Introduction to Driverless AI

Agenda

- Introduction
  - Setting Up Qwiklabs
  - Driverless AI Overview
- Credit Card Experiment
  - Deep Dive into Driverless AI Experiment
  - Deep Dive into Model Interpretability
- Amazon Fine Food Reviews Experiment
  - Deep Dive into Text Feature Engineering
- Washington State Daily Cannabis Sales Experiment
  - Deep Dive into Time Series Analysis

Bring Your Own Data

Analyze your own tabular data using supervise learning if you have it!

Setting Up Qwiklabs

Qwiklabs Instructions

1. Go to: https://h2oai.qwiklab.com/

2. Click “Join” in upper right hand corner to create an account
   - If you have already done a Qwiklabs you can click “Sign In”

3. Go to the “Catalog” view

4. Click on the lab: “Introduction to Driverless AI (1 GPU)”

5. Click Start Lab

6. Enter License

Driverless AI Overview

H2O Products

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

H2O AI Open Source Engine Integration with Spark

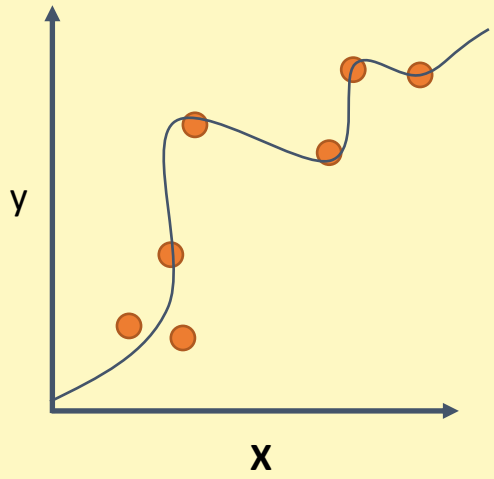
Lightning Fast machine learning on GPUs

Automatic feature engineering, machine learning and interpretability

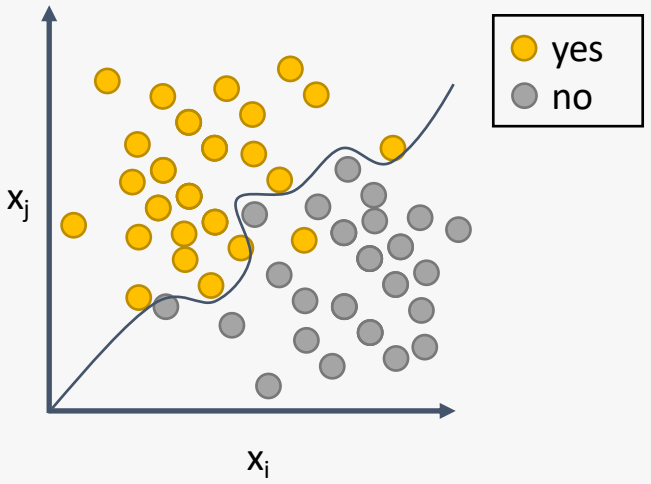
Secure multi-tenant H2O clusters

Supervised Learning

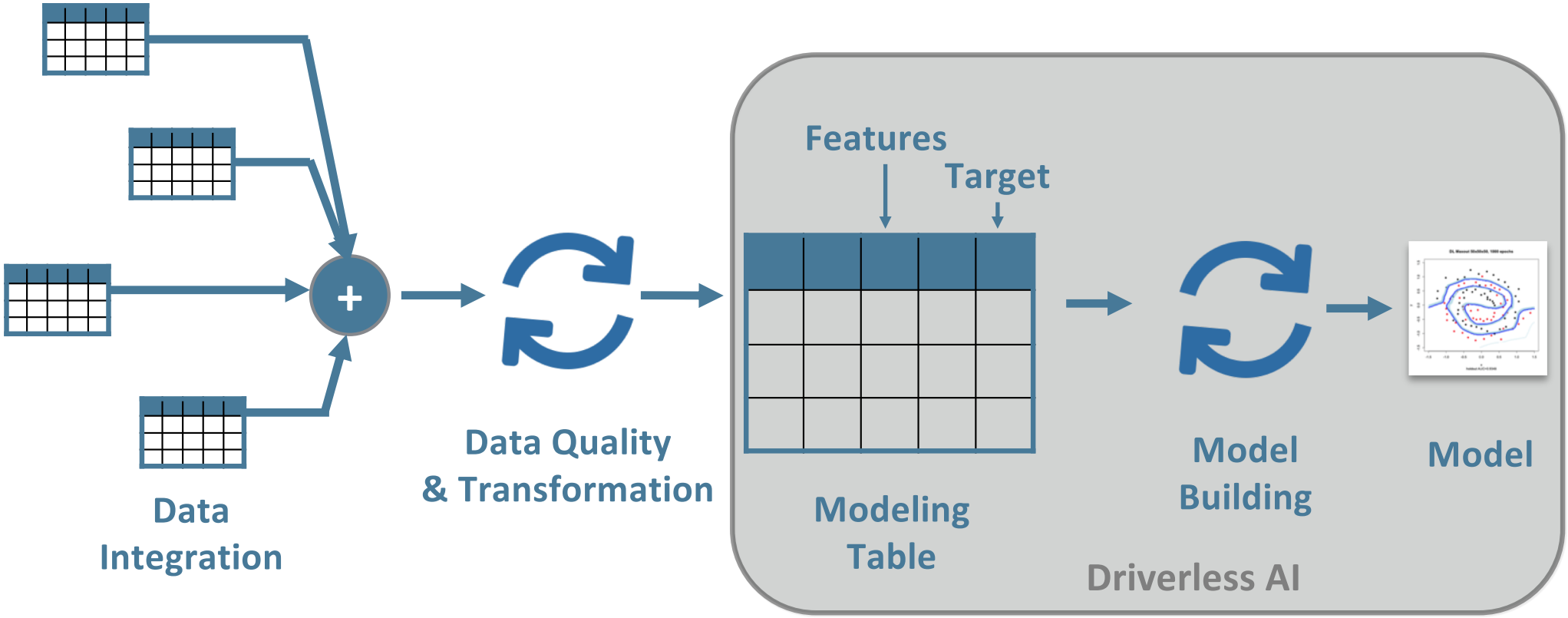
Regression:
How much will a customer spend?



