

# Extending H2O Driverless AI with Your Own Recipes



Slides: [bit.ly/h2o\\_meetups](https://bit.ly/h2o_meetups)

Stefan Pacinda  
Senior Solution Architect @ H2O.ai  
[stefan@h2o.ai](mailto:stefan@h2o.ai)

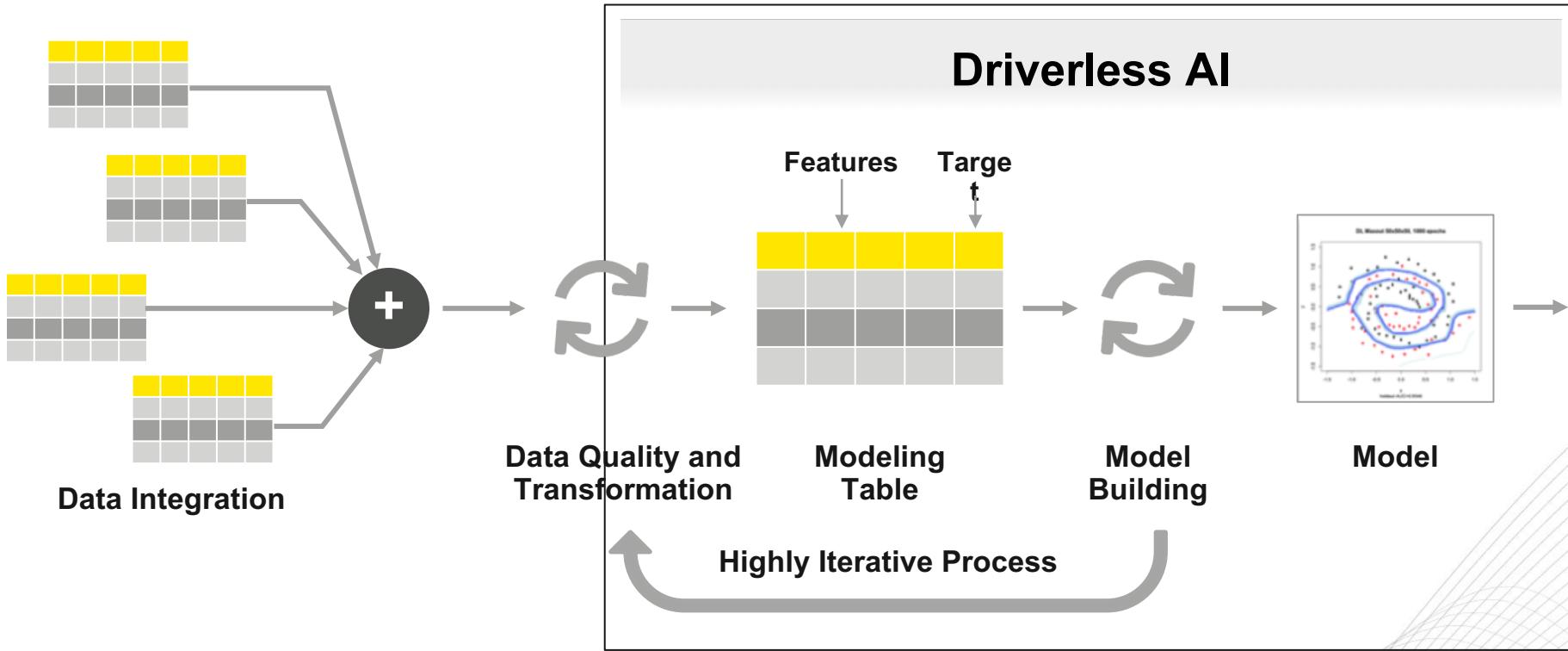


# H2O Driverless AI

## Automatic Machine Learning for the Enterprise

# A Quick Introduction to Driverless AI

## Auto Feature Engineering, Model Building & More



# A Proper Introduction to H2O Driverless AI

[www.youtube.com/watch?v=5t2zw4bVfsw](https://www.youtube.com/watch?v=5t2zw4bVfsw)

H<sub>2</sub>O.ai

The image shows a YouTube video player interface. The video itself is titled "Introduction to H2O Driverless AI" and features a man named John Spooner, Director of Solution Engineering - EMEA. The video is set against a background of hexagonal icons related to machine learning and technology. The YouTube interface includes a search bar at the top, a play button, a progress bar showing 1:34 / 1:29:18, and various interaction buttons like like, dislike, share, and save. Below the video, there's a channel info card for "H2O.ai" with 10.1K subscribers, and a note about the session being held at the Dive into H2O: London training on June 17, 2019.

Introduction to H2O Driverless AI

John Spooner  
Director of Solution Engineering - EMEA

1:34 / 1:29:18

Hands-On with H2O Driverless AI by John Spooner

565 views • Published on 19 Jun 2019

10.1K subscribers

This session was held at the Dive into H2O: London training on June 17, 2019.

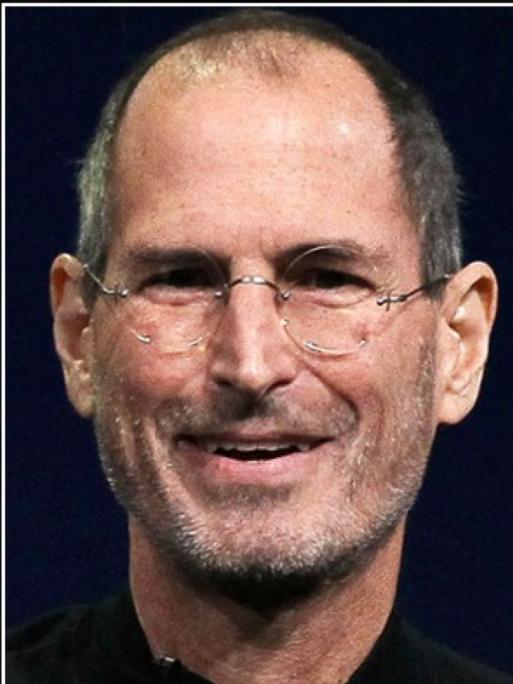
SHOW MORE

LIKE   DISLIKE   SHARE   SAVE

SUBSCRIBED

# BYOR - The Best Ideas Have to Win

H<sub>2</sub>O.ai



If you want to hire great people and have them stay, you have to be run by ideas, not hierarchy. The best ideas have to win.

