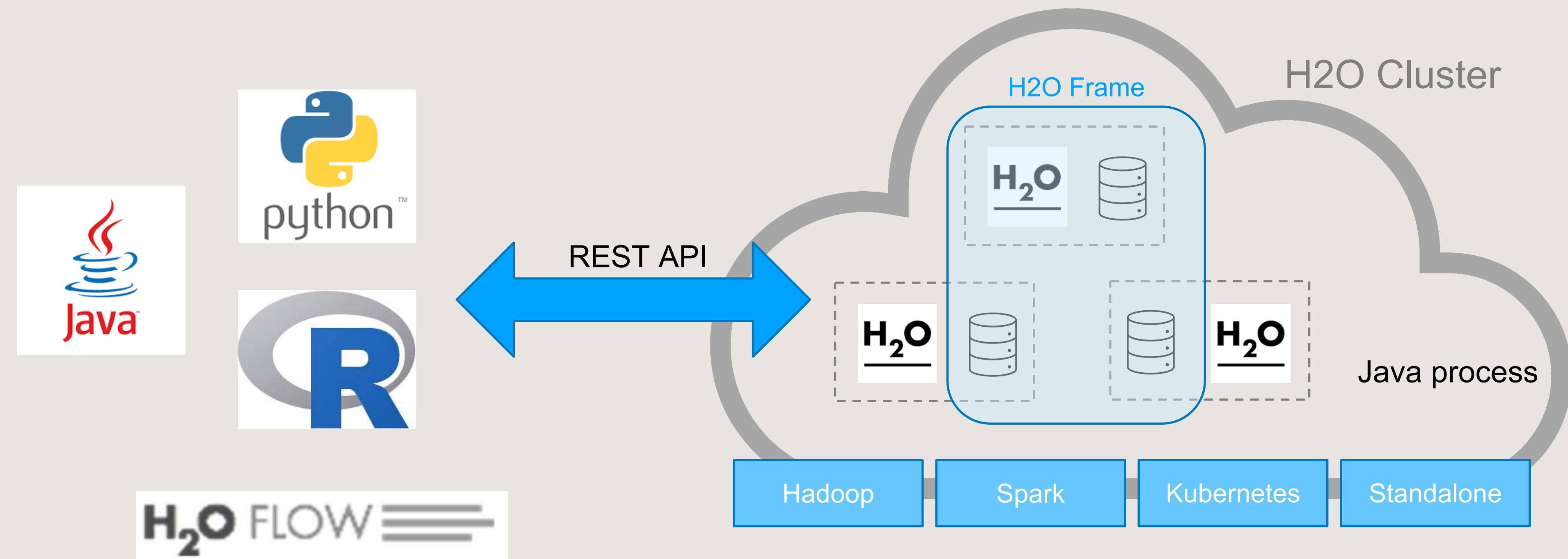
H2O AI Cloud

- AI Cloud上のH2O-3の起動
- Client Access

# H2O-3とは

H2O.ai

- オープンソース、分散型インメモリ機械学習フレームワーク
- さまざまな統計・機械学習モデルとAutoMLのサポート
- R, PythonなどへのAPIの提供と、H2O Flowを用いたGUI操作
- Javaで開発、Key/Value Store方式によるデータやモデルオブジェクトへのアクセス
- HadoopやSpark環境での利用



# H2O-3で利用可能なアルゴリズム

H2O.ai

## 教師あり/Supervised

- AutoML: Automatic Machine Learning
- Cox Proportional Hazards
- Deep Learning (Neural Networks)
- Distributed Random Forest (DRF)
- Generalized Linear Model (GLM)
- Model Selection
- Generalized Additive Models (GAM)
- ANOVA GLM
- Gradient Boosting Machine (GBM)
- Naïve Bayes Classifier
- Rule Fit
- Stacked Ensembles
- Support Vector Machine (SVM)
- Distributed Uplift Random Forest (Uplift DRF)
- XGBoost

## 教師なし/Unsupervised

- Aggregator
- Generalized Low Rank Models (GLRM)
- Isolation Forest
- Extended Isolation Forest
- K-Means Clustering
- Principal Component Analysis (PCA)

## その他

- Target Encoding
- TF-IDF
- Word2vec
- Permutation Variable Importance

# H2O-3 AutoML: Automated Machine Learning



H2O.ai

- 教師あり学習のAutoML
- アルゴリズムと（予め決められた設定の中から）ハイパーパラメータの選択を、ユーザーが指定する時間やモデル数の制約のもと自動実施
- アンサンブル（スタッキング）の実施
- 分類問題における不均衡データの問題への対応
- 特徴量エンジニアリング（Target Encoding）の実施
- Leaderboardで全ての作成したモデルの精度を確認

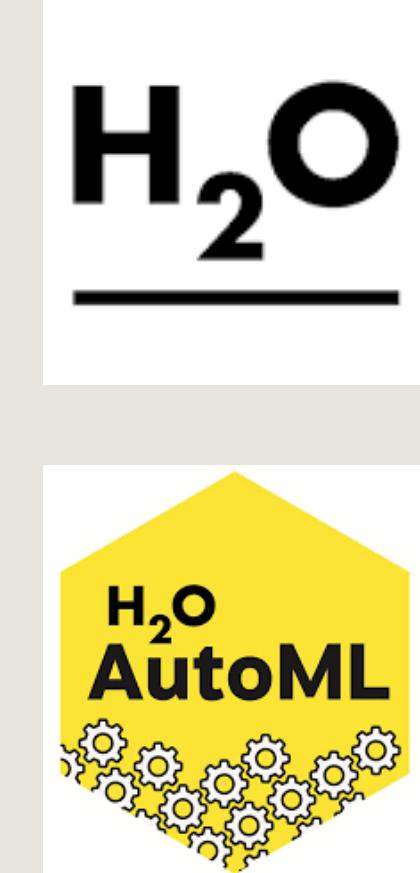
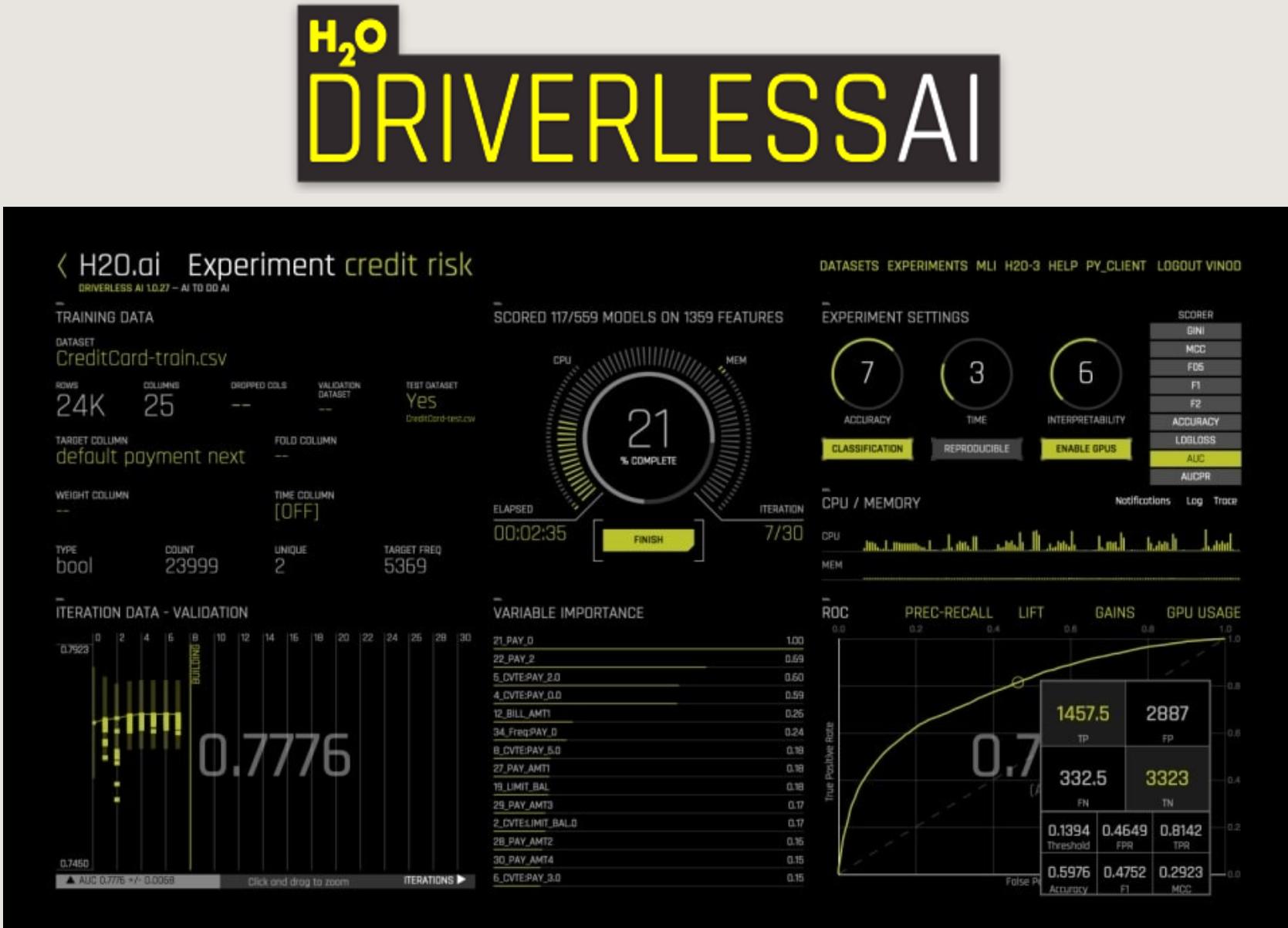
Leaderboard

models sorted in order of auc, best first					
	model_id	auc	logloss	aucpr	mean_perce
0	StackedEnsemble_AllModels_1_AutoML_1_20220615_150727	0.7861919322979182	0.42453562000117034	0.5658311432673605	0.28578
1	StackedEnsemble_AllModels_2_AutoML_1_20220615_150727	0.7860907671436528	0.42450874457257465	0.5661821125350466	0.28636
2	StackedEnsemble_AllModels_5_AutoML_1_20220615_150727	0.7859633254934691	0.4246185197519597	0.5662448231590076	0.28643
3	StackedEnsemble_BestOfFamily_6_AutoML_1_20220615_150727	0.7844598668255073	0.42561077996471725	0.5641915933014441	0.28983
4	StackedEnsemble_BestOfFamily_3_AutoML_1_20220615_150727	0.7843800343934435	0.4256263428398944	0.5638664889238088	0.28857
5	StackedEnsemble_BestOfFamily_2_AutoML_1_20220615_150727	0.7838256222588983	0.4261597208759282	0.5621047380727804	0.28883
6	GBM_3_AutoML_1_20220615_150727	0.7832553081359669	0.4270795778124074	0.5590650313824038	0.28887
7	GBM_2_AutoML_1_20220615_150727	0.7832227237330757	0.42660893557995333	0.5625063612090158	0.28844
8	StackedEnsemble_AllModels_4_AutoML_1_20220615_150727	0.7831084590296669	0.42641427375019	0.5574691241282507	0.28722
9	StackedEnsemble_BestOfFamily_1_AutoML_1_20220615_150727	0.782791612475425	0.4269769457437145	0.5588712872669337	0.28724
10	GBM_1_AutoML_1_20220615_150727	0.7826634129734322	0.4271107086322563	0.5587513670645117	0.28646
11	StackedEnsemble_BestOfFamily_5_AutoML_1_20220615_150727	0.7814353092793878	0.4279296623194157	0.5551481044968649	0.29145
12	GBM_4_AutoML_1_20220615_150727	0.779269472005741	0.42982941378017436	0.5523103867267611	0.29183
13	XGBoost_3_AutoML_1_20220615_150727	0.7772857320097718	0.4312491181667432	0.5527516693085847	0.29326
14	StackedEnsemble_AllModels_3_AutoML_1_20220615_150727	0.7743072454038448	0.4364886059888557	0.5445341396277963	0.29865
15	StackedEnsemble_BestOfFamily_4_AutoML_1_20220615_150727	0.7736633745069093	0.4365075106840971	0.5421840106666854	0.29245