Classification:
Will a customer churn?



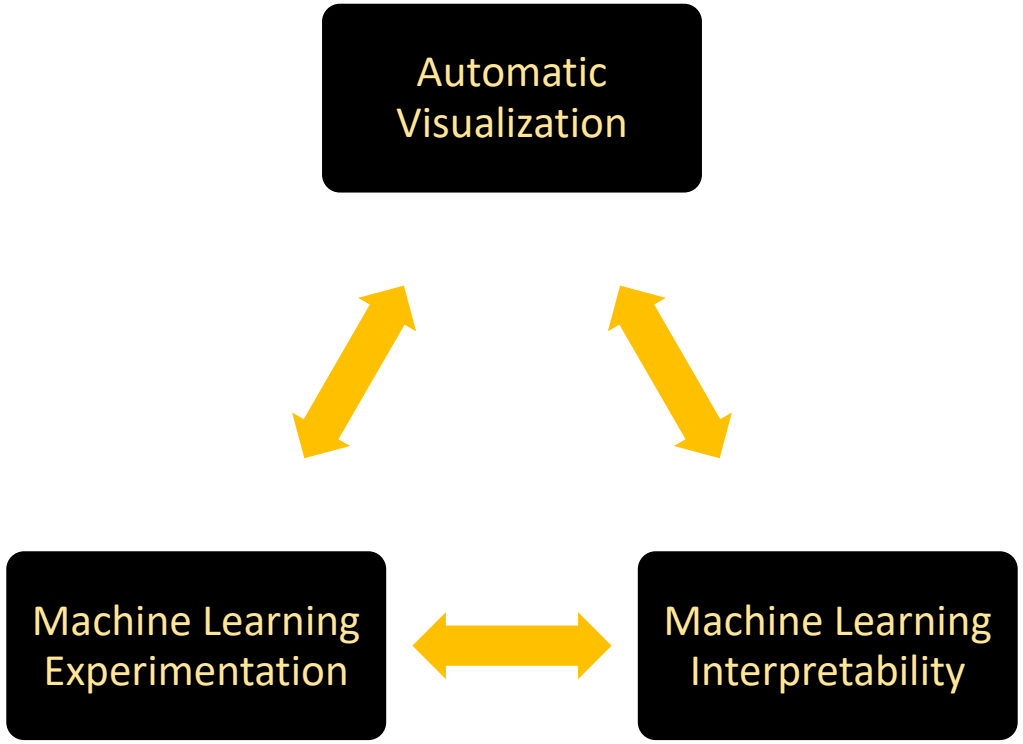
Typical Enterprise Machine Learning Workflow