— Steve Jobs —

AZ QUOTES

<https://www.youtube.com/watch?v=AtuNR7MSj-o>

# H2O.ai Resources

- Main Documentation Site: [docs.h2o.ai](https://docs.h2o.ai)
- Tutorials: [h2oai.github.io/tutorials/](https://h2oai.github.io/tutorials/)
- Blog: [www.h2o.ai/blog/](https://www.h2o.ai/blog/)
- Webinars: [www.h2o.ai/webinars/](https://www.h2o.ai/webinars/)
- Community Slack: [www.h2o.ai/community/developerless-ai-community/](https://www.h2o.ai/community/developerless-ai-community/)
- YouTube Channel: [www.youtube.com/user/0xdata/videos](https://www.youtube.com/user/0xdata/videos)

# BYOR

# Bring Your Own Recipe

New Feature in Version 1.7

# BYOR - Adding My Own Feature Transformation

H<sub>2</sub>O.ai

## Driverless AI

### Kaggle Grandmasters' Recipes

|                             |                 |
|-----------------------------|-----------------|
| CVCatNumEncodeTransformer   | GENERAL         |
| CVTargetEncodeTransformer   |                 |
| CatOriginalTransformer      | informative     |
| ClusterDistTransformer      | SELECT VALUES ▾ |
| ClusterIdTransformer        |                 |
| ClusterTETransformer        | failures of     |
| DateOriginalTransformer     | ENABLED         |
| DateTimeOriginalTransformer |                 |
| DatesTransformer            |                 |
| EwmaLagsTransformer         |                 |
| FrequentTransformer         |                 |
| InteractionsTransformer     |                 |
| IsHolidayTransformer        |                 |
| LagsAggregatesTransformer   |                 |
| LagsInteractionTransformer  |                 |
| LagsTransformer             |                 |
| NumCatTETransformer         |                 |



## BYOR

### Your Domain Knowledge

- Data Augmentation
  - local holidays, zipcode
- Domain Specific
  - Special count of numbers/words
- NLP
  - Specific features generation
- Your favourite algorithms

# BYOR - Adding My Own Feature Transformation

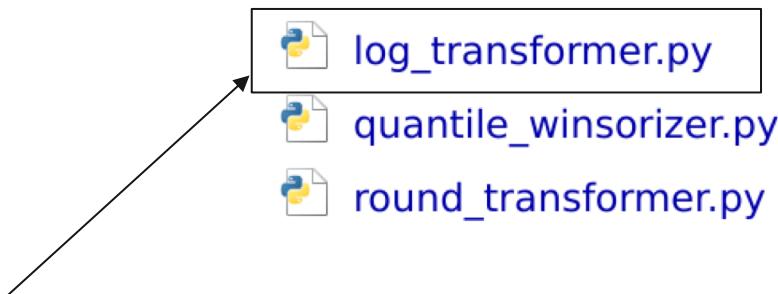
For example: Log(x)

```
"""Converts numbers to their Logarithm"""
from h2oaicore.transformer_utils import CustomTransformer
import datatable as dt
import numpy as np

class JoeSupaDupaLogTransformer(CustomTransformer):
    @staticmethod
    def get_default_properties():
        return dict(col_type="numeric", min_cols=1, max_cols=3, relative_importance=1)

    def fit_transform(self, X: dt.Frame, y: np.array = None):
        return self.transform(X)

    def transform(self, X: dt.Frame):
        return X[:, [dt.log(dt.f[i]) for i in range(X.ncols)]]
```



# BYOR - Adding My Own Feature Transformation

## Upload recipes to Driverless AI

Expert Experiment Settings

Upload local .py or import from URL

+ UPLOAD CUSTOM RECIPE    + LOAD CUSTOM RECIPE FROM URL    OFFICIAL RECIPES (EXTERNAL)

GENERAL MODEL TIMESERIES NLP SYSTEM RECIPES OTHERS

Include specific transformers  
SELECT VALUES ▾

Include specific models  
SELECT VALUES ▾

Include specific scorers  
SELECT VALUES ▾

Whether to skip failures of transformers  
ENABLED

JoeSupaDupaLogTransformer  
JoeSupaDupaQuantileWinsorizer  
JoeSupaDupaRoundTransformer

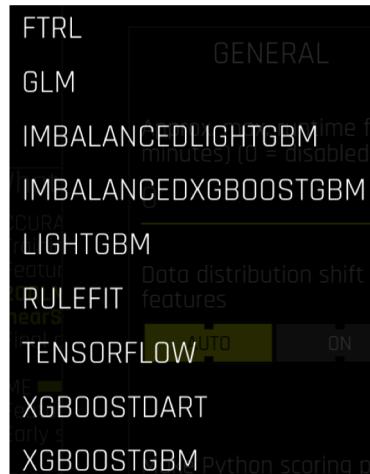
My own recipes are now available to “fight” with Kaggle Grandmasters’ recipes.

# BYOR - Adding Machine Learning Algorithms

H<sub>2</sub>O.ai

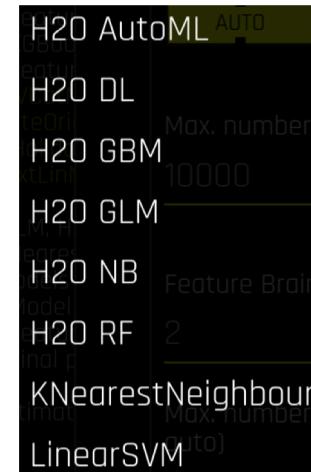
## Driverless AI

Out-of-the box and optimised



## BYOR

The Wider Python Ecosystem



... and a lot  
more!

Microsoft  
**LightGBM**

dmlc  
**XGBoost**



# BYOR - Adding Machine Learning Algorithms

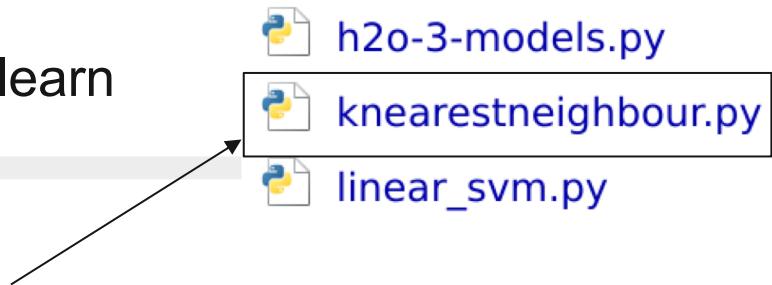
For example: K-Nearest Neighbor from sklearn

```
"""K-Nearest Neighbor implementation by sklearn. For small data (< 200k rows)."""
import datatable as dt
import numpy as np
from sklearn.preprocessing import LabelEncoder
from h2oai.core.models import CustomModel
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Ridge # will be used to derive feature importances
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier

class KNearestNeighbourModel(CustomModel):
    _regression = True
    _binary = True
    _multiclass = True

    _display_name = "KNearestNeighbour"
    _description = "K Nearest Neighbour Model based on sklearn. Not advised if the data is larger than 200K rows"

    def set_default_params(self,
                           accuracy=None, time_tolerance=None, interpretability=None,
                           **kwargs):
        n_jobs = -1
        n_neighbors = min(kwargs['n_neighbors'], 1000) if 'n_neighbors' in kwargs else 10
        metric = kwargs['metric'] if "metric" in kwargs and kwargs['metric'] in ["minkowski",
                                                                           "cityblock"] else "cityblock"
        self.params = {'n_neighbors': n_neighbors,
                      'metric': metric,
                      'weights': "uniform",
                      'n_jobs': n_jobs, # -1 is not supported
                      }
```