AutoMLの実行結果が、予測精度の高いモデルの順で表示される

# Driverless AI と H2O-3

H2O.ai

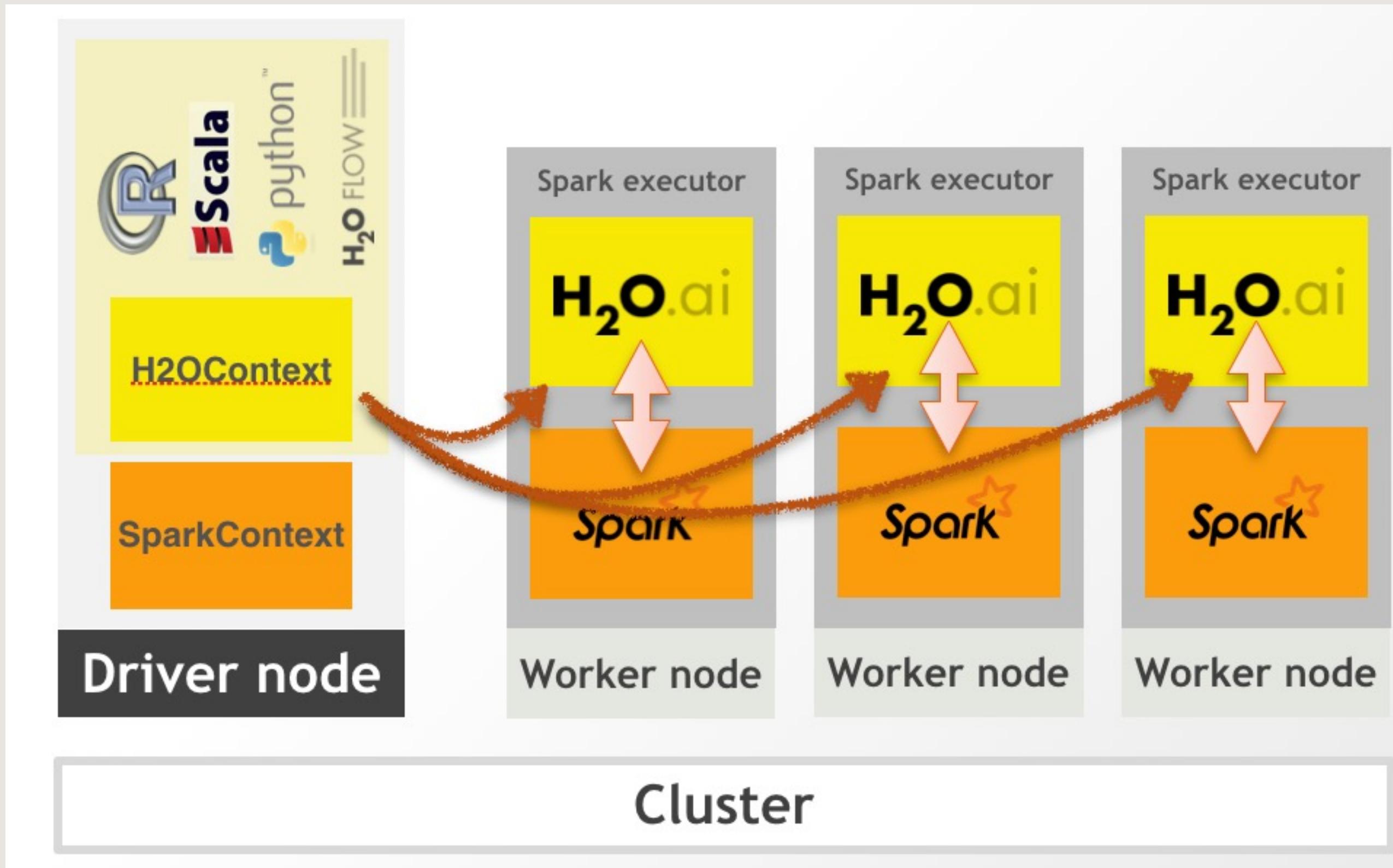


- 最高水準のAutoML
- モデル作成の大部分（アルゴリズムの選択、特徴量エンジニアリング、ハイパー パラメータチューニング、精度評価）を自動化
- 特徴量エンジニアリングのバリエーション
- 時系列、画像、テキストデータへの対応
- 機械学習の解釈可能性（Machine Learning Interpretability (MLI)）
- GUI操作
- Python/R APIの提供

- オープンソース
- AutoML含む機械学習フレームワーク
- Python/R APIによるフルコントロール
- GUI (Flow) の提供
- 分散処理による大容量データへの対応 (Hadoop、Spark)
- さまざまなアルゴリズム（教師あり/なし、その他補助的な機能）の実装
- データ加工機能

# Sparkling Water

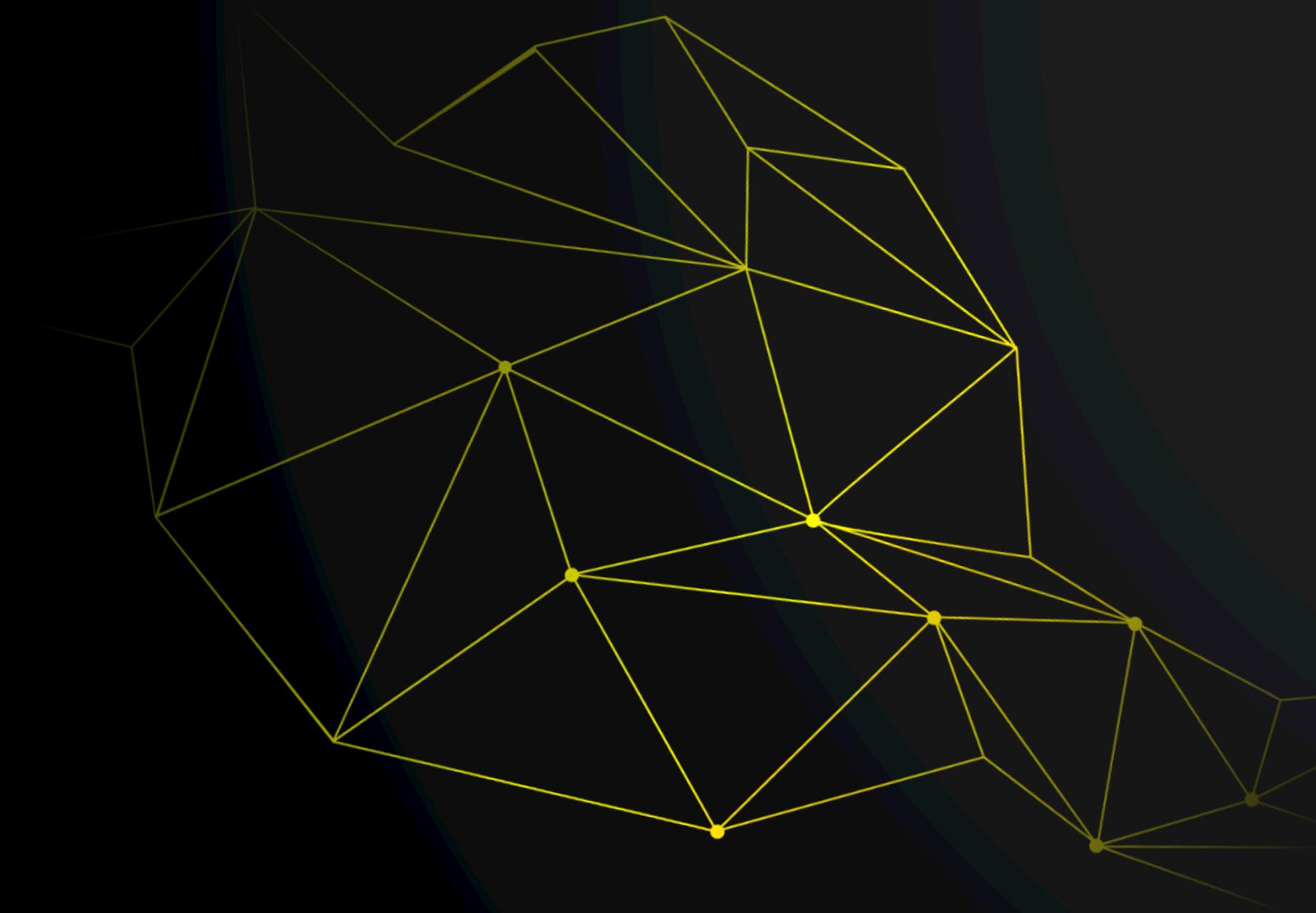
Sparkクラスター上でH2O-3を実行



- ✓ Flowによる操作
- ✓ H2O-3と同じプログラム（Python, R）での動作
- ✓ Spark DataFrame、H2O Frameの変換

# Flow

- GUIによるH2O-3の操作



# インストール

H2O.ai

Download : <https://h2o.ai/resources/download/> ("H2O Open Source Platform"よりバージョンを選択)