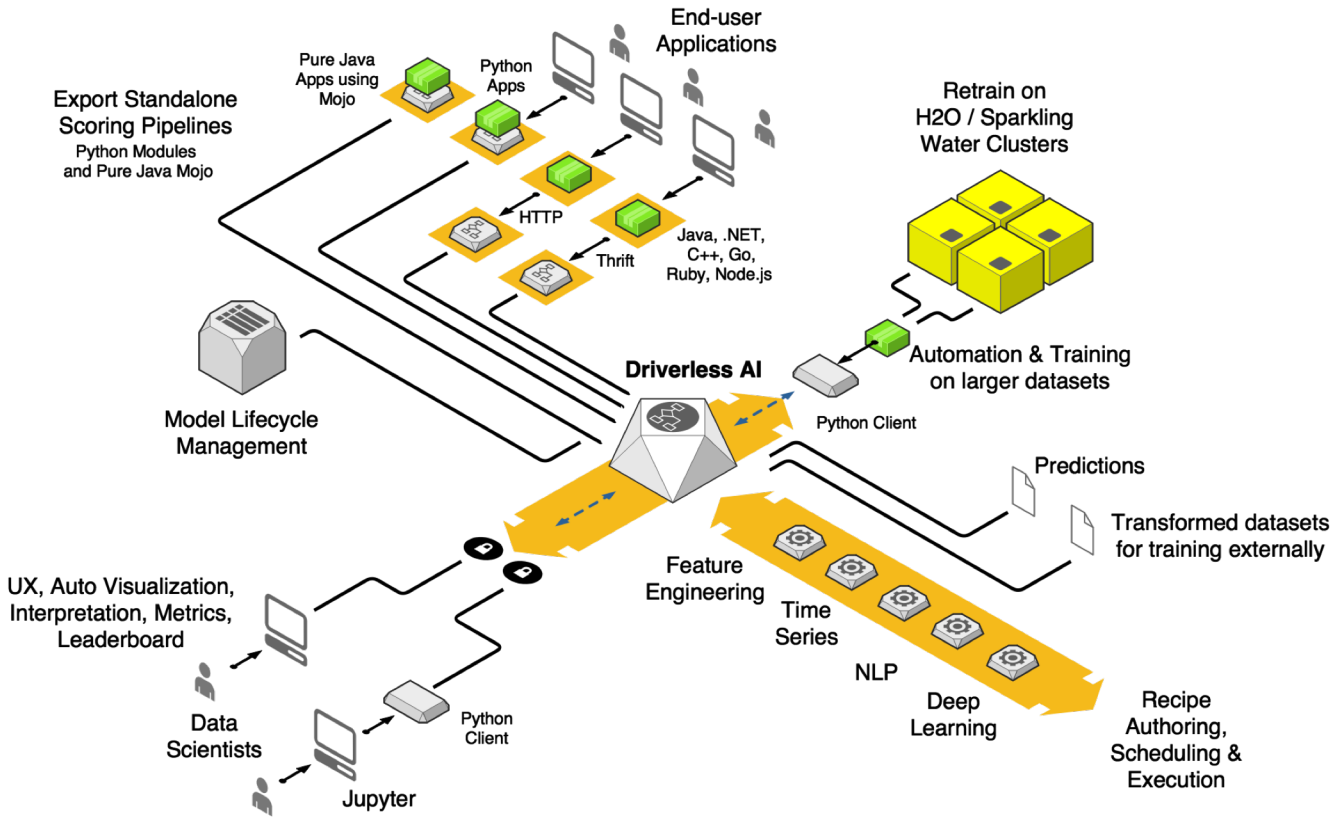
Problems Addressed by Driverless AI

- Supervised Learning
  - Regression
  - Classification
    - Binary
    - Multinomial
- Tabular structured data
  - Numeric
  - Categorical
  - Time/Date
  - Text
  - Missing values are allowed
- Identically and Independently Distributed (iid) rows
- Time-series
  - Single time-series
  - Grouped time-series
  - Time-series problems with a gap between training and testing to account for time to deploy