# BYOR - Adding Machine Learning Algorithms

## Upload recipes to Driverless AI

Expert Experiment Settings

Upload local .py or import from URL

+ UPLOAD CUSTOM RECIPE

+ LOAD CUSTOM RECIPE FROM URL

OFFICIAL RECIPES (EXTERNAL)

GENERAL

MODEL

TIMESERIES

NLP

Include specific transformers

SELECT VALUES ▾

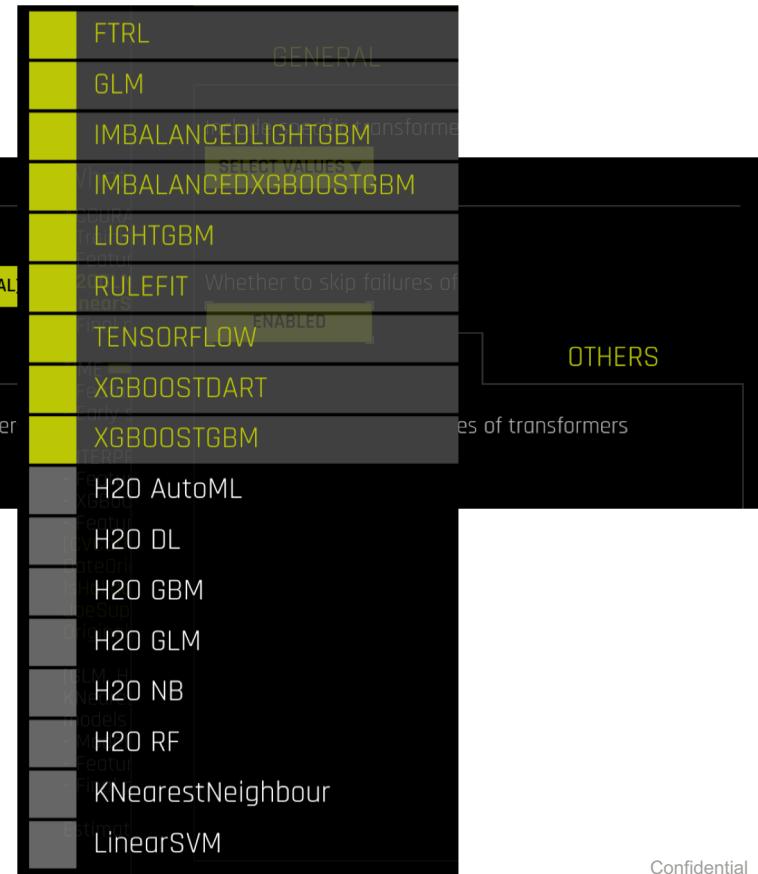
Include specific models

SELECT VALUES ▾

Include specific scorer

SELECT VALUES ▾

My choices of ML algorithms  
are now available to “fight” with  
out-of-the-box algorithms.



# Bring Your Own Recipe — Driverless Platform Community

## Models / Algorithms



## Transformers

|                             |
|-----------------------------|
| Zipcode5Transformer         |
| ZipcodeCityTransformer      |
| ZipcodeIsActiveTransformer  |
| ZipcodeLatitudeTransformer  |
| ZipcodeLongitudeTransformer |
| ZipcodeStateTransformer     |
| ZipcodeTypeTransformer      |

## Scorers/Metrics

|          |
|----------|
| ACCURACY |
| AUC      |
| AUCPR    |
| AVGMCC   |
| BRIER    |
| CAMPAIGN |
| F05      |

<https://github.com/h2oai/driverlessai-recipes>

(Apache V2)

## Sample Recipes

[Go to Recipes for Driverless 1.7.0](#)

Count: 81

- MODELS
  - [model\\_template.py](#) [Template base class]
  - [ALGORITHMS](#)
    - [catboost.py](#) [CatBoost gradient boosting classification.]
    - [daal\\_trees.py](#) [Binary Classification using Intel DAAL]
    - [extra\\_trees.py](#) [Extremely Randomized Trees]
    - [h2o-3-models.py](#) [H2O-3 Distributed (DL/GLM/GBM/DRF/NB/AutoML)]
    - [h2o-glm-poisson.py](#) [H2O-3 Distributed GLM Poisson]
    - [knearestneighbour.py](#) [K-Nearest Neighbors]
    - [libfm\\_fastfm.py](#) [LibFM implementation]
    - [linear\\_svm.py](#) [Linear Support Vector Machine]
    - [nusvm.py](#) [Nu-SVM implementation]
  - [CUSTOM\\_LOSS](#)
    - [lightgbm\\_with\\_custom\\_loss.py](#) [Modifying LightGBM's custom objective function (used for regression).]
    - [xgboost\\_with\\_custom\\_loss.py](#) [Modifying XGBoost's custom objective function (used for regression).]
  - [TIMESERIES](#)
    - [exponential\\_smoothing.py](#) [Linear Moving Average. Provide appropriate lags and seasonalities.]
    - [fb\\_prophet.py](#) [Prophet by Facebook]
    - [historic\\_mean.py](#) [Historic Mean for timegroup for regression problems.]

## Recipes for H2O Driverless AI

### About Driverless AI

H2O Driverless AI is Automatic Machine Learning for the Enterprise. Driverless AI auto model building, visualization and interpretability.

- Learn more about Driverless AI from the [H2O.ai website](#)
- Take the [test drive](#)
- Go to the [Driverless AI community Slack channel](#) and ask your BYOR related questions

### About BYOR

BYOR stands for Bring Your Own Recipe and is a key feature of Driverless AI. It allows you to solve problems faster and with more precision.

### What are Custom Recipes?

Custom recipes are Python code snippets that can be uploaded into Driverless AI at runtime. You can upload them to Driverless AI and restart Driverless AI. Custom recipes can be provided for transformers, models and scorers. Driverless AI provides a supervised machine learning modeling pipeline (aka experiment), Driverless AI can then use these building blocks, in combination with all built-in code pieces (or instead of). By providing a custom recipe, you can gain control over the optimization choices that Driverless AI makes to best solve your problem.

### Best Practices for Recipes

#### Security

- Recipes are meant to be built by people you trust and each recipe should be code reviewed before production.
- Assume that a user with access to Driverless AI has access to the data inside the recipe.
- Apart from securing access to the instance via private networks, various methods can be used to secure the data inside the recipe.