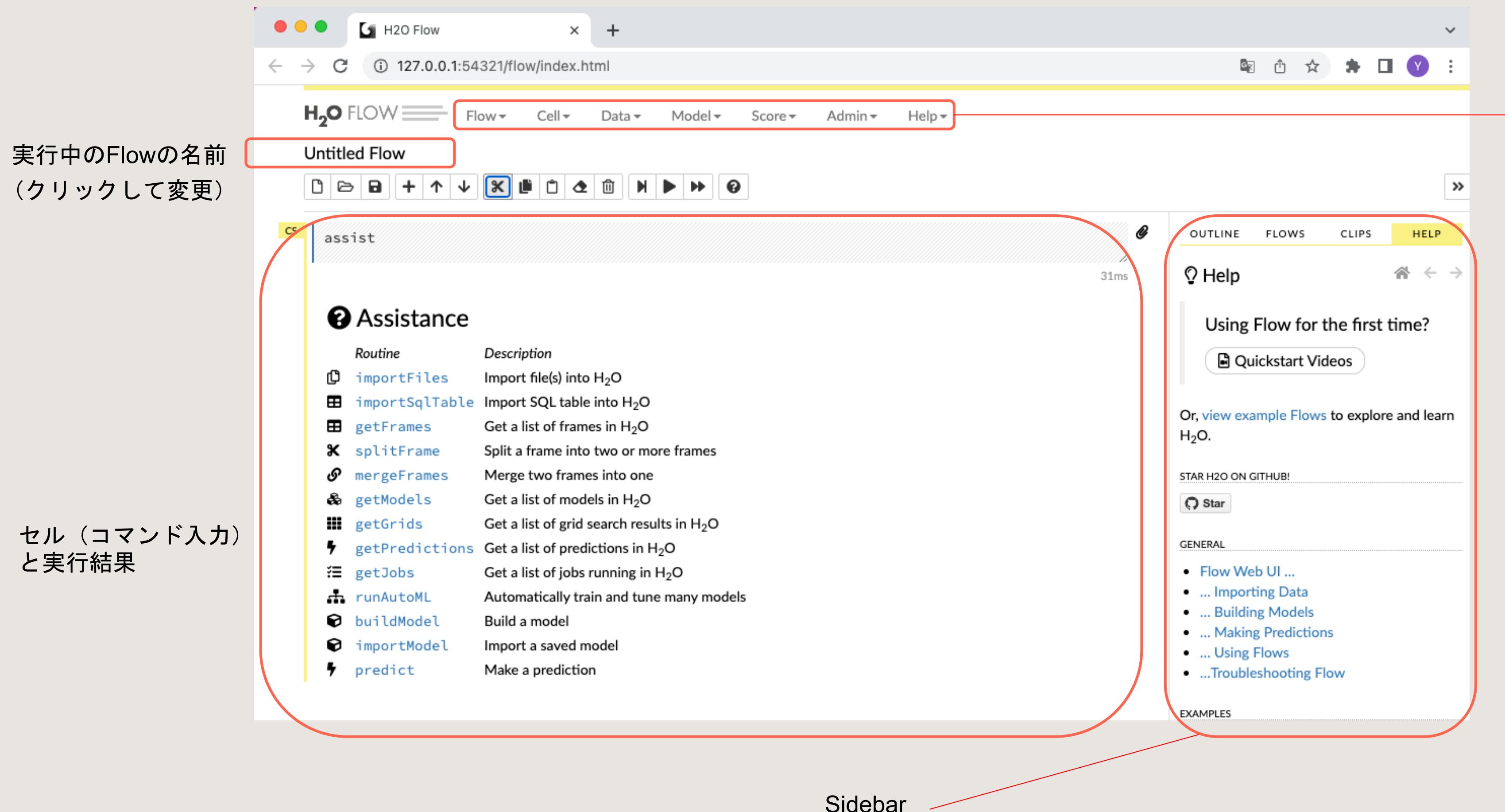
The screenshot shows the H2O download page. At the top, the H2O logo and version 3.36.1.2 are displayed. Below the logo, a dark banner contains the text "Fast Scalable Machine Learning API" and "For Smarter Applications". A navigation bar below the banner includes links for "INSTALL IN R", "INSTALL IN PYTHON", "INSTALL ON HADOOP", "USE FROM MAVEN", and "KUBERNETES". The "DOWNLOAD AND RUN" button is highlighted with a red box. In the center, there is a large "DOWNLOAD H2O" button with a red box around it. Below these buttons, the text "Get started with H2O in 3 easy steps" is followed by three numbered steps: 1. Download H2O. This is a zip file that contains everything you need to get started. 2. From your terminal, run: 

```
cd ~/Downloads  
unzip h2o-3.36.1.2.zip  
cd h2o-3.36.1.2  
java -jar h2o.jar
```

 3. Point your browser to <http://localhost:54321>. To the right of the terminal command, there is a clipboard icon.

- DL後、フォルダを解凍し、jarファイルを実行（\$java -jar h2o.jar）（Javaが必要）
- H2O-3の開始後、<http://localhost:54321>、もしくは実行サーバのアドレスへブラウザからアクセス

## マウス操作で、機械学習モデルの作成を実施



### メニュー

#### Flow

- Flowの保存やロード

#### Cell

- コピーや貼り付けなどセルに対するアクション

#### Data

- データのロードや分割など、データに対するアクション

#### Model

- 当てはめるモデルの選択や、学習済みモデルのエクスポート

#### Score

- スコアリングの実施

#### Admin

- H2O Clusterやメモリ、CPUの利用状況

#### Help

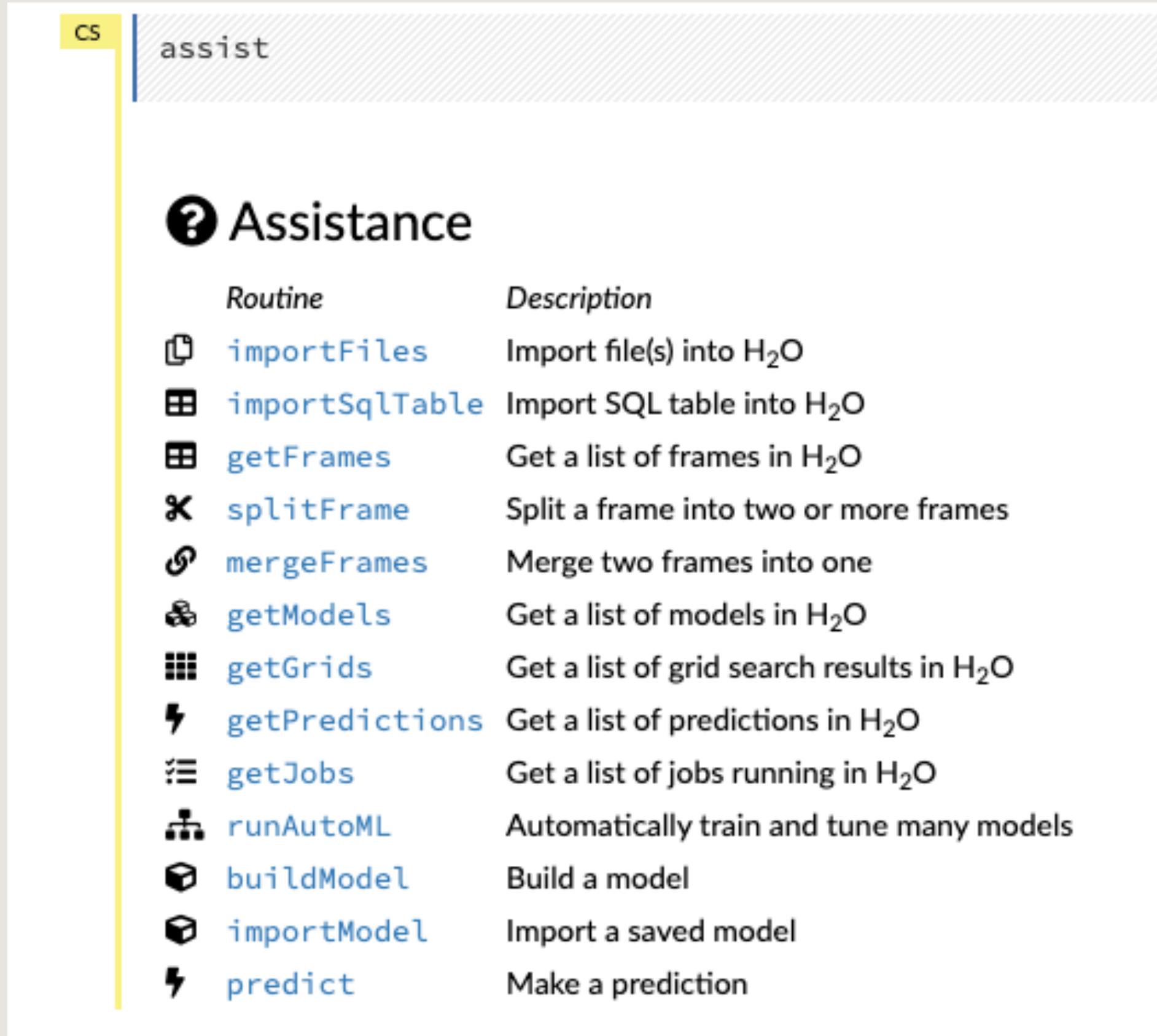
- キーボードショートカットやドキュメンテーションへのリンク

# Assistance

H2O.ai

セルに“assist”と入力し、実行

Flowで実施する主な操作をアシスト（コマンド入力なしで操作を実行）



## importFiles

- H2O-3実行環境のデータパス、もしくはデータへのURLを指定し、データのインポートを実施
- ローカルからサーバへデータをアップロードする場合は、“Data”メニュー>”Upload File..”から実施

## getFrames

- インポートされているデータと、モデル学習過程で作成されたデータの一覧

## splitFrame

- インポートされているデータの学習/テストの分割

## getModels

- 学習済みモデルの一覧

## getGrids

- グリッドサーチを実施したモデルの一覧

## getPredictions

- 各学習済みモデルの予測結果（精度）

## runAutoML

- AutoMLによるモデル作成

## buildModel

- アルゴリズムを選択してモデルを作成

## importModel

- モデルオブジェクト（MOJO）のインポート

## predict

- 学習済みモデルとデータを選択して予測を実施

# Import/Upload データ

データをH2O Clusterへ読み込み

CS importFiles 14ms

**Import Files**

Search: /Users/tmp/data/

Search Results: Found 3 files: [Add all](#)

- + /Users/tmp/data/BostonHousing.csv
- + /Users/tmp/data/UCI\_Credit\_Card3.csv
- /Users/tmp/data/TitanicData2.csv

Selected Files: 1 file selected: [Clear All](#)

Actions: [Import](#)

データへのローカルパスやURLを指定

パス上のデータ  
読み込むデータの「+」ボタンを選択

読み込みを実施するデータ

- ✓ ローカルのH2O-3クライアントから、サーバのH2O-3へデータをアップロードする場合は、"Data"メニュー>"Upload File.."から実施

\* 現在接続しているH2O-3プロセスを終了すると、データも消去される

# データParse（解析）

読み込んだデータの設定やカラムの定義を実施

CS | setupParse source\_frames: [ "nfs://Users/tmp/data/TitanicData2.csv" ] 90ms

### Setup Parse

**PARSE CONFIGURATION**

Sources: nfs://Users/tmp/data/TitanicData2.csv  
ID: TitanicData2.hex

Parser: CSV  
Separator: ;'044'

Escape Character: 0

Column Headers:  Auto  First row contains column names  First row contains data

Options:  Enable single quotes as a field quotation character  Delete on done

**EDIT COLUMN NAMES AND TYPES**

Search by column name...