Driverless AI Components

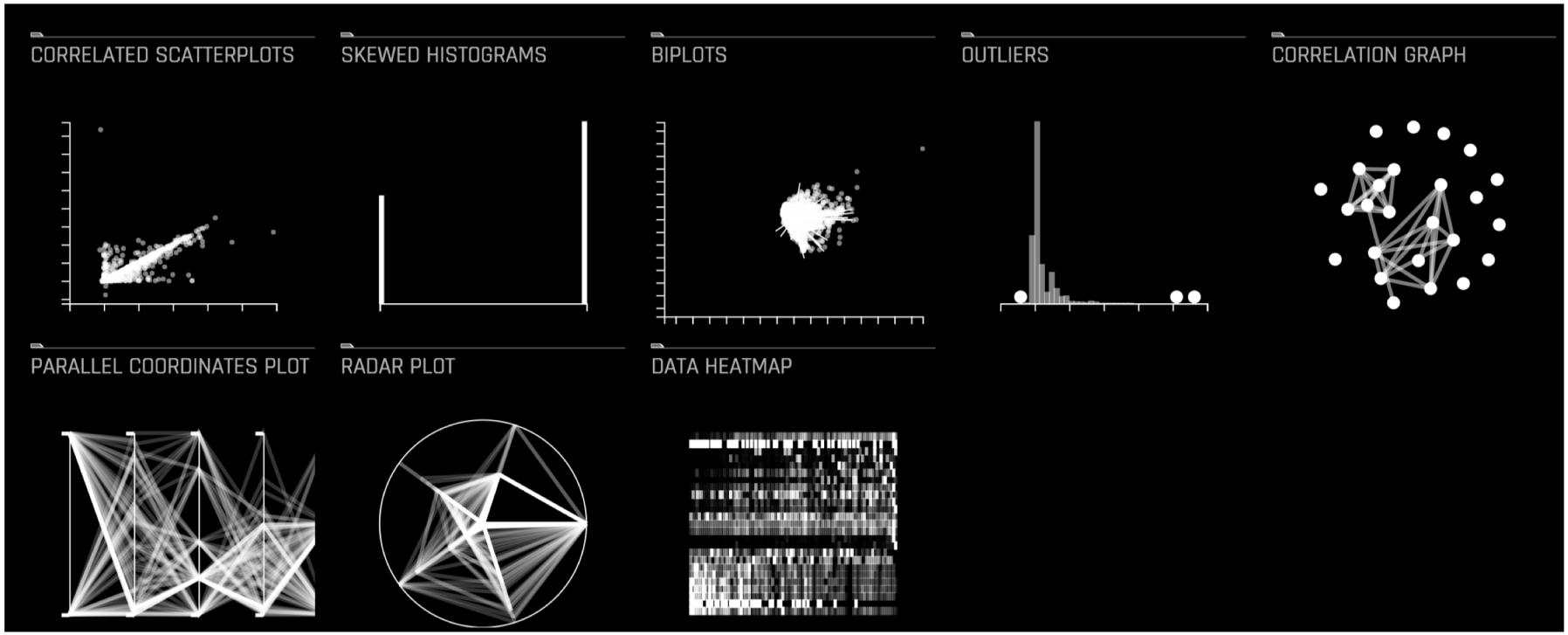


Driverless Architecture



Automatic Visualizations

Automatic Visualizations



Automatic Visualizations

- Clumpy Scatterplots
- Correlated Scatterplots
- Unusual Scatterplots
- Spikey Histograms
- Skewed Histograms
- Varying Boxplots
- Heteroscedastic Boxplots
- Biplots
- Outliers
- Correlation Graph
- Parallel Coordinates Plot
- Radar Plot
- Data Heatmap
- Missing Values Heatmap