# BYOR Use Case Credit Card Loan

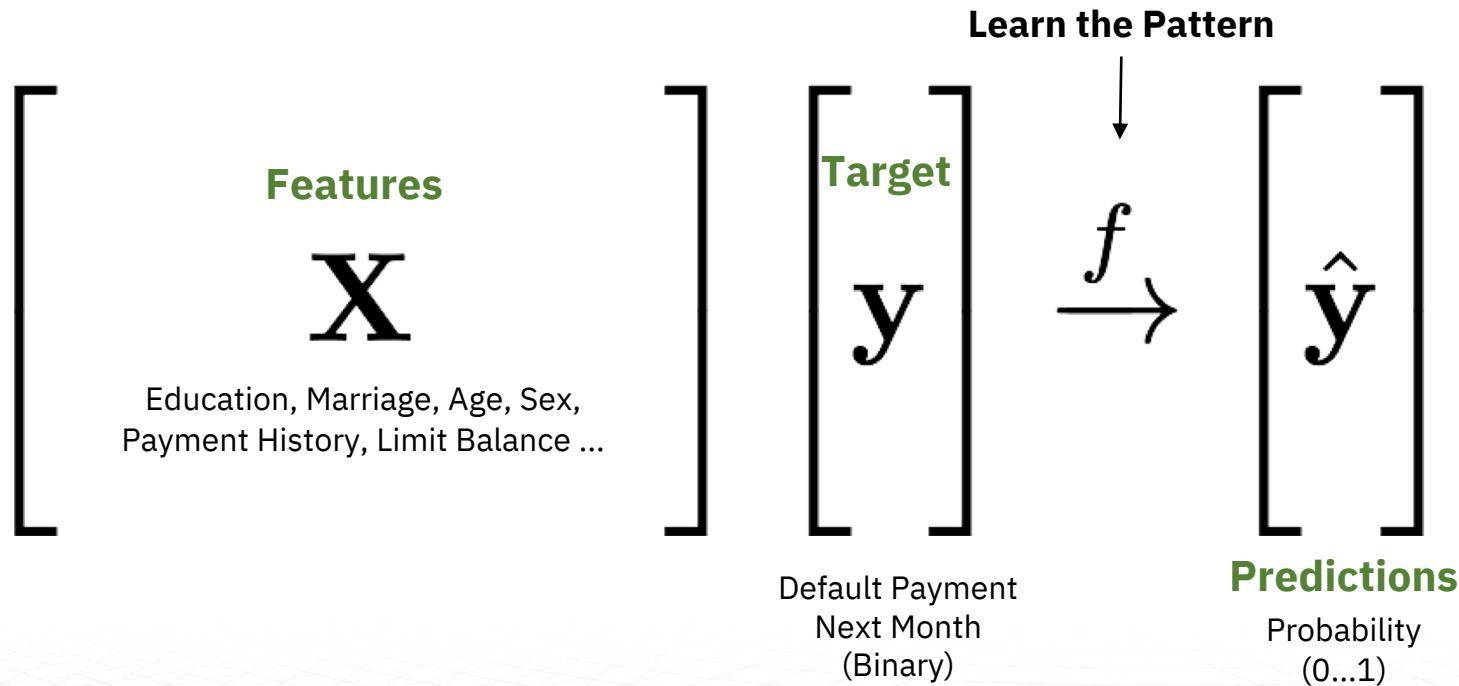
# Credit Card Example

- Based on a customer's past behavior, can we predict if they will default on a credit card payment?
- Based on the information known at the start of a loan, can we predict if the loan will be fully paid off?

# Credit Card Example

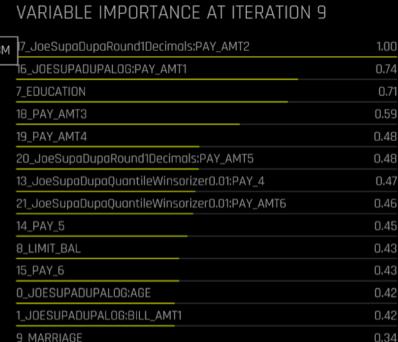
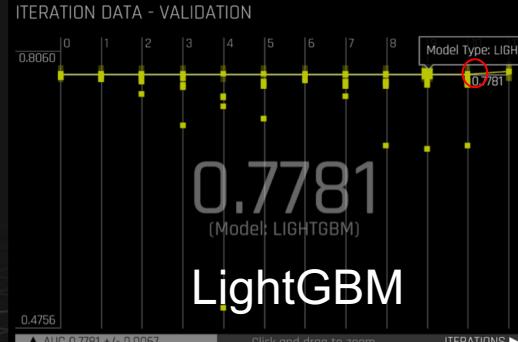
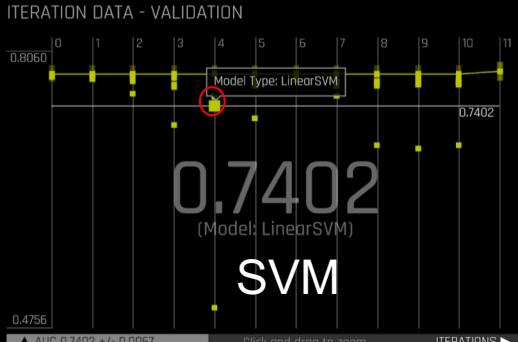
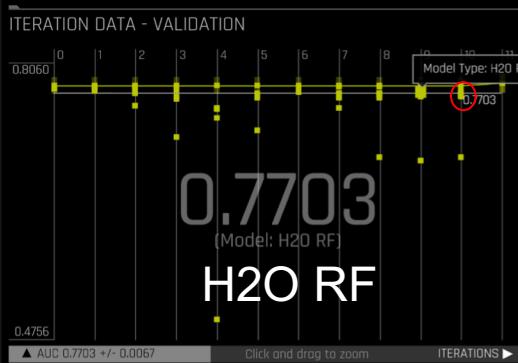
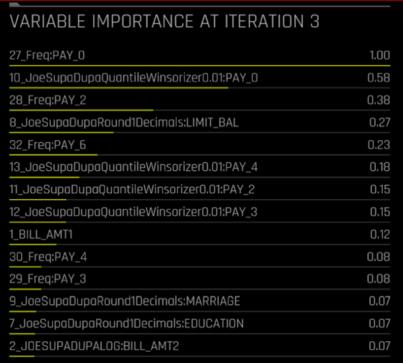
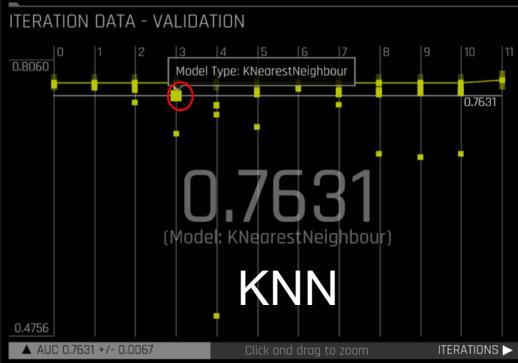
| Column Name                | Description  |
|----------------------------|--|
| ID                         | ID of each client  |
| LIMIT_BAL                  | Amount of given credit in NT dollars (includes individual and family/supplementary credit) |
| SEX                        | Gender (1=male, 2=female)  |
| EDUCATION                  | (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)           |
| MARRIAGE                   | Marital status (1=married, 2=single, 3=others)   |
| AGE                        | Age in years   |
| PAY_x {-1,0,2,3,4,5,6}     | Repayment status in August, 2005 – April, 2005   |
| BILL_AMTx {1, ..., 6}      | Amount of bill statement in September, 2005 – April, 2005 (NT dollar)                      |
| PAY_AMTx {1, ..., 6}       | Amount of previous payment in September, 2005 – April, 2005 (NT dollar)                    |
| default.payment.next.month | Default payment (1=yes, 0=no)  |