Column	Name	Type	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7
1	Passenger_	Numeric	1	2	3	4	5	6	7
2	pclass	Enum	3rd	3rd	2nd	3rd	3rd	2nd	3rd
3	survived	Numeric	0	1	1	0	1	0	0
4	name_with_	String	Andersson, Mr. Anders Johan	McGowan, Miss. Anna ""Annie""	Caldwell, Mr. Albert Francis	Augustsson, Mr. Albert	Lindqvist, Mr. Eino William	Carter, Mrs. Ernest	Sa (Lilian Hughes)
5	name_witho	String	Andersson, Anders Johan	McGowan, Anna ""Annie""	Caldwell, Albert	Augustsson, Albert	Lindqvist, Eino William	Carter, Ernest	Sa (A Courtenay Bu)
6	sex	Enum	male	female	male	male	male	female	female

データのフォーマット等の指定

カラムのデータ型の指定

# データの型

H2O.ai

## 各カラムのデータ型の定義

EDIT COLUMN NAMES AND TYPES							
Search by column name...							
1	PassengerId	Numeric	1	2	3	4	5
2	pclass	Unknown	3rd	3rd	2nd	3rd	3rd
3	survived	Numeric	0	1	1	0	1
4	name_with_	Enum	Andersson, Mr. Anders Johan	McGowan, Miss. Anna ""Annie""	Caldwell, Mr. Albert Francis	Augustsson, Mr. Albert	Lindqvist, Mr. Eino William
		Time	Anders	Anna	Albert	Albert	Eino
		UUID	Johan	""Annie""	Francis		William
		String					(Lilian Hughes)
		Invalid					
5	name_withou	String	Andersson, Anders Johan	McGowan, Anna ""Annie""	Caldwell, Albert Francis	Augustsson, Albert	Lindqvist, Eino William
			Anders	Anna	Albert	Albert	Eino
			Johan	""Annie""	Francis		William
							(Lilian Hughes)
6	sex	Enum	male	female	male	male	female
7	age	Numeric	39.0	15.0	26.0	23.0	20.0
8	sibsp	Numeric	1	0	1	0	1

**Numeric** : 数値変数

**Enum** : カテゴリカル変数

**Time** : 時間を示すカラム

**UUID** : IDカラム

**String** : テキストデータ

- ✓ 教師あり学習のアルゴリズムに投入する場合、特徴量は数値 (Numeric) 、もしくはカテゴリカル変数 (Enum) に指定しておく
- ✓ ターゲット変数が0/1であり、分類問題として扱いたい場合はEnumとなっていることを確認

# データに対するアクション

データParse後、データのサマリと、データに対するアクションが表示される

The screenshot shows the H2O AI interface with the following details:

- Code Snippet:** CS | getFrameSummary "UCI\_Credit\_Card3.hex"
- Time:** 105ms
- Title:** UCI\_Credit\_Card3.hex
- Actions:** View Data, Split, Build Model, Run AutoML, Predict, Download, Export, Delete (highlighted with a red border)
- Summary Table:**

	Rows	Columns	Compressed Size
	30000	25	2MB
- Column Summaries:** A table showing statistics for each column, including label, type, missing values, and actions like 'Convert to enum'.

label	type	Missing	Zeros	+Inf	-Inf	min	max	mean	sigma	cardinality	Actions
ID	int	0	0	0	0	1.0	30000.0	15000.5000	8660.3984	·	<a href="#">Convert to enum</a>
LIMIT_BAL	int	0	0	0	0	10000.0	1000000.0	167484.3227	129747.6616	·	<a href="#">Convert to enum</a>
SEX	enum	0	11888	0	0	0	1.0	0.6037	0.4891	2	<a href="#">Convert to numeric</a>
EDUCATION	enum	345	10585	0	0	0	3.0	·	·	4	<a href="#">Convert to numeric</a>
MARRIAGE	enum	54	13659	0	0	0	2.0	·	·	3	<a href="#">Convert to numeric</a>
AGE	int	0	0	0	0	21.0	79.0	35.4855	9.2179	·	<a href="#">Convert to enum</a>
PAY_1	int	0	14737	0	0	-2.0	8.0	-0.0167	1.1238	·	<a href="#">Convert to enum</a>
PAY_2	int	0	15730	0	0	-2.0	8.0	-0.1338	1.1972	·	<a href="#">Convert to enum</a>
PAY_3	int	0	15764	0	0	-2.0	8.0	-0.1662	1.1969	·	<a href="#">Convert to enum</a>
PAY_4	int	0	16455	0	0	-2.0	8.0	-0.2207	1.1691	·	<a href="#">Convert to enum</a>
DAY_F	int	0	16947	0	0	-2.0	8.0	-0.2662	1.1222	·	<a href="#">Convert to enum</a>

各カラムの一変量集計を実施したい場合、  
カラム名をクリック

データ型（数値（Numeric） カテゴリカル（Enum））の変更

# データの分割 (Split)

学習/テストへのデータ分割

AssistanceのsplitFrameや、データのサマリのSplitから実施

The screenshot shows the H2O AI interface with two main sections:

- Split Frame Section:**
  - Frame: UCI\_Credit\_Card3.hex
  - Splits: Ratio
 

0.75	Key: frame_0.750
0.250	Key: frame_0.250
  - Add a new split
  - Seed: 765999
- Split Frames Section:**
  - Type Key
 

frame_0.750
frame_0.250
  - Ratio
 

0.75
0.25

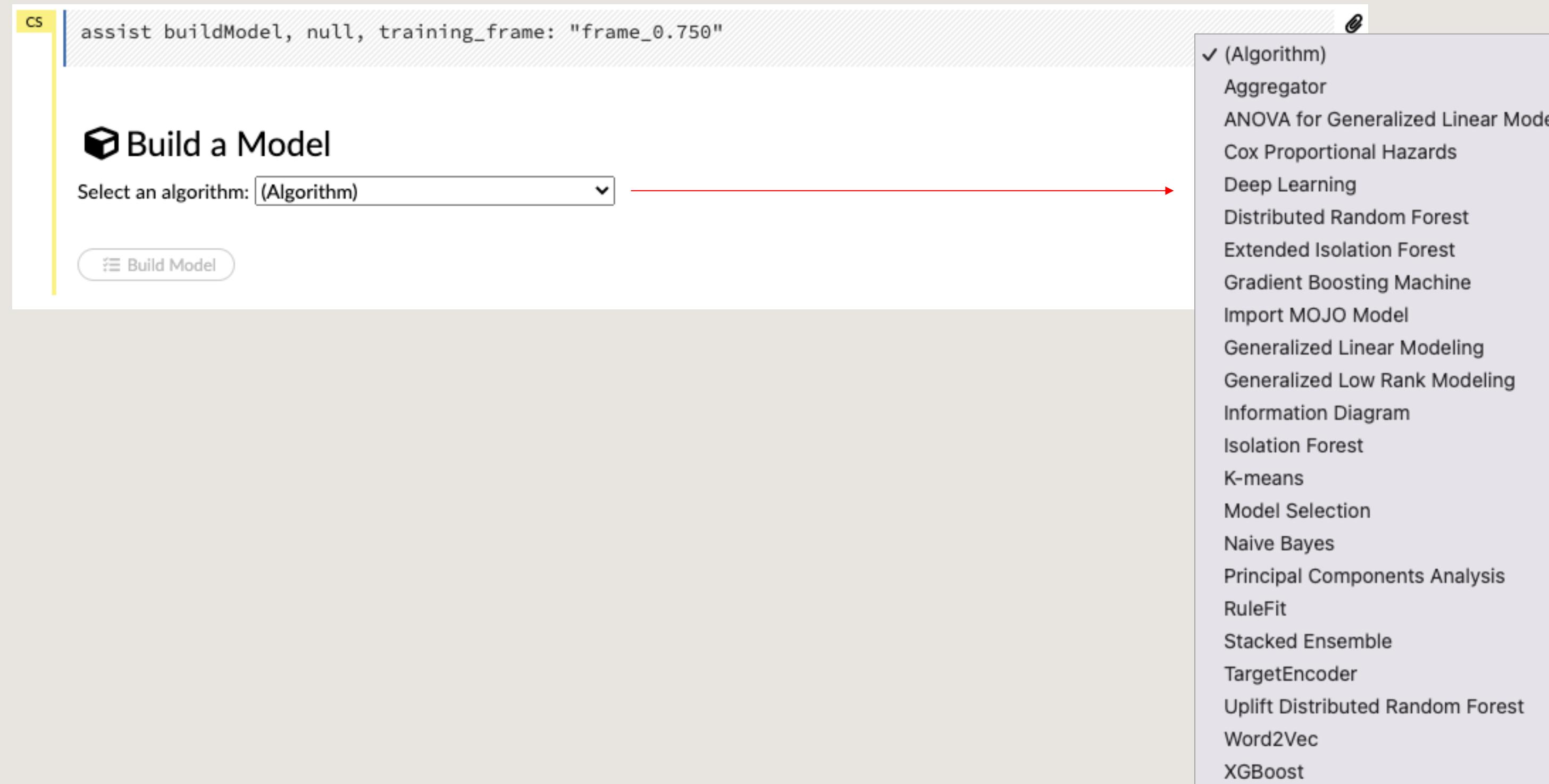
データの分割割合と分割後のデータ名  
分割数を増やす場合は“Add a new split”を選択

分割の実施後のデータが表示  
データ名をクリックすると、データのサマリと、データに対するアクションが表示される

# モデルの作成 (Build Model)

モデルの作成

データのサマリのBuild Modelから実施



アルゴリズム以外の機能

## Import MOJO Model

- MOJO（保存済みのモデルオブジェクトファイル）をインポートし、Flowで利用できるようにする

## Information Diagram

- <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/admissible.html>

# モデルの作成 (Build Model)

## モデル学習の設定

CS | assist buildModel, null, training\_frame: "frame\_0.750"

12ms

### Build a Model

Select an algorithm: Gradient Boosting Machine

**PARAMETERS**

model_id	gbm-ffb7dfdd-f0ef-4d13-bcb9-4597b446c08a	Destination id for this model; auto-generated if not specified.
training_frame	frame_0.750	Id of the training data frame.
validation_frame	(Choose...)	Id of the validation data frame.
nfold	0	Number of folds for K-fold cross-validation (0 to disable or >= 2).
response_column	(Choose...)	Response variable column.
ignored_columns	Search...	Names of columns to ignore for training.