Machine Learning Experimentation

Driverless AI

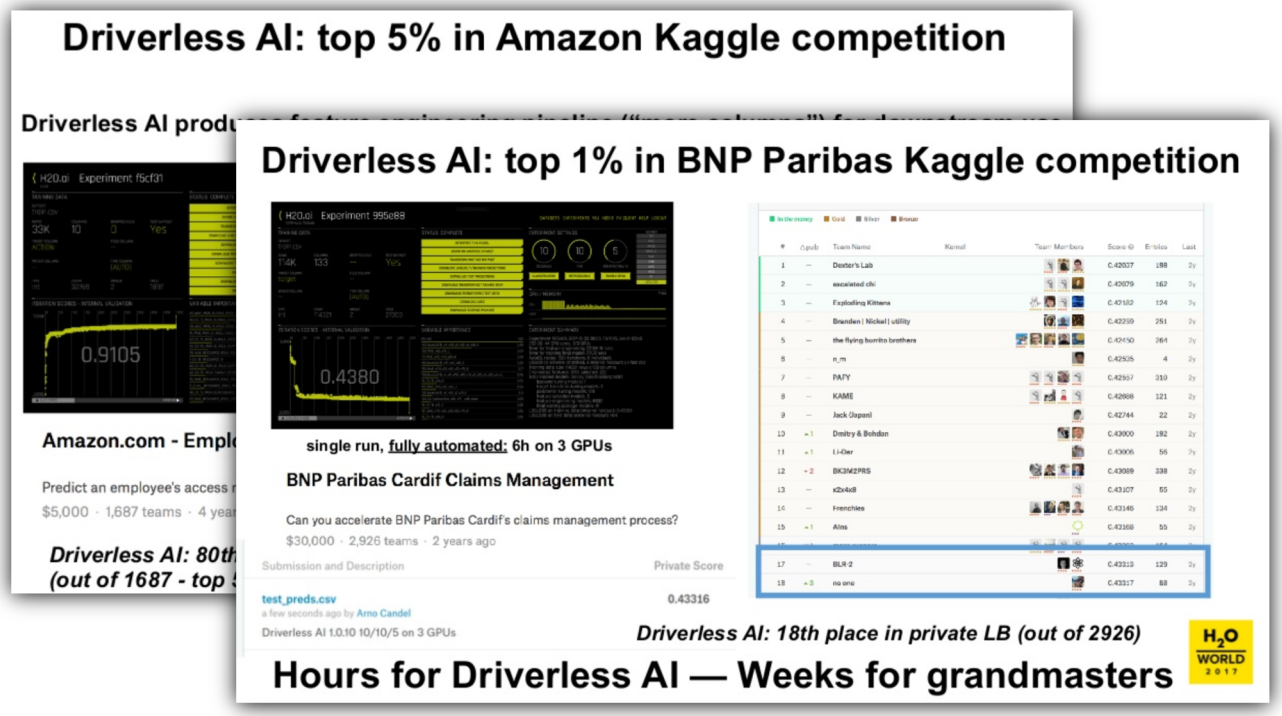
Accuracy

Time

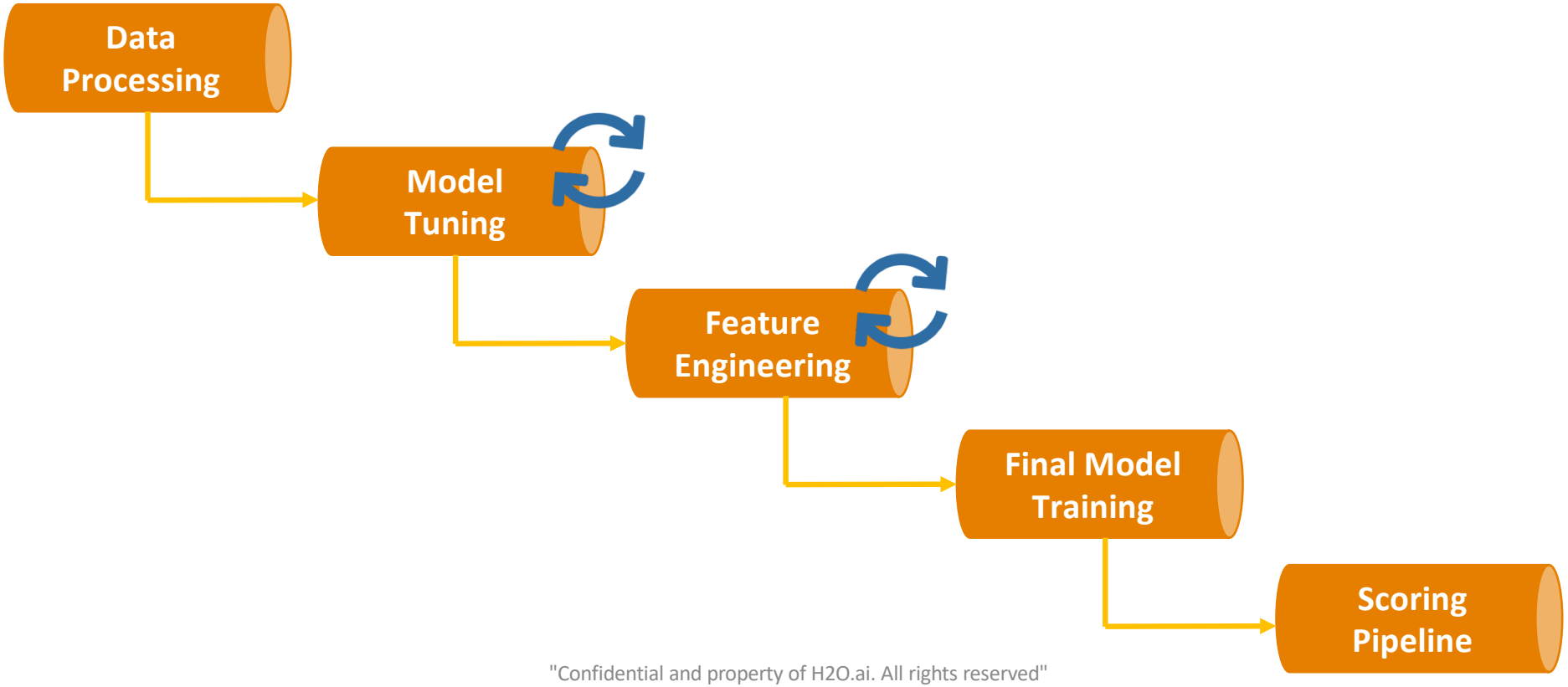
Interpretability

Accuracy

- Automatic feature engineering to increase accuracy - AlphaGo for AI