# Learning from Credit Card Data



# Live Demo

# Evaluating Different Combinations of Feature Transformation and ML Algorithms



**Evaluation Metric: AUC (Higher = Better)**

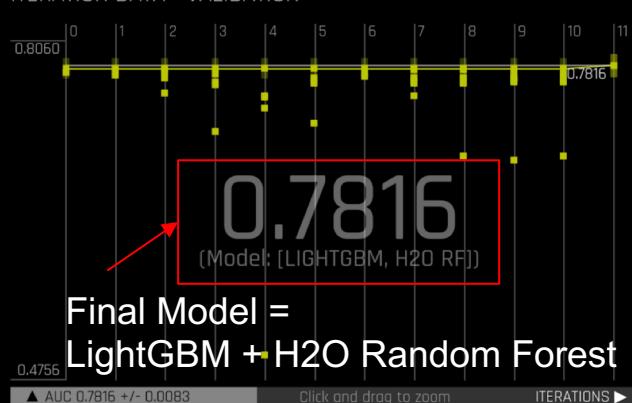
# H2O.ai Experiment BYOR Demo

DRIVERLESS AI 1.7.1 - AI TO DO AI  
Licensed to H2O.ai (SN35). Current User - JOE

## EXPERIMENT SETUP

| DISPLAY NAME       | CreditCard-train.csv |
|--------------------|----------------------|
| ROWS               | 24K                  |
| COLUMNS            | 25                   |
| DROPPED COLS       | 0                    |
| VALIDATION DATASET | --                   |
| TEST DATASET       | Yes                  |
|                    | CreditCard-test.csv  |
| TARGET COLUMN      | default payment next |
| FOLD COLUMN        | --                   |
| WEIGHT COLUMN      | --                   |
| TIME COLUMN        | [OFF]                |
| TYPE               | bool                 |
| COUNT              | 23999                |
| UNIQUE             | 2                    |
| TARGET FREQ        | 5369                 |

## ITERATION DATA - VALIDATION



## ASSISTANT

### STATUS: COMPLETE

DEPLOY (LOCAL & CLOUD)

INTERPRET THIS MODEL

DIAGNOSE MODEL ON NEW DATASET...

SCORE ON ANOTHER DATASET

TRANSFORM ANOTHER DATASET...

DOWNLOAD PREDICTIONS ▾

BUILD PYTHON SCORING PIPELINE

BUILD MOJO SCORING PIPELINE

DOWNLOAD EXPERIMENT SUMMARY

DOWNLOAD LOGS

## TRAINING SETTINGS

6  
ACCURACY

2  
TIME

6  
INTERPRETABILITY

AUC  
SCRER

CLASSIFICATION

REPRODUCIBLE

GPUS ENABLED

PARENT EXPERIMENT:

CPU / MEMORY

Notifications Log Trace

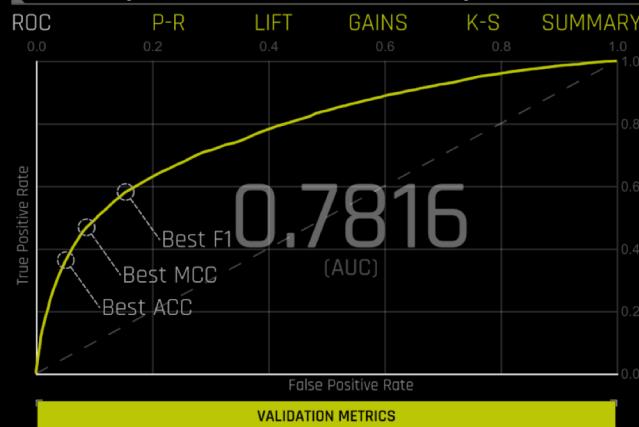
CPU

MEM

Some of Joe's "SupaDupa" recipes made it to the top!

## VARIABLE IMPORTANCE

|   |      |
|---|------|
| 10_JOESUPADUPALOG:PAY_0                       | 1.00 |
| 11_PAY_2                                      | 0.45 |
| 12_JOESUPADUPALOG:PAY_3                       | 0.20 |
| 16_PAY_AMT1                                   | 0.14 |
| 8_LIMIT_BAL                                   | 0.13 |
| 1_JoeSupaDupaRoundDecimals:BILL_AMT1          | 0.13 |
| 18_JOESUPADUPALOG:PAY_AMT3                    | 0.13 |
| 14_JOESUPADUPALOG:PAY_5                       | 0.10 |
| 2_JoeSupaDupaRoundDecimals:BILL_AMT2          | 0.09 |
| 17_PAY_AMT2                                   | 0.09 |
| 13_JoeSupaDupaQuantileWinsorizer0.01:PAY_4    | 0.09 |
| 5_JOESUPADUPALOG:BILL_AMT5                    | 0.08 |
| 19_PAY_AMT4                                   | 0.08 |
| 21_JoeSupaDupaQuantileWinsorizer0.01:PAY_AMT6 | 0.07 |



# More BYOR Examples

# Time Series - Better Outcomes with Multiple Models

## Compare Experiments

X

### EXPERIMENT SUMMARY: DAI+arima+prophet

Experiment: corenifa (320066564-a2d5-11e9-9aa5-0242ac110002)  
Version: 1.7.0, 2019-07-10 07:56  
Settings: 7/5/6, seed=645805723, GPUs enabled  
Train data: [walmart\\_tts\\_small\\_train.csv](#) (73165, 11)  
Validation data: N/A  
Test data: [walmart\\_tts\\_small\\_test.csv](#) (16280, 9)  
Target column: [Weekly\\_Sales](#) (regression)  
System specs: Docker/Linux, 240 GB, 32 CPU cores, 4/4 GPUs  
Max memory usage: 25.6 GB, 5.13 GB GPU  
Recipe: [AutoDL](#) (63 iterations, 8 individuals)  
Validation scheme: time-based, 3 internal holdouts  
Feature engineering: 1115 features scored (12 selected)  
Timing:  
Data preparation: 84.97 secs  
Shift/Leakage detection: 43.61 secs  
Model and feature tuning: 1657.45 secs (145 of 312 models trained)  
Feature evolution: 3346.19 secs (807 models trained)  
Final pipeline training: 2990.87 secs (1 model trained)  
Python / MOJO scorer building: 54.08 secs / 0.00 secs  
Validation score: MAE = 5137.9 +/- 75.203 (baseline)  
Validation score: MAE = 3849.3 +/- 84.571 (final pipeline)  
Test score: MAE = 1743.8 +/- 27.666 (final pipeline)

**MAE = 1744 (DAI+Prophet+ARIMA)**

### VARIABLE IMPORTANCE

|  |      |
|--|------|
| 56_MYPARALLELPROPHET:Date:Dept:Store   | 1.00 |
| 16_MYPARALLELAUTOARIMA:Date:Dept:Store | 0.05 |
| 11_Date-get_dayofyear                  | 0.03 |
| 100_TargetLog:Date:Dept:Store:52       | 0.03 |
| 11_Date-get_day                        | 0.03 |
| 11_Date-get_week                       | 0.01 |
| 100_TargetLog:Date:Dept:Store:32       | 0.01 |
| 100_TargetLog:Date:Dept:Store:40       | 0.01 |
| 11_Date-get_month                      | 0.01 |
| 100_TargetLog:Date:Dept:Store:44       | 0.01 |
| 100_TargetLog:Date:Dept:Store:45       | 0.00 |