Showing page 1 of 3. -25 ignored.

<input type="checkbox"/> ID	INT
<input type="checkbox"/> LIMIT_BAL	INT
<input type="checkbox"/> SEX	ENUM(2)
<input type="checkbox"/> EDUCATION	ENUM(4) 1% NA
<input type="checkbox"/> MARRIAGE	ENUM(3) 0% NA
<input type="checkbox"/> AGE	INT
<input type="checkbox"/> PAY_1	INT
<input type="checkbox"/> PAY_2	INT
<input type="checkbox"/> PAY_3	INT
<input type="checkbox"/> PAY_4	INT

All     None    ← Previous 10    → Next 10

Only show columns with more than 0 % missing values.

ignore\_const\_cols

Ignore constant columns.

### 代表的な設定

#### データに関する設定

##### training\_frame

- 学習データ。必須

##### validation\_frame

- 検証データ。指定すると検証データに対する予測結果が計算される

nfolds: (GLM, GBM, DL, DRF)

- 検証データを指定せず、k分割交差検証を実施する場合

##### response\_column

- ターゲット変数。教師ありモデルの場合、必須

##### ignored\_columns

- 用いない特徴量

#### ハイパーパラメータに関する設定

- 各アルゴリズムに対応したハイパーパラメータを設定

# ハイパーパラメータチューニング

グリッドサーチによるハイパーパラメータチューニングの実施が可能

**PARAMETERS**

<b>model_id</b>	gbm-f5a88346-b709-4e44-a0b8-5f1e3d31022f	Destination id for this model; auto-generated if not specified.																				
<b>training_frame</b>	UCI_Credit_Card3.hex	Id of the training data frame.																				
<b>validation_frame</b>	(Choose...)	Id of the validation data frame.																				
<b>nfold</b>	3	Number of folds for K-fold cross-validation (0 to disable or >= 2).																				
<b>response_column</b>	default_payment_next_month	Response variable column.																				
<b>ignored_columns</b>	Search...	Names of columns to ignore for training.																				
Showing page 1 of 3. -24 ignored. <table border="1"> <tr><td><input checked="" type="checkbox"/> ID</td><td>INT</td></tr> <tr><td><input type="checkbox"/> LIMIT_BAL</td><td>INT</td></tr> <tr><td><input type="checkbox"/> SEX</td><td>ENUM(2)</td></tr> <tr><td><input type="checkbox"/> EDUCATION</td><td>ENUM(4) 1% NA</td></tr> <tr><td><input type="checkbox"/> MARRIAGE</td><td>ENUM(3) 0% NA</td></tr> <tr><td><input type="checkbox"/> AGE</td><td>INT</td></tr> <tr><td><input type="checkbox"/> PAY_1</td><td>INT</td></tr> <tr><td><input type="checkbox"/> PAY_2</td><td>INT</td></tr> <tr><td><input type="checkbox"/> PAY_3</td><td>INT</td></tr> <tr><td><input type="checkbox"/> PAY_4</td><td>INT</td></tr> </table> <input checked="" type="checkbox"/> All <input type="checkbox"/> None			<input checked="" type="checkbox"/> ID	INT	<input type="checkbox"/> LIMIT_BAL	INT	<input type="checkbox"/> SEX	ENUM(2)	<input type="checkbox"/> EDUCATION	ENUM(4) 1% NA	<input type="checkbox"/> MARRIAGE	ENUM(3) 0% NA	<input type="checkbox"/> AGE	INT	<input type="checkbox"/> PAY_1	INT	<input type="checkbox"/> PAY_2	INT	<input type="checkbox"/> PAY_3	INT	<input type="checkbox"/> PAY_4	INT
<input checked="" type="checkbox"/> ID	INT																					
<input type="checkbox"/> LIMIT_BAL	INT																					
<input type="checkbox"/> SEX	ENUM(2)																					
<input type="checkbox"/> EDUCATION	ENUM(4) 1% NA																					
<input type="checkbox"/> MARRIAGE	ENUM(3) 0% NA																					
<input type="checkbox"/> AGE	INT																					
<input type="checkbox"/> PAY_1	INT																					
<input type="checkbox"/> PAY_2	INT																					
<input type="checkbox"/> PAY_3	INT																					
<input type="checkbox"/> PAY_4	INT																					
Only show columns with more than 0 % missing values.																						
<b>ignore_const_cols</b>	<input checked="" type="checkbox"/>	Ignore constant columns.																				
<b>ntrees</b>	30;50;70	Number of trees.																				
<b>max_depth</b>	5;7	Maximum tree depth (0 for unlimited).																				
<b>min_rows</b>	10	Fewest allowed (weighted) observations in a leaf.																				
<b>nbins</b>	20	For numerical columns (real/int), build a histogram of (at																				

## GRID?

- チューニングを実施したいハイパーパラメータの横にチェックを付ける

## グリッドサーチ実施例

「;」で値（探索対象のハイパーパラメータの値）を区切る

## モデル作成後、結果の確認

getModel "gbm-f0764573-0907-4a72-a473-61e892d72da0"

**Model**

Model ID: gbm-f0764573-0907-4a72-a473-61e892d72da0

Algorithm: Gradient Boosting Machine

Actions: Refresh, Predict..., Download POJO, Download Model Deployment Package (MOJO), Export, Inspect, Delete, Download Gen Model

**MODEL PARAMETERS**

Parameter	Value	Description
model_id	gbm-f0764573-0907-4a72-a473-61e892d72da0	Destination id for this model; auto-generated if not specified.
training_frame	frame_0.750	Id of the training data frame.
validation_frame	frame_0.250	Id of the validation data frame.
fold_assignment		Cross-validation fold assignment scheme, if fold_column is not specified. The 'Stratified' option will stratify the folds based on the response variable, for classification problems.
response_column	default_payment_next_month	Response variable column.
ignored_columns	ID	Names of columns to ignore for training.
r2_stopping	1.7976931348623157e+308	r2_stopping is no longer supported and will be ignored if set - please use stopping_rounds, stopping_metric and stopping_tolerance instead. Previous version of H2O would stop making trees when the R^2 metric equals or exceeds this
stopping_metric		Metric to use for early stopping (AUTO: logloss for classification, deviance for regression and anomaly_score for Isolation Forest). Note that custom and custom_increasing can only be used in GBM and DRF with the Python client.
seed	-5099368648791651706	Seed for pseudo random number generator (if applicable)
distribution	bernoulli	Distribution function
histogram_type	UniformAdaptive	What type of histogram to use for finding optimal split points

結果には、モデルのハイパーパラメータや、学習とテストデータに対する予測精度、変数の重要度などが表示される

### 結果に対するアクション

#### Predict

- 新規データセットに対する予測（新規データセットにターゲット変数が含まれる場合は、精度の評価も実施）

#### Inspect

- モデルの設定や結果をテーブル形式で表示

#### Download Model Deployment Package (MOJO)

- MOJO形式のモデルのダウンロード

#### Download POJO

- POJO形式のモデルのダウンロード

#### Download Gen Model

- MOJO, POJOのランタイムのダウンロード

#### Export

- バイナリファイルとしてモデルを保存

## AutoMLの実施と結果 (Leaderboard)

The screenshot shows the 'Run AutoML' configuration page. At the top, there is a code input field containing 'assist runAutoML, training\_frame: "UCI\_Credit\_Card3.hex"'. Below it is a large 'Run AutoML' button. The configuration area is titled 'PARAMETERS' and contains the following fields:

- project\_name**: A text input field with a placeholder '(Choose...)'. Description: 'Optional project name used to group models from multiple AutoML runs into a single Leaderboard; derived from the training data name if not specified.'
- training\_frame\***: A dropdown menu set to 'UCI\_Credit\_Card3.hex'.
- response\_column\***: A dropdown menu set to 'default\_payment\_next\_month'.
- validation\_frame**: A dropdown menu with '(Choose...)' selected.
- blending\_frame**: A dropdown menu with '(Choose...)' selected.

On the right side of the configuration area, there are two small explanatory text blocks:

- 'validation\_frame or nfolds': '検証データでの検証を実施する場合'
- 'sort\_metric': 'Leaderboardの表示順のモデル評価指標'

### 代表的な設定

#### validation\_frame or nfolds

- 検証データでの検証を実施する場合

#### sort\_metric

- Leaderboardの表示順のモデル評価指標

#### max\_models

- 実施時間に制約をかける場合。モデル数

#### max\_runtime\_secs

- 実施時間に制約をかける場合。実施時間

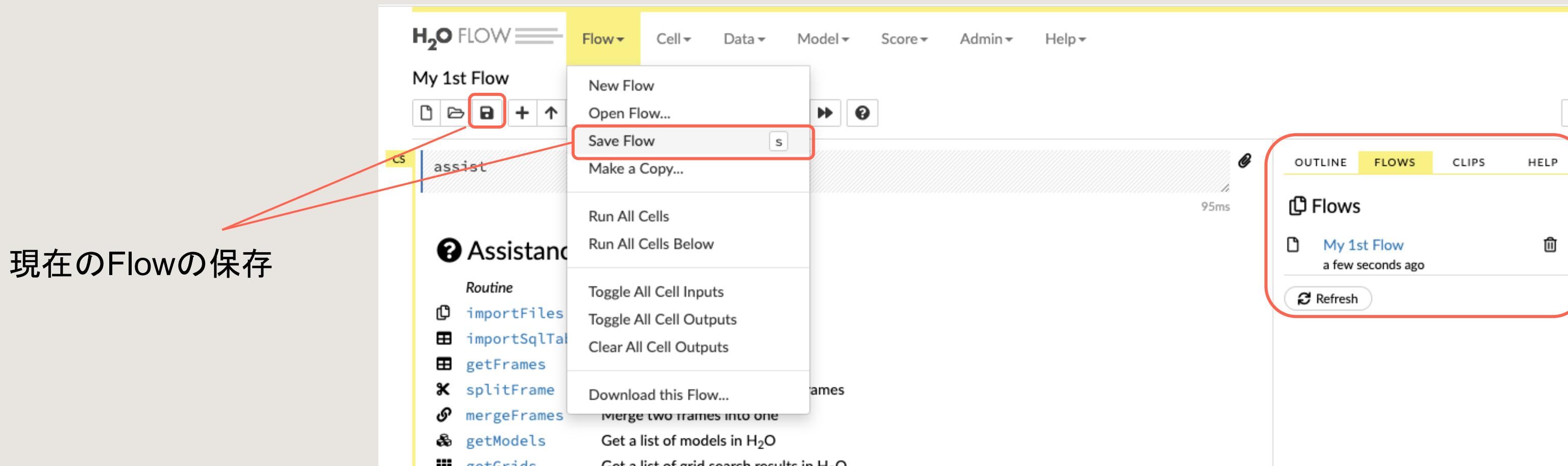
The screenshot shows the 'Leaderboard' results page. At the top, there is a 'Monitor Live' button. The main area is titled 'MODELS' and displays a table of model results sorted by aucpr, best first. The table has the following columns:

model_id	aucpr	auc	logloss	mean_per_class_error	rmse	mse
0 StackedEnsemble_AllModels_3_AutoML_2_20220627_94633	0.5623313877975674	0.7800975911896315	0.42825904393691466	0.28927600862271535	0.36603698316369504	0.13398307
1 StackedEnsemble_AllModels_2_AutoML_2_20220627_94633	0.559771398177151	0.777301804917928	0.4300993297177977	0.28775170096774905	0.366576612531051	0.13437841
2 StackedEnsemble_BestOfFamily_3_AutoML_2_20220627_94633	0.5546165599479881	0.7758486160116712	0.43150243884513295	0.28852432282490215	0.367221916531411	0.13485193
3 XGBoost_3_AutoML_2_20220627_94633	0.5532331035142879	0.7754918613036507	0.5408819216607944	0.2889332145124893	0.4221384598510427	0.17820087
4 XGBoost_grid_1_AutoML_2_20220627_94633_model_1	0.5427776844384405	0.7701931710728107	0.43993686648229124	0.30139787088403264	0.37135859523467196	0.13790720
5 StackedEnsemble_AllModels_1_AutoML_2_20220627_94633	0.5420883214340447	0.7652578788467009	0.4386495703238359	0.29978656829118105	0.3705965665967111	0.13734181
6 StackedEnsemble_BestOfFamily_2_AutoML_2_20220627_94633	0.5328049869658985	0.75922775842321	0.4424648426497892	0.3023893345444515	0.3721685957513516	0.13850946
7 StackedEnsemble_BestOfFamily_1_AutoML_2_20220627_94633	0.5260407172004318	0.7548058382375052	0.4453269508809259	0.3064483372357219	0.3736766839524946	0.13963426
8 XGBoost_2_AutoML_2_20220627_94633	0.5173604491522509	0.7522582532706433	0.5321948333929017	0.3082915231327589	0.41698602820880276	0.17387734
9 XGBoost_1_AutoML_2_20220627_94633	0.5055315574827847	0.7448317538024682	0.5362985681372726	0.3156461943739352	0.417546256218893	0.17434487
10 GLM_1_AutoML_2_20220627_94633	0.502242856748446	0.7224651411387091	0.46598103448820755	0.3141602759442279	0.38103341875628394	0.14518646

AutoML実施後、検証データにおける、指定したモデル評価基準の良い順にモデルが表示される

# Flowの保存

Flowはスクリプトとして保存が可能（データや学習済みモデルが保存されるわけでは無い）



保存されているFlowの一覧

\* ホームディレクトリ下にh2oflowsディレクトリが作成され、Flowが保存される

```

H2O FLOW Flow Cell Data Model Score Admin Help
test0627
[Icons] assist
[Icons] getFrames
[Icons] getFrameSummary "UCI_Credit_Card3.hex"
[Icons] assist runAutoML, training_frame: "UCI_Credit_Card3.hex"

runAutoML {"input_spec":
{"training_frame":"UCI_Credit_Card3.hex","response_column":"default_payment_next_month","ignored_columns": ["ID"], "sort_metric": "AUCPR"}, "build_control": {"nfolds": 3, "balance_classes": true, "stopping_criteria": {"seed": -1, "max_models": 0, "max_runtime_secs": 60, "max_runtime_secs_per_model": 0, "stopping_rounds": 3, "stopping_metric": "AUTO", "stopping_tolerance": -1}, "class_sampling_factors": [], "max_after_balance_size": 5, "keep_cross_validation_predictions": true, "keep_cross_validation_models": true, "keep_cross_validation_fold_assignment": false}, "build_models": {"exclude_algos": []}, "exploitation_ratio": -1, "monotone_constraints": []}, 'exec'

getLeaderboard "AutoML_2_20220627_94633@default_payment_next_month"

getModel "StackedEnsemble_AllModels_3_AutoML_2_20220627_94633"

```

Flowメニューの"Download this Flow.."を選択すると、現在のFlowをflowファイルとしてDL、ファイルの共有が可能

ロードする場合は、"Open Flow.."から実施

保存済みFlowを開いた場合

実施内容が記録されているので、同じ内容の再実行が可能

# モデルのインポート

## MOJO形式で保存したモデルのインポート

The screenshot shows the H2O FLOW interface. The top navigation bar has 'Model' selected. In the main area, there's a sidebar titled 'Assistance' with various Routines listed. Below it, a section titled 'getModels' shows a message: 'Your H2O cloud has no models.' On the right, a dropdown menu under 'Models' has 'Import MOJO Model' highlighted with a red box and arrow.

The screenshot shows the 'Build a Model' dialog. The algorithm dropdown is set to 'Import MOJO Model'. The 'PARAMETERS' section shows 'model\_id' set to 'imported\_model', 'model\_key' as '(Choose...)', and 'path' set to 'bbb\_4130\_ad11\_f2cd44e0a106.zip'. A red box highlights the 'model\_id' field, and another red box highlights the 'path' field. Red arrows point from these highlighted fields to the corresponding sections in the text below.

### MOJOファイル (zip) の例

- gbm\_6f3364e8\_bbbb\_4130\_ad11\_f2cd44e0a106.zip

インポート後に利用する任意のモデル名を指定

MOJOファイル

# Python

- Python ClientによるH2O-3の操作

# インストール

H2O.ai

Download : <https://h2o.ai/resources/download/> ("H2O Open Source Platform"よりバージョンを選択)

The screenshot shows the "INSTALL IN PYTHON" section of the H2O download page. It includes instructions for prerequisites, dependency installation, command-line commands for removing and installing the module, and Conda installation information.

**Use H<sub>2</sub>O directly from Python**

1. Prerequisite: Python 2.7.x, 3.5.x to 3.7.x
2. Install dependencies (prepending with `sudo` if needed):

```
pip install requests  
pip install tabulate  
pip install future
```

At the command line, copy and paste these commands one line at a time:

```
# 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 https://h2o-release.s3.amazonaws.com/h2o/rel-zumbo/2/Python/h2o-3.36.1.2-py2.py3-none-any.whl
```

**Conda Installation**  
Available at <https://anaconda.org/h2oai/h2o/>  
To install this package with conda run:

```
conda install -c h2oai h2o
```

\* H2O-3本体 (j2o.jar) も含まれるのでFlowのインストールでDLしたファイル (h2o-3.xx.x.x.zip) のDLは必要なし

# H2O-3の開始

H2O.ai

Jupyter Notebookを利用

```
In [1]: import h2o  
print(h2o.__version__)  
  
3.36.1.2  
  
In [2]: h2o.init()  
  
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 "11.0.9" 2020-10-20 LTS; Java(TM) SE Runtime Environment 18.9 (build 11.0.9+7-LTS); Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.9+7-LTS, mixed mode)  
Starting server from /opt/anaconda3/lib/python3.8/site-packages/h2o/backend/bin/h2o.jar  
Ice root: /var/folders/50/glkvvvhjx03g22fwy2r0tgylr0000gn/T/tmp2_aht5ki  
JVM stdout: /var/folders/50/glkvvvhjx03g22fwy2r0tgylr0000gn/T/tmp2_aht5ki/h2o_YShimada万博16_started_from_python.out  
JVM stderr: /var/folders/50/glkvvvhjx03g22fwy2r0tgylr0000gn/T/tmp2_aht5ki/h2o_YShimada万博16_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_timezone:      | Asia/Tokyo                                                  |
| H2O_data_parsing_timezone: | UTC                                                         |
| H2O_cluster_version:       | 3.36.1.2                                                    |
| H2O_cluster_version_age:   | 10 days                                                     |
| H2O_cluster_name:          | H2O_from_python_YShimada万博16_ygy2tn                         |
| H2O_cluster_total_nodes:   | 1                                                           |
| H2O_cluster_free_memory:   | 8 Gb                                                        |
| H2O_cluster_total_cores:   | 16                                                          |
| H2O_cluster_allowed_cores: | 16                                                          |
| H2O_cluster_status:        | locked, healthy                                             |
| H2O_connection_url:        | <a href="http://127.0.0.1:54321">http://127.0.0.1:54321</a> |
| H2O_connection_proxy:      | {"http": null, "https": null}                               |
| H2O_internal_security:     | False                                                       |
| Python_version:            | 3.8.3 final                                                 |


```

h2o.init()により、H2O-3が開始、もしくは動作中のH2O-3へ接続を実施

クリックすると、Flowが開く

# H2O-3の設定

H2O.ai

## h2o.init()のオプション

### url, ip, port

- 接続するH2O-3
- ローカルのH2O-3へ接続する場合は指定の必要なし

### nthreads

- 利用するCPUコア数

### max\_mem\_size

- 利用するメモリ容量

```
In [1]: import h2o
h2o.__version__
Out[1]: '3.36.0.4'

In [2]: h2o.init(nthreads=2, max_mem_size=4)
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 "11.0.9" 2020-10-20 LTS; Java(TM) SE Runtime Environment 18.9 (build 11.0.9+7-LTS); Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.9+7-LTS, mixed mode)
Starting server from /opt/anaconda3/lib/python3.8/site-packages/h2o/backend/bin/h2o.jar
Ice root: /var/folders/50/glkvvhjx03g22fwy2r0tgylr0000gn/T/tmp8l45jdqw
JVM stdout: /var/folders/50/glkvvhjx03g22fwy2r0tgylr0000gn/T/tmp8l45jdqw/h2o_YShimada_MBP16_starte
d_from_python.out
JVM stderr: /var/folders/50/glkvvhjx03g22fwy2r0tgylr0000gn/T/tmp8l45jdqw/h2o_YShimada_MBP16_starte
d_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_timezone: Asia/Tokyo
H2O_data_parsing_timezone: UTC
H2O_cluster_version: 3.36.0.4
H2O_cluster_version_age: 3 months and 4 days
H2O_cluster_name: H2O_from_python_YShimada_MBP16_7sii7k
H2O_cluster_total_nodes: 1
H2O_cluster_free_memory: 4 Gb
H2O_cluster_total_cores: 16
H2O_cluster_allowed_cores: 2
H2O_cluster_status: locked, healthy
H2O_connection_url: http://127.0.0.1:54321
H2O_connection_proxy: {"http": null, "https": null}
H2O_internal_security: False
Python_version: 3.8.3 final
```

# 実施例

H2O.ai

デモ用Notebook : [https://github.com/yukismd/H2O\\_3\\_Tutorial/blob/master/examples\\_python/Quick\\_Demo.ipynb](https://github.com/yukismd/H2O_3_Tutorial/blob/master/examples_python/Quick_Demo.ipynb)