- Automatic Kaggle Grandmaster recipes in a box for solving wide variety of use-cases
- Automatic machine learning to find and tune the right ensemble of models



Driverless AI Workflow



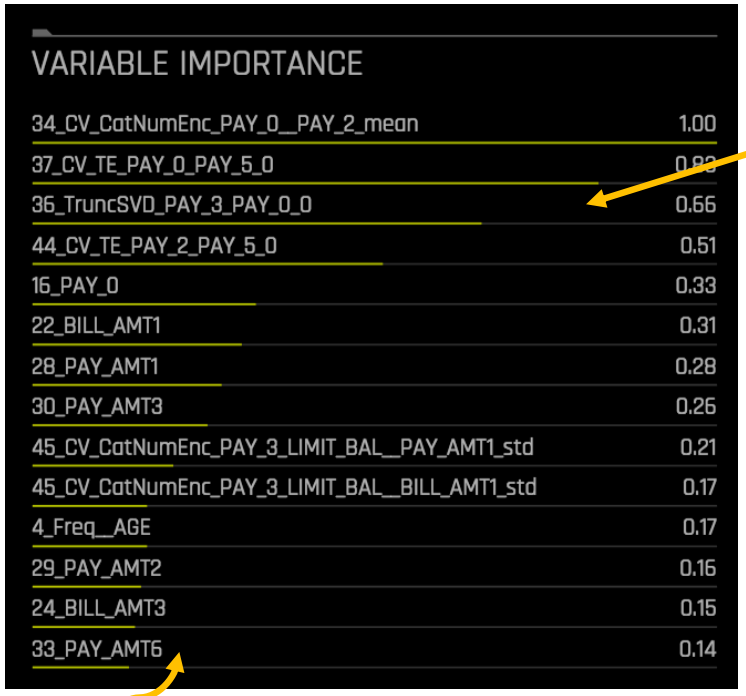
Will I Take the Day Off?