### EXPERIMENT SUMMARY: Prophet

Experiment: giruwuke (e41d5272-a2bf-11e9-9aa5-0242ac110002)  
Version: 1.7.0, 2019-07-10 04:54  
Settings: 7/5/6, seed=73039948, GPUs enabled  
Train data: [walmart\\_tts\\_small\\_train.csv](#) (73165, 11)  
Validation data: N/A  
Test data: [walmart\\_tts\\_small\\_test.csv](#) (16280, 9)  
Target column: [Weekly\\_Sales](#) (regression, log-transformed)  
System specs: Docker/Linux, 240 GB, 32 CPU cores, 4/4 GPUs  
Max memory usage: 11.7 GB, 4.36 GB GPU  
Recipe: [AutoDL](#) (36 iterations, 8 individuals)  
Validation scheme: time-based, 3 internal holdouts  
Feature engineering: 1 features scored (1 selected)  
Timing:  
Data preparation: 112.49 secs  
Shift/Leakage detection: 5.26 secs  
Model and feature tuning: 1739.17 secs (46 of 312 models trained)  
Feature evolution: 1961.69 secs (210 of 624 models trained)  
Final pipeline training: 2540.43 secs (1 model trained)  
Python / MOJO scorer building: 34.02 secs / 0.00 secs  
Validation score: MAE = 5500.7 +/- 160.28 (baseline)  
Validation score: MAE = 5487.5 +/- 173.41 (final pipeline)  
Test score: MAE = 2060.1 +/- 30.778 (final pipeline)

**MAE = 2060 (Facebook Prophet)**

### VARIABLE IMPORTANCE

|                                     |      |
|-------------------------------------|------|
| 0_MYPARALLELPROPHET:Date:Dept:Store | 1.00 |
|                                     |      |
|                                     |      |
|                                     |      |
|                                     |      |

### EXPERIMENT SUMMARY: AutoARIMA

Experiment: vililibudu (17a25b84-a2bf-11e9-9aa5-0242ac110002)  
Version: 1.7.0, 2019-07-10 04:41  
Settings: 7/5/6, seed=434547769, GPUs enabled  
Train data: [walmart\\_tts\\_small\\_train.csv](#) (73165, 11)  
Validation data: N/A  
Test data: [walmart\\_tts\\_small\\_test.csv](#) (16280, 9)  
Target column: [Weekly\\_Sales](#) (regression, sqrt-transformed)  
System specs: Docker/Linux, 240 GB, 32 CPU cores, 4/4 GPUs  
Max memory usage: 11.2 GB, 5.39 GB GPU  
Recipe: [AutoDL](#) (34 iterations, 8 individuals)  
Validation scheme: time-based, 3 internal holdouts  
Feature engineering: 1 features scored (1 selected)  
Timing:  
Data preparation: 69.13 secs  
Shift/Leakage detection: 4.03 secs  
Model and feature tuning: 1760.41 secs (46 of 312 models trained)  
Feature evolution: 2367.08 secs (177 of 624 models trained)  
Final pipeline training: 1666.89 secs (1 model trained)  
Python / MOJO scorer building: 72.93 secs / 0.00 secs  
Validation score: MAE = 4565.1 +/- 672.31 (baseline)  
Validation score: MAE = 4620.4 +/- 673.97 (final pipeline)  
Test score: MAE = 3442.1 +/- 56.167 (final pipeline)

**MAE = 3442 (ARIMA)**

### VARIABLE IMPORTANCE

|                                       |      |
|---------------------------------------|------|
| 0_MYPARALLELAUTOARIMA:Date:Dept:Store | 1.00 |
|                                       |      |
|                                       |      |
|                                       |      |
|                                       |      |

# NLP: Building More Text Features with Recipes

H<sub>2</sub>O.ai

h2oai / driverlessai-recipes

Unwatch 64 Unstar 56 Fork 24

Code Issues 3 Pull requests 2 Actions Projects 0 Wiki Security Insights

Branch: master ➔ driverlessai-recipes / transformers / nlp /

Create new file Upload files Find file History

| SudalaiRajkumar  | update readme file   | Latest commit d046622 2 days ago |
|--|--|----------------------------------|
| ..   |  |                                  |
| <a href="#">fuzzy_text_similarity_transformers.py</a>    | Upgrade version  | 3 months ago                     |
| <a href="#">text_embedding_similarity_transformer.py</a> | Disable not fully tested / OOM prone recipes.                    | 2 months ago                     |
| <a href="#">text_lang_detect_transformer.py</a>          | Cosmetics.   | 3 months ago                     |
| <a href="#">text_meta_transformers.py</a>                | More desc  | 3 months ago                     |
| <a href="#">text_pos_tagging_transformer.py</a>          | count of pos tag features for text                               | 2 days ago                       |
| <a href="#">text_preprocessing_transformer.py</a>        | update readme file   | 2 days ago                       |
| <a href="#">text_readability_transformers.py</a>         | readability feature from text transformers                       | 21 days ago                      |
| <a href="#">text_sentiment_transformer.py</a>            | More desc  | 3 months ago                     |
| <a href="#">text_similarity_transformers.py</a>          | formatting   | last month                       |
| <a href="#">text_spelling_correction_transformer.py</a>  | Disabled spell correct transformer due to acceptance test issues | 3 days ago                       |
| <a href="#">text_topic_modeling_transformer.py</a>       | update readme and doc strings                                    | 14 days ago                      |
| <a href="#">text_url_summary_transformer.py</a>          | Added multiprocessing to speed up url parse                      | 12 days ago                      |
| <a href="#">vader_text_sentiment_transformer.py</a>      | Update vader_text_sentiment_transformer.py                       | 29 days ago                      |

# Data Augmentation - Local Holidays

Branch: master ▾ [driverlessai-recipes](#) / [transformers](#) / [augmentation](#) / [singapore\\_public\\_holidays.py](#)