実施内容 :

- H2O-3の起動
- データのロード
- データ分割
- モデルのあてはめ(GBM)
- 機械学習の解釈可能性 (Global、Local (1オブザベーション) なモデルの解釈)
- スコアリング (予測値、SHAP)
- モデルの保存とロード
- ハイパーパラメータチューニング
- AutoML
- H2O-3の終了

各アルゴリズム実施例 : [https://github.com/yukismd/H2O\\_3\\_Tutorial/tree/master/examples\\_python](https://github.com/yukismd/H2O_3_Tutorial/tree/master/examples_python)

# R

- R ClientによるH2O-3の操作

# インストール

H2O.ai

Download : <https://h2o.ai/resources/download/> ("H2O Open Source Platform"よりバージョンを選択)

The screenshot shows the H2O download page. At the top, there's a yellow header with the H2O logo and "Version 3.36.1.2". Below it is a black header with the text "Fast Scalable Machine Learning API" and "For Smarter Applications". A navigation bar below the headers includes "DOWNLOAD AND RUN", "INSTALL IN R" (which is highlighted with a red box), "INSTALL IN PYTHON", "INSTALL ON HADOOP", "USE FROM MAVEN", and "KUBERNETES". The main content area is titled "Use H2O directly from R" and contains R code for installation:

```
# 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.  
pkgs <- c("RCurl", "jsonlite")  
for (pkg in pkgs) {  
  if (! (pkg %in% rownames(installed.packages()))) { install.packages(pkg) }  
}  
  
# Now we download, install and initialize the H2O package for R.  
install.packages("h2o", type="source", repos="https://h2o-release.s3.amazonaws.com/h2o/rel-zumbo/2/R")  
  
# Finally, let's load H2O and start up an H2O cluster  
library(h2o)  
h2o.init()
```

\* H2O-3本体 (j2o.jar) も含まれるのでFlowのインストールでDLしたファイル (h2o-3.xx.x.x.zip) のDLは必要なし

# 実施例

H2O.ai

デモ用R Script : [https://github.com/yukismd/H2O\\_3\\_Tutorial/blob/master/examples\\_r/Quick\\_Demo.R](https://github.com/yukismd/H2O_3_Tutorial/blob/master/examples_r/Quick_Demo.R)

実施内容 :

- H2O-3の起動
- データのロード
- データ分割
- モデルのあてはめ(GBM)
- 機械学習の解釈可能性 (Global、Local (1オブザベーション) なモデルの解釈)
- スコアリング (予測値、SHAP)
- モデルの保存とロード
- ハイパーパラメータチューニング
- AutoML
- H2O-3の終了

各アルゴリズム実施例 : [https://github.com/yukismd/H2O\\_3\\_Tutorial/tree/master/examples\\_r](https://github.com/yukismd/H2O_3_Tutorial/tree/master/examples_r)

# 詳細

- データの前処理（欠損、エンコーディング）について
- K分割交差検証法について
- モデルの解釈（解釈可能性、モデルの比較）について
- モデルファイル

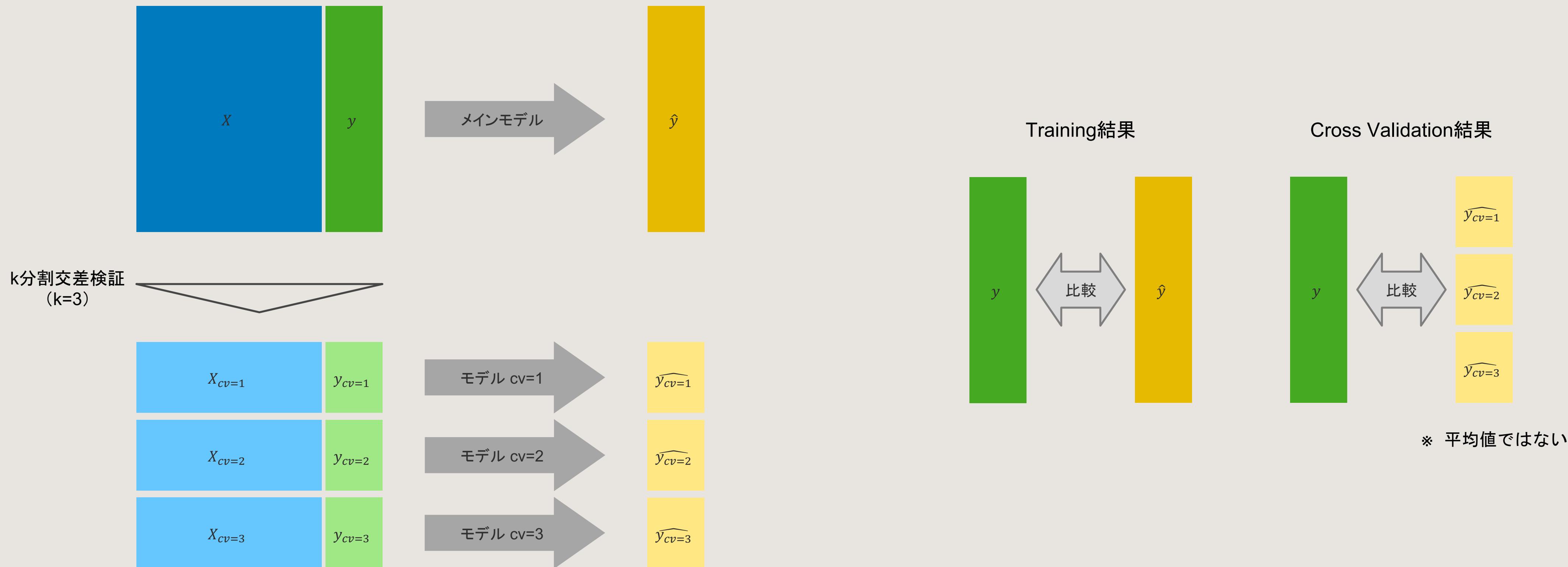
H2O-3では、欠損値（Imputation）とカテゴリカル変数（One-Hot Encodingなど）の処理はアルゴリズム内で実施されるので、前処理の事前実施は不要  
(処理効率の観点から、前処理なしでのアルゴリズムへの投入を推奨)

処理方法は、アルゴリズム毎に異なる。詳細は各アルゴリズムのFAQ参照

- GLMでは、カテゴリカル変数に対しOne-Hot-Encodingが実施され、欠損はスキップや平均値・最頻値補完の選択が可能 (missing\_values\_handling引数 ([https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/missing\\_values\\_handling.html](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/missing_values_handling.html)) )
- Tree系アルゴリズムでは、カテゴリカル変数のエンコーディング方法の選択が可能 (categorical\_encoding引数 ([https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/categorical\\_encoding.html](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/categorical_encoding.html)) )、欠損は欠損として扱われる（欠損を示すカテゴリカル変数を内部作成）

# k分割交差検証 (k Cross-Validation)

k分割交差検証を実施した際、 $k+1$ 個のモデルが作成され、全量データで推定されたモデル（メインモデル）の学習データに対する結果（Training結果）と、交差検証による $k$ 個のモデルによる予測を結合して学習データと比較した結果（Cross Validation結果）が表示される



## 機械学習の解釈可能性 (Machine Learning Interpretability) やモデルの比較を実施

### 機械学習の解釈可能性

- Confusion Matrix (分類問題に対して)
- Residual Analysis (回帰問題に対して)
- Variable Importance (Model Specificなアプローチ)
- Permutation Variable Importance (Model Agnosticなアプローチ)
- SHAP (GlobalとLocal両方へ対応)
- Partial Dependence (PD) Plots
- Individual Conditional Expectation (ICE) Plots

### モデルの比較 (AutoMLの実施結果など、複数のモデルを渡した場合に実施)

- Variable Importance Heatmap (compare all non-Stacked models)
- Model Correlation Heatmap (compare all models)

実施例 : [https://github.com/yukismd/H2O\\_3\\_Tutorial/blob/master/examples\\_python/model\\_explain.ipynb](https://github.com/yukismd/H2O_3_Tutorial/blob/master/examples_python/model_explain.ipynb)

# モデルファイル

## ● Javaベースのモデルファイル – MOJO/POJO

H2O-3では、POJO(Plain Old Java Object)やMOJO(Model ObJect, Optimized)形式でモデルファイルを提供

- MOJOは、POJOの改良版
- Java環境での実行の際、モデルファイルとh2o-genmodel.jarライブラリファイルのみ必要 (H2O-3が不要)
- Python、R APIからMOJOファイルのロードが可能

Javaでの実行例 : [https://github.com/yukismd/H2O\\_3\\_Tutorial/tree/master/scoring/java](https://github.com/yukismd/H2O_3_Tutorial/tree/master/scoring/java)

Pythonでの実行例 : [https://github.com/yukismd/H2O\\_3\\_Tutorial/tree/master/scoring/python](https://github.com/yukismd/H2O_3_Tutorial/tree/master/scoring/python)