Date	Off
Jan. 23, 2016	Yes
Apr. 20, 2016	No
Jun. 7, 2016	No
Nov. 24, 2016	Yes
Nov. 28, 2016	Yes

Will I Take the Day Off?

Month	Day	Year	DayOfWeek	Off
1	23	2016	Sat	Yes
4	20	2016	Wed	No
6	7	2016	Tues	No
11	24	2016	Thurs	Yes
11	28	2016	Mon	Yes

Auto Feature Generation
Kaggle Grand Master Out of the Box



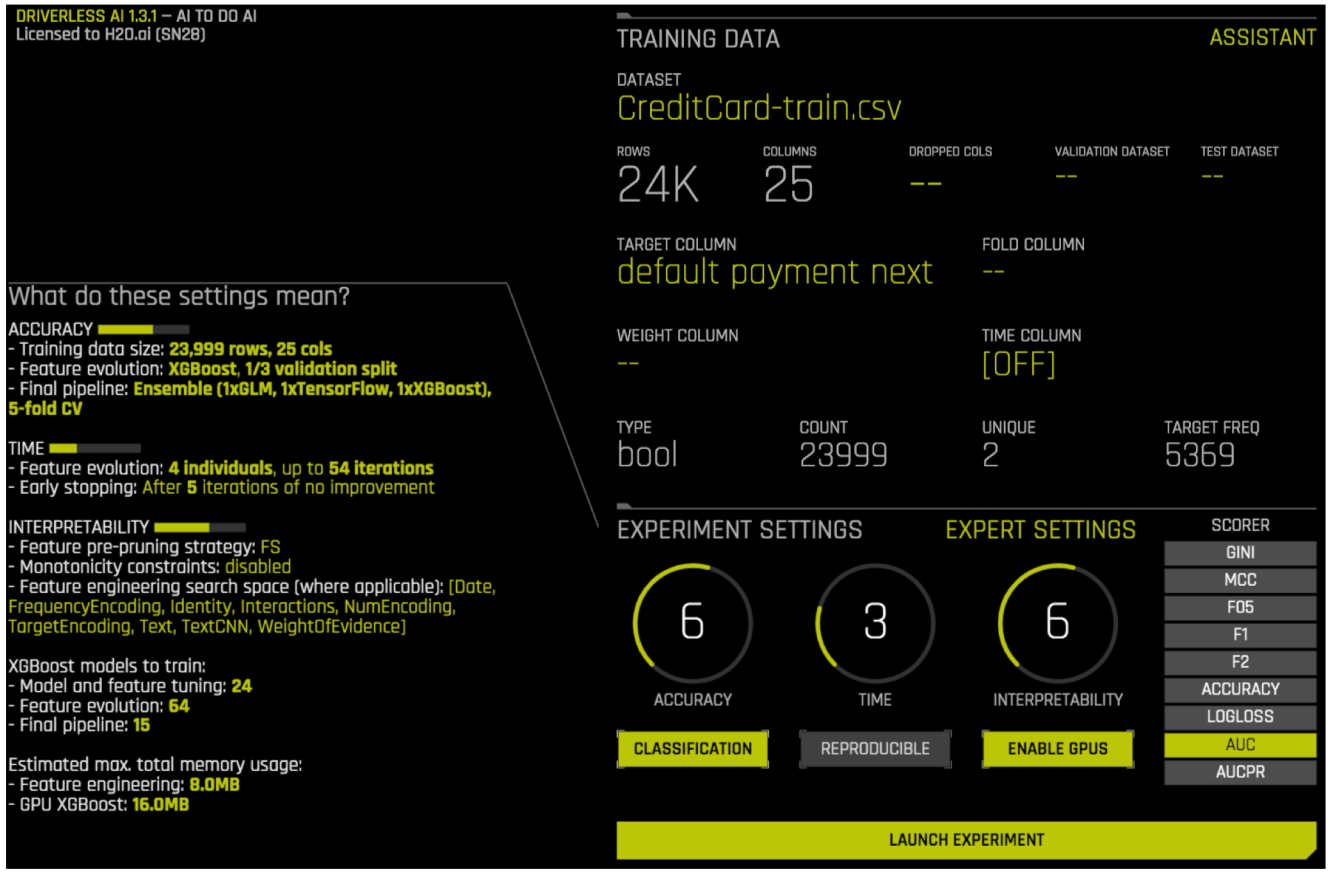
Generated Features

Original Features

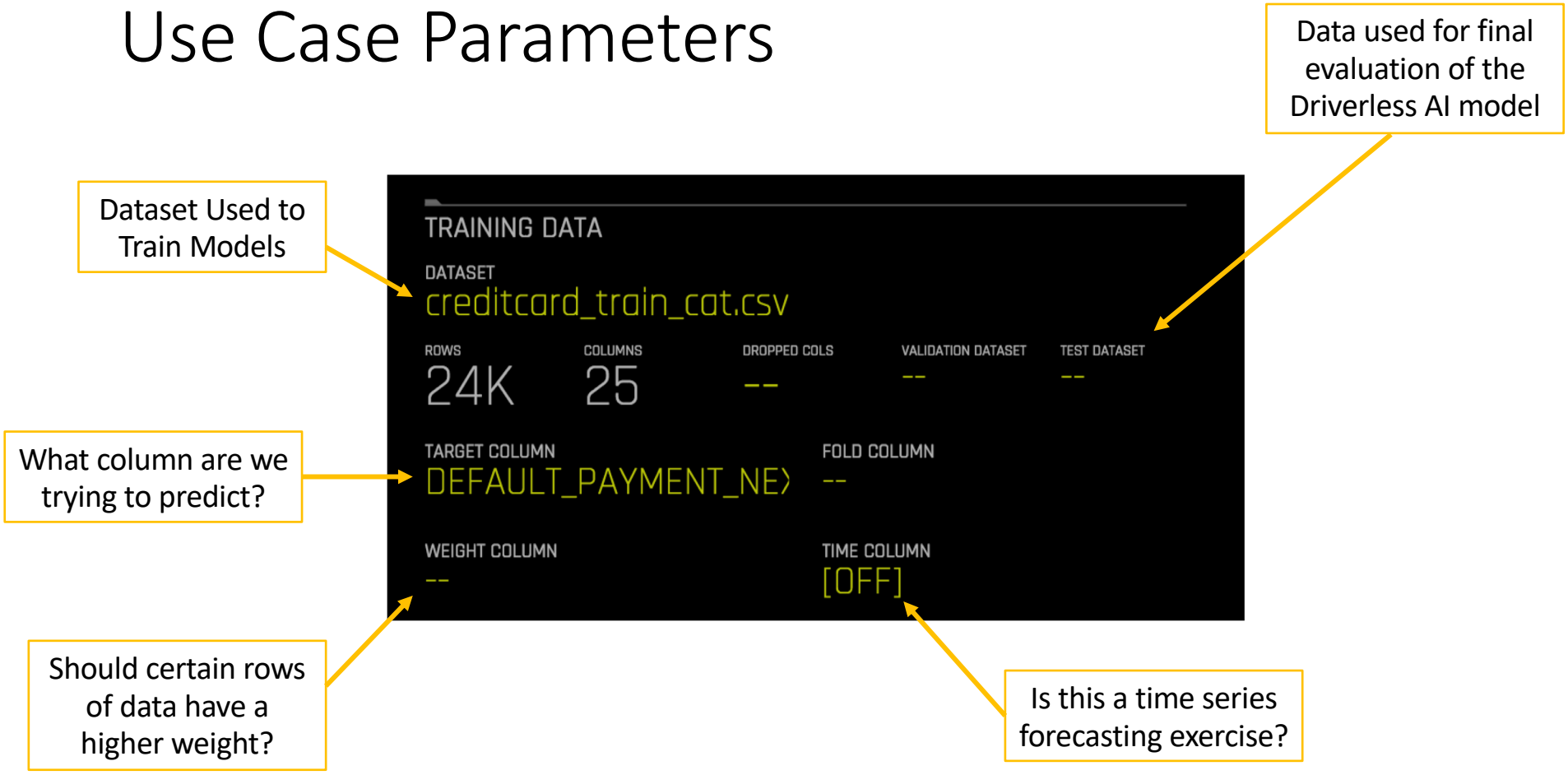
Feature Transformations

- Automatic Text Handling
- Frequency Encoding
- Cross Validation Target Encoding
- Truncated SVD
- Clustering and more

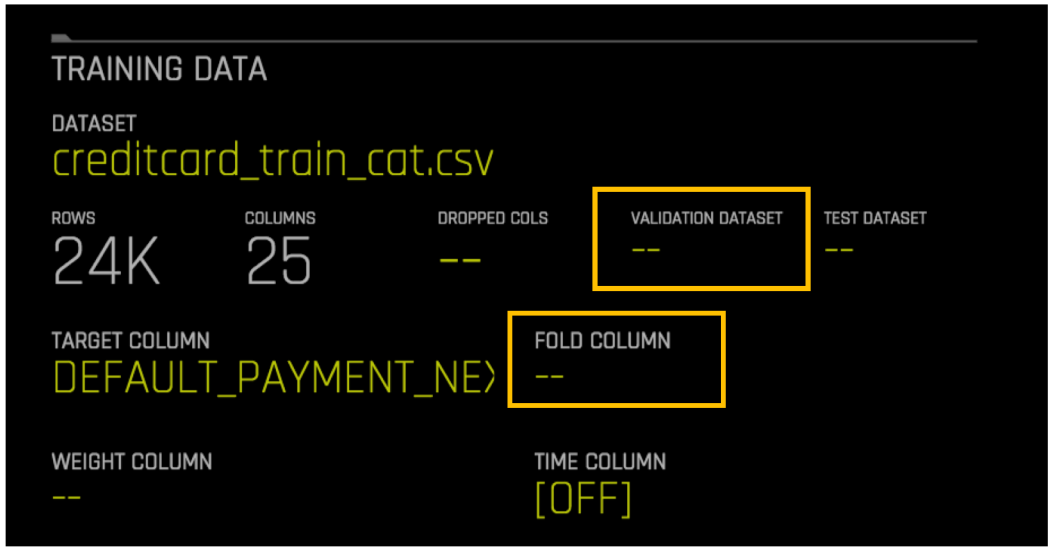
Running an Experiment



Use Case Parameters

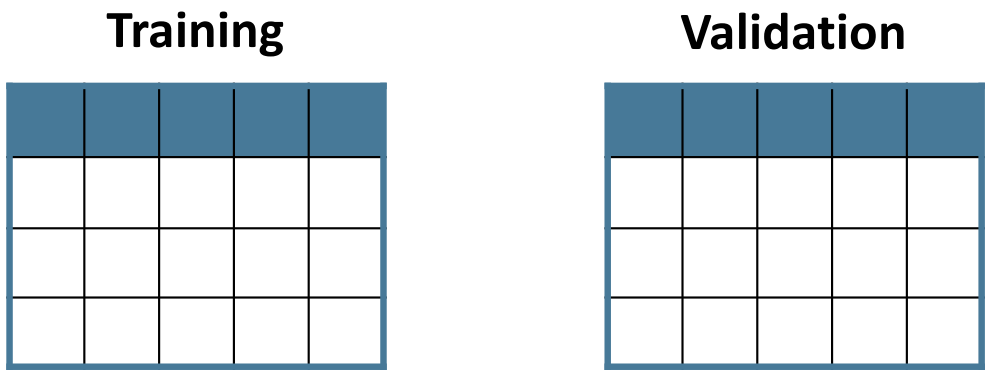


Internal Validation Parameters



Internal Validation

- User provides validation dataset
- Provide Validation Dataset when you have shifting data distributions
  - Helps improve generalization



Internal Validation

- User provides fold column
- Provide fold column when you want each fold to have a specific distribution
  - For example, if you want to validate on one country and train on all others

Training

				Fold Column

Internal Validation

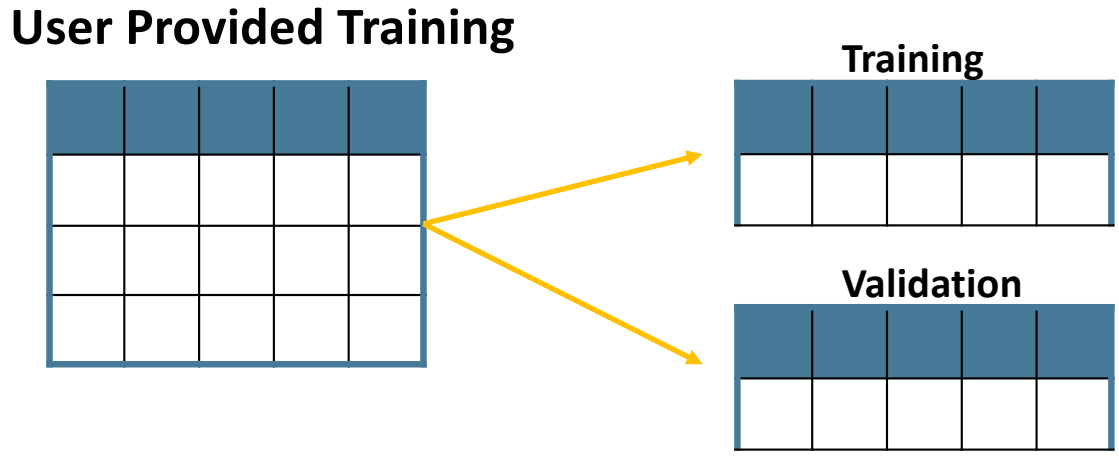
- User provides time column
- Provide time column when you want to train on historical data and validate on more recent data
  - If Time Column is set to “AUTO”, Driverless AI will try to auto detect any time ordering

Training