[Find file](#) [Copy path](#)

 arnocandel Move back

38b4669 on 22 Jun

1 contributor

109 lines (99 sloc) | 5.56 KB

[Raw](#) [Blame](#) [History](#)  

```
1 """Flag for whether a date falls on a public holiday in Singapore."""
2 from h2oaicore.transformer_utils import CustomTransformer
3 import datatable as dt
4 import numpy as np
5 import pandas as pd
6
7
8 # https://github.com/rjchow/singapore_public_holidays
9 def make_holiday_frame():
10     return dt.fread(
11         """
12             Date,Name,Day,Observance,Observance Strategy
13             2016-01-01,New Year's Day,Friday,2016-01-01,actual_day
14             2016-02-08,Chinese New Year Day 1,Monday,2016-02-08,actual_day
15             2016-02-09,Chinese New Year Day 2,Tuesday,2016-02-09,actual_day
16             2016-03-25,Good Friday,Friday,2016-03-25,actual_day
17             2016-05-01,Labour Day,Sunday,2016-05-01,next_monday
18             2016-05-21,Vesak Day,Saturday,2016-05-21,actual_day
19             2016-07-06,Hari Raya Puasa,Wednesday,2016-07-06,actual_day
20             2016-08-09,National Day,Tuesday,2016-08-09,actual_day
21             2016-09-12,Hari Raya Haji,Monday,2016-09-12,actual_day
22             2016-10-29,Deepavali,Saturday,2016-10-29,actual_day
23             2016-12-25,Christmas Day,Sunday,2016-12-26,next_monday
24             2017-01-01,New Year's Day,Sunday,2017-01-02,next_monday
25             2017-01-28,Chinese New Year Day 1,Saturday,2017-01-28,actual_day
26             2017-01-29,Chinese New Year Day 2,Sunday,2017-01-30,next_monday
```

***"We are pleased with the recipe feature added to H2O Driverless AI. We can be more creative in how we evaluate and serve those new to credit with the ability to customize and extend the platform to meet our unique needs."***

Yan Yang, VP of Data Science  
Deserve





# Thank you!

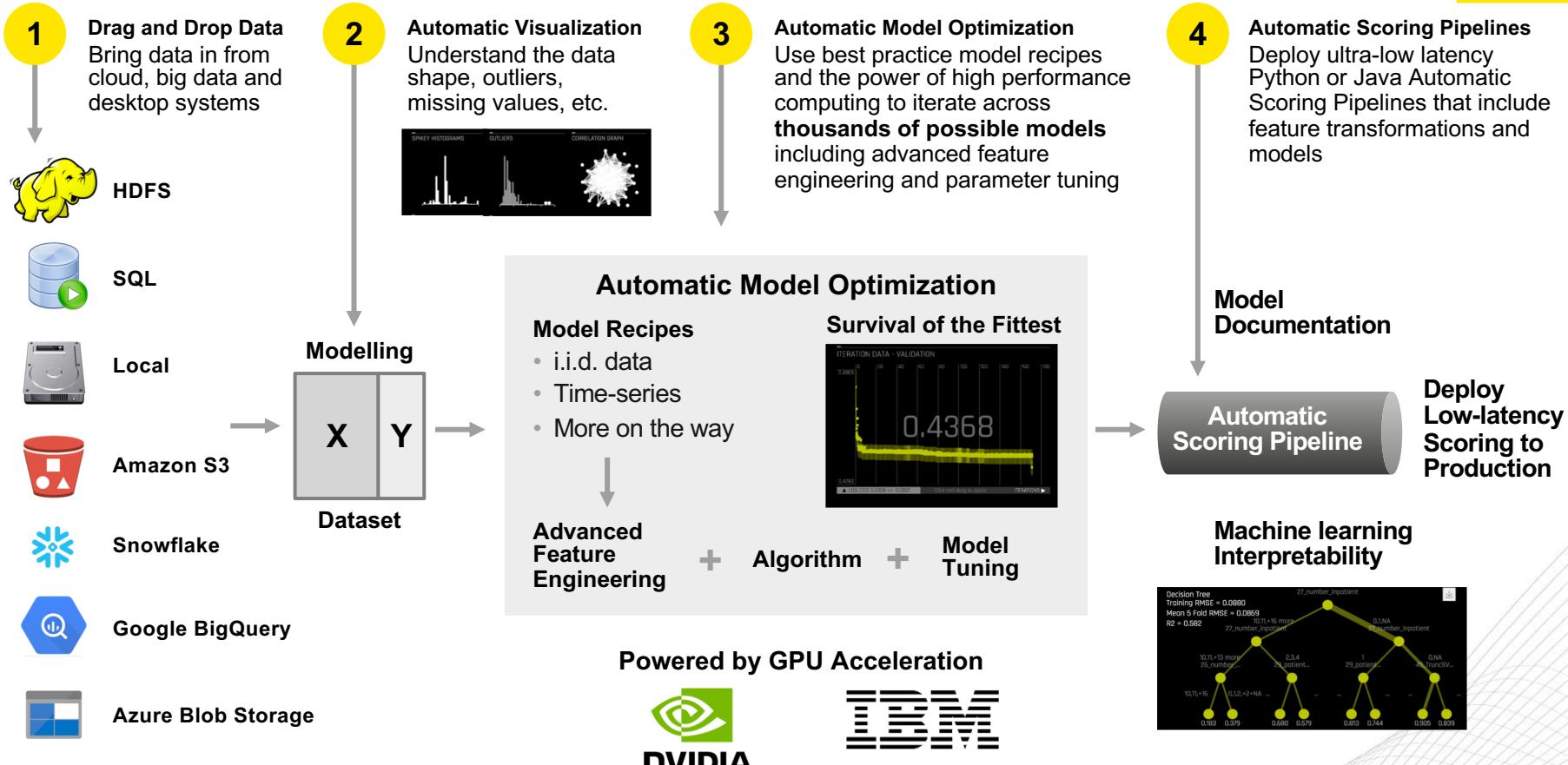
Slides: [bit.ly/h2o\\_meetups](https://bit.ly/h2o_meetups)

[www.h2o.ai/try-driverless-ai](http://www.h2o.ai/try-driverless-ai)

Want to give a talk? [joe@h2o.ai](mailto:joe@h2o.ai)