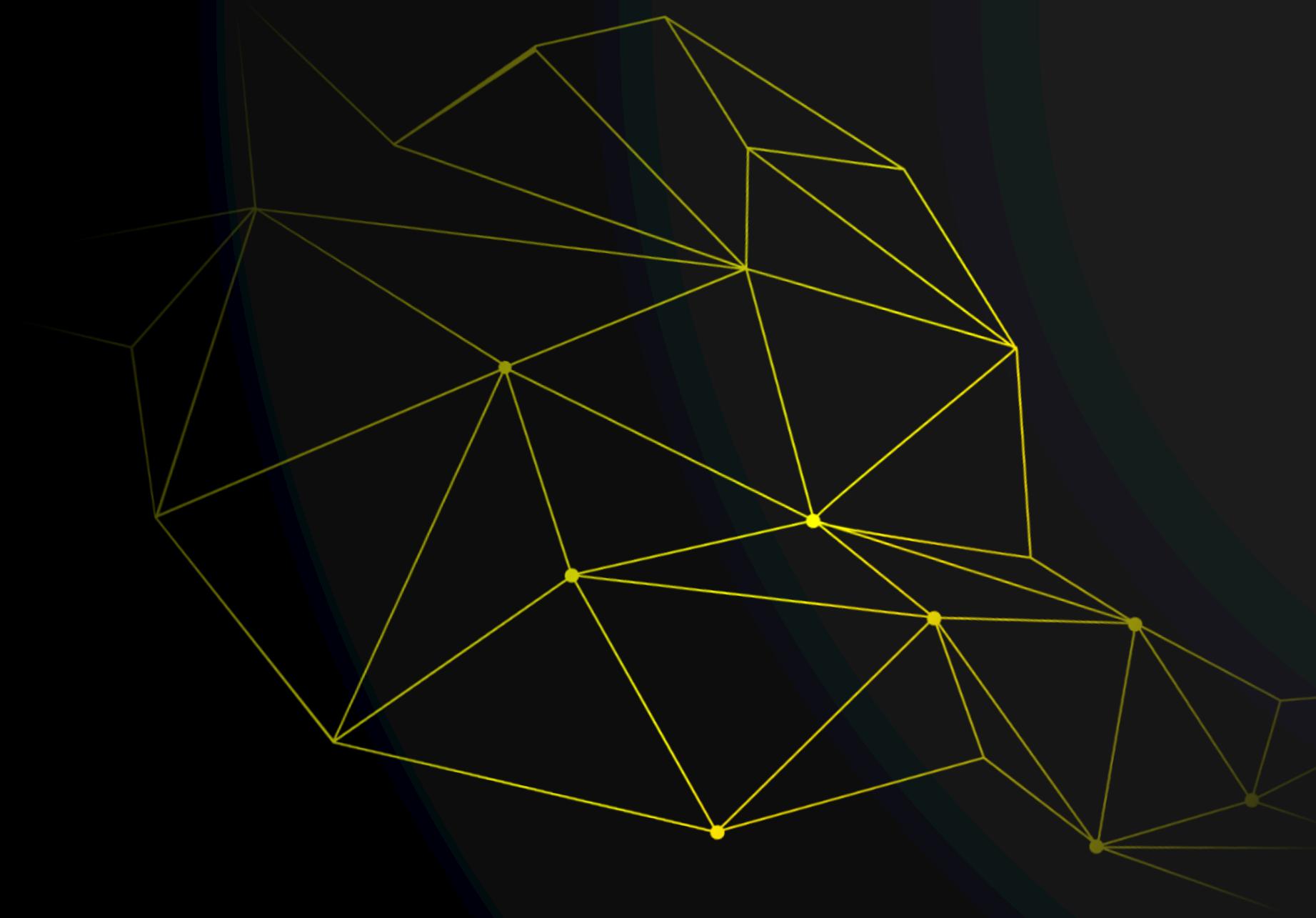
## ● バイナリファイル

PythonやRクライアント、Flow環境で作成したモデルをバイナリファイルとして保存

バイナリファイルとしての保存したモデルは、Python、R、Flow間での受け渡しが可能 (H2O-3のバージョンを一致させる必要があることに注意)

# H2O AI Cloud

- AI Cloud上のH2O-3へのクライアント接続



# AI Cloud上のH2O-3の起動

H2O.ai

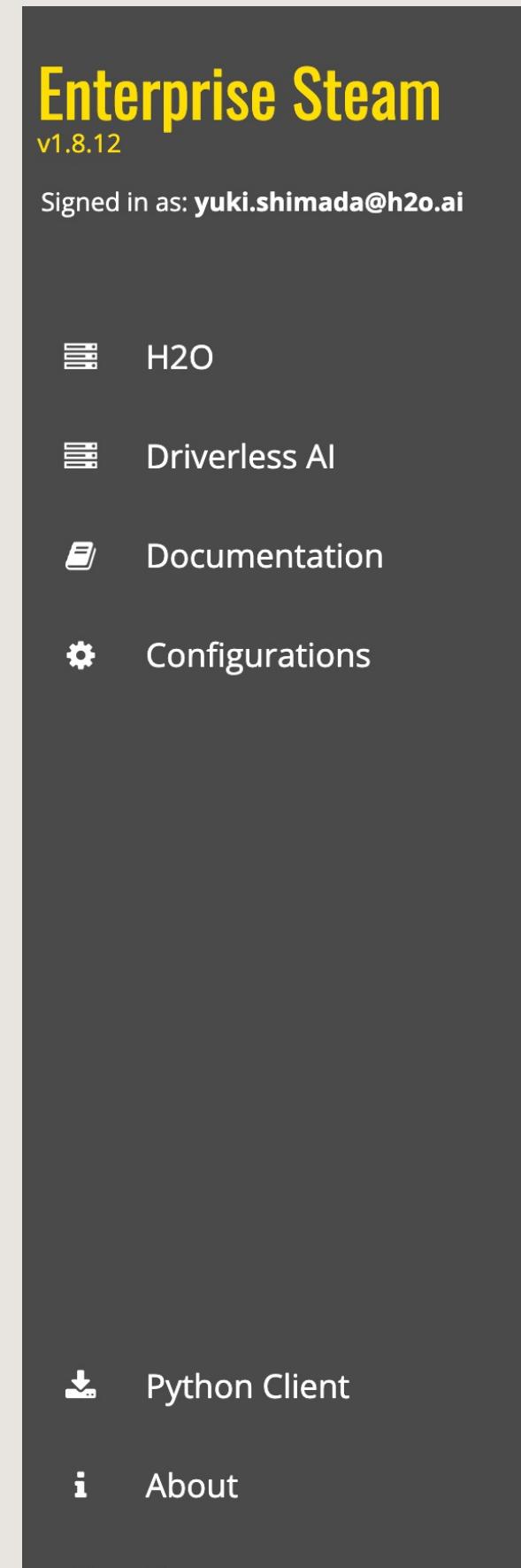
## AI Cloud上のH2O-3の起動

Home > H2O > Clusters > Launch

### NEW H2O CLUSTER

SELECT PROFILE

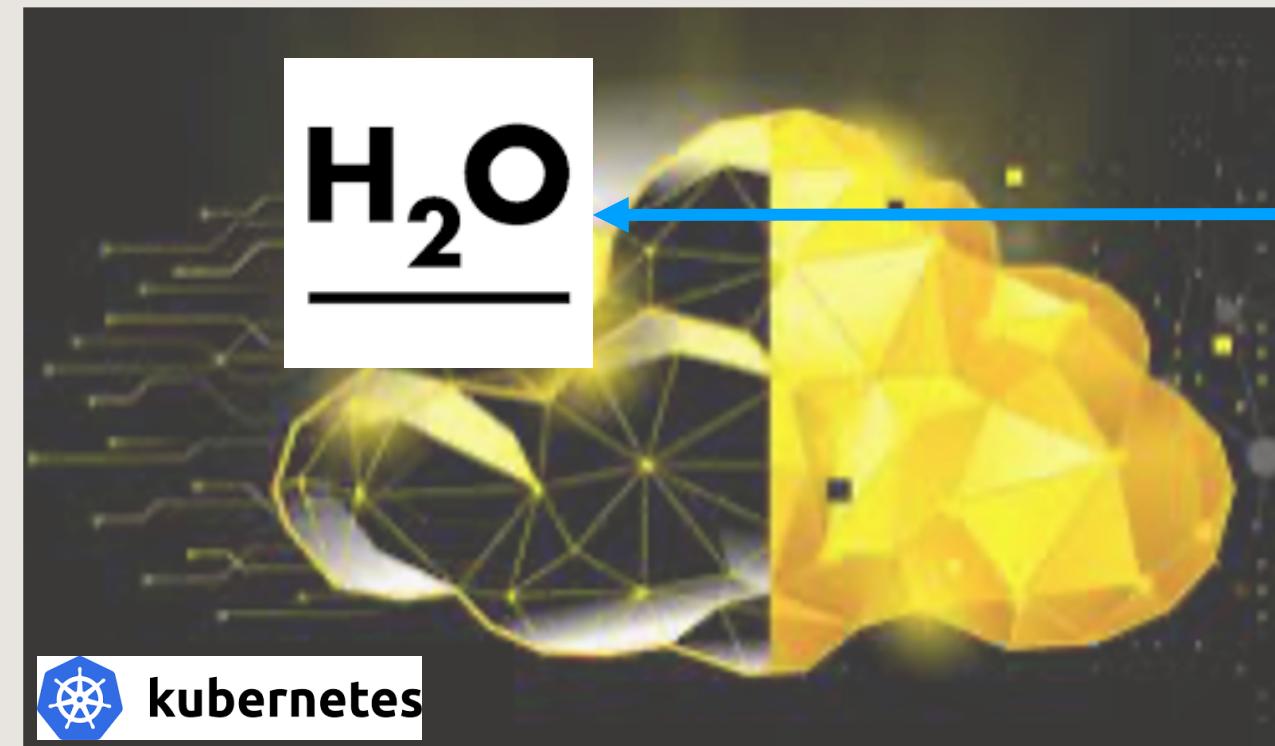
CLUSTER NAME	<input type="text" value="my-h2o3"/>	required must meet requirements
H2O VERSION	<input type="button" value="3.36.1.2"/>	required
DATASET PARAMETERS	<input type="button" value="Set parameters (not set)"/>	optional
NUMBER OF NODES	<input type="text" value="1"/>	required min: 1, max: 12
NUMBER OF CPUS	<input type="text" value="1"/>	required min: 1, max: 30
NUMBER OF GPUS	<input type="text" value="0"/>	required min: 0, max: 0
MEMORY PER NODE [GB]	<input type="text" value="4"/>	required min: 4, max: 248
MAXIMUM IDLE TIME [HRS]	<input type="text" value="8"/>	required min: 1, max: 24
MAXIMUM UPTIME [HRS]	<input type="text" value="12"/>	required min: 1, max: 24
TIMEOUT [S]	<input type="text" value="600"/>	required min: 300, max: 1800



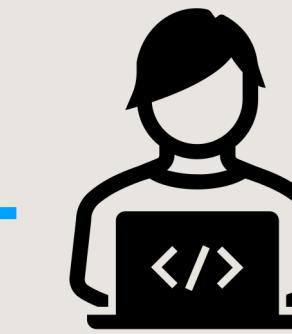
# Client Access

H2O.ai

AI Cloud上のH2O-3へのアクセスする場合、AI Cloud用のインターフェイス（h2osteam）の利用が必要



H2O AI Cloud プラットフォーム  
(kubernetes上で稼働)



Python Example

```
import h2osteam
import h2o

# AI Cloud上で、H2O-3クラスターを作成
h2o_cluster = h2osteam.clients.H2oKubernetesClient().launch_cluster(
    name="h2o-cluster",
    version="3.36.1.2")

# 作成したH2O-3クラスターへ接続
h2o_cluster.connect()

# h2oライブラリを用い操作が可能
# 例えば、データのアップロード
h2o.upload_file("mydata.csv")
```

実施例：

- [https://github.com/yukismd/H2O\\_AI\\_Cloud/tree/main/client\\_access](https://github.com/yukismd/H2O_AI_Cloud/tree/main/client_access)
- <https://github.com/h2oai/haic-tutorials/blob/main/4%20H2O-3:%20Distributed%20ML.ipynb>

# Appendix

# Requirementsとインストール

Document - Requirements : <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/welcome.html#requirements>

- Javaのインストール

Document - DL and Install : <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/downloading.html#downloading-installing-h2o>