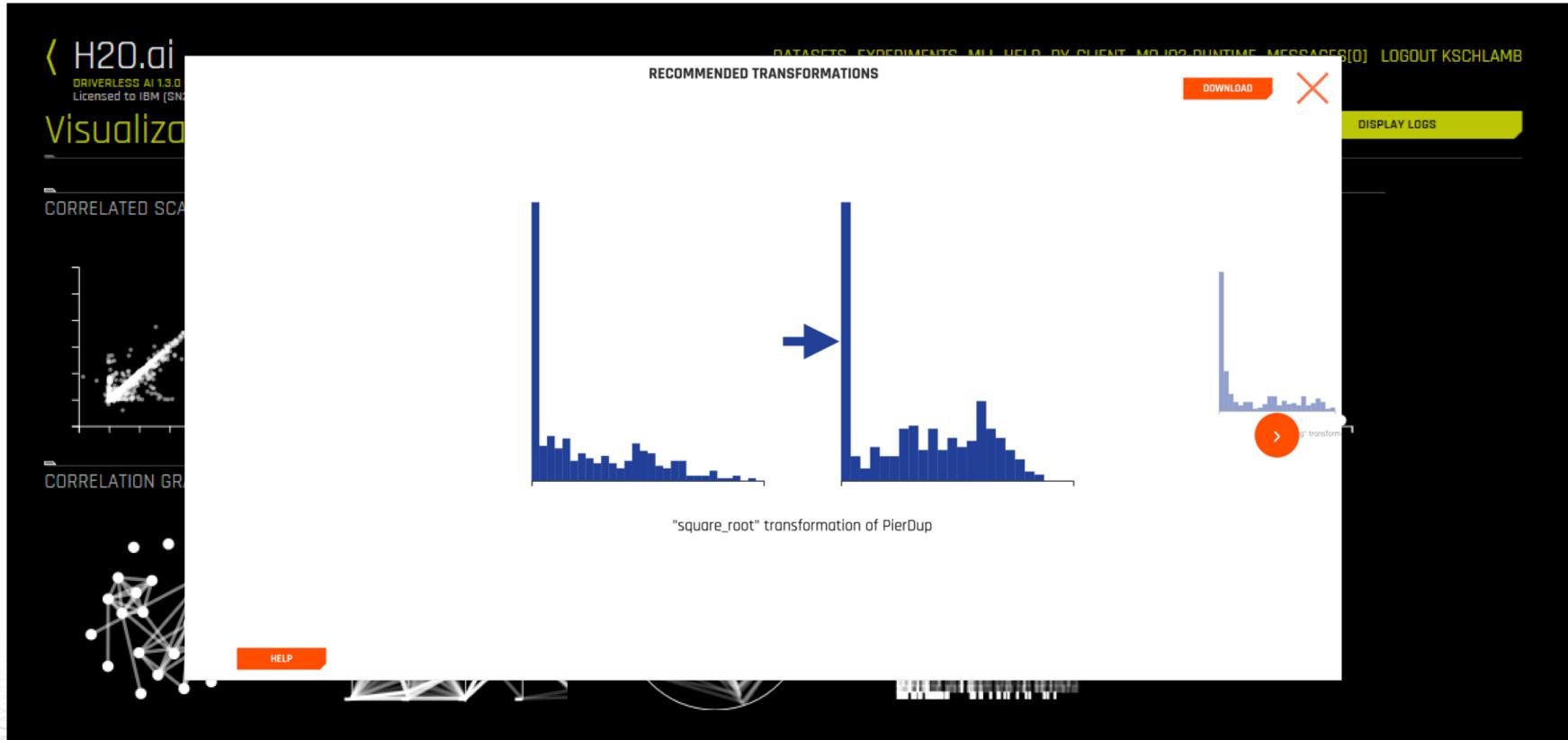
# Appendix

# H2O Driverless AI – How it Works



# Other Improvements in Version 1.7

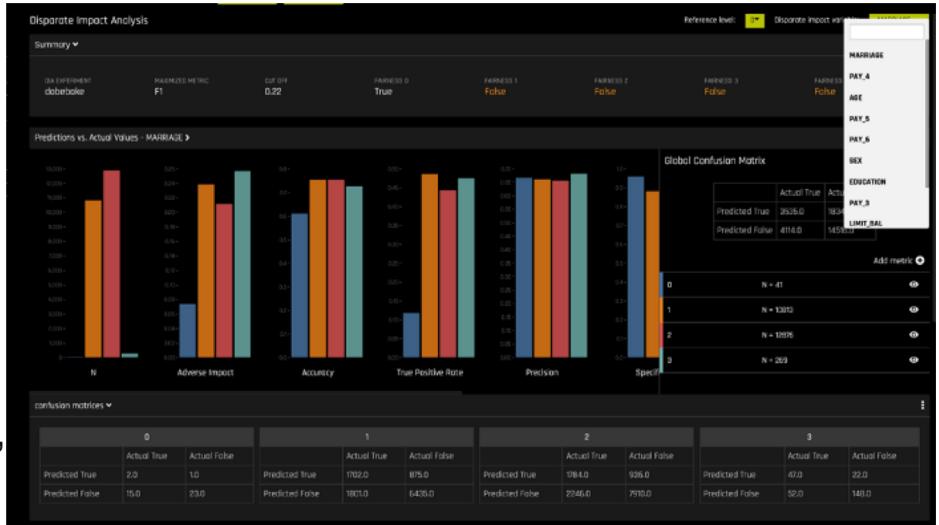
# Recommendations for columnar transformations in AutoViz



# H2O Driverless AI's **MLI module**

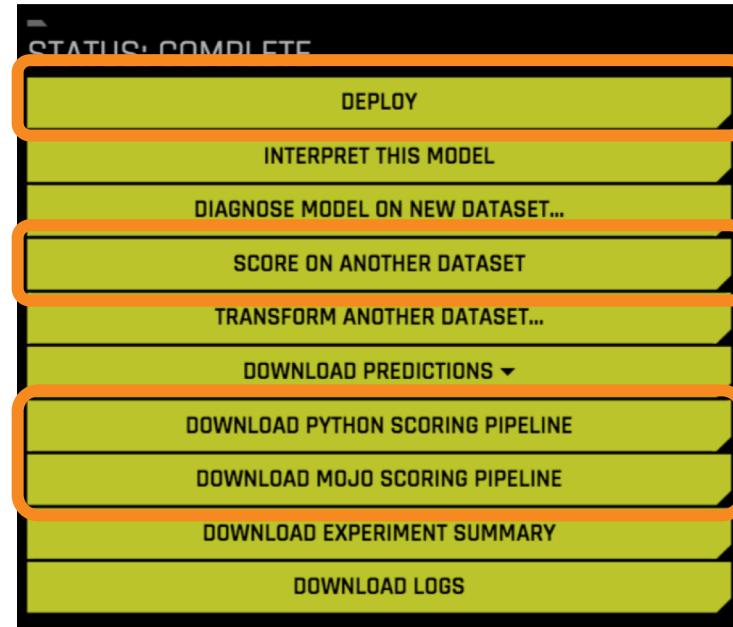
# Disparate Impact Analysis

- Measure quantitatively the adverse treatment of protected classes
- Why?
  - Mitigating Bias in AI/ML Models
  - For Example
    - Discrimination in hiring, housing, etc., or in general any public policy decisions
- Key tool that is broadly applicable to a wide variety of use cases under the regulatory compliance umbrella, especially around intentional discrimination.
  - Home Mortgage Disclosure Act (HMDA )
  - Federal Protections in the Mortgage Marketplace (ECOA/FHA)



# Scoring Pipeline

- Deploy as REST Service, Lambda or into Sagemaker
- Python
- MOJO (Java)
- C++ (R/Python bindings)



# Deployment Options – Local REST Server

The screenshot shows the H2O.ai Experiment interface with the project name "kihipera". The top navigation bar includes links for PROJECTS, DATASETS, AUTOVIZ, EXPERIMENTS, DIAGNOSTICS, MLI, DEPLOYMENTS, RESOURCES, MESSAGES (3), and LOGOUT. A yellow banner at the top right displays the H<sub>2</sub>O.ai logo.

The main content area shows deployment details for the experiment "kihipera". The "Deployment for:" dropdown is set to "kihipera". A modal window titled "Endpoint info" provides specific deployment information:

- API Key: N/A - local rest server
- Location: <http://localhost:7767/models>
- Model ID: 1e3e9a68-a0bf-11e9-b493-0242ac110002

Two "CLOSE" buttons are visible at the bottom of the modal window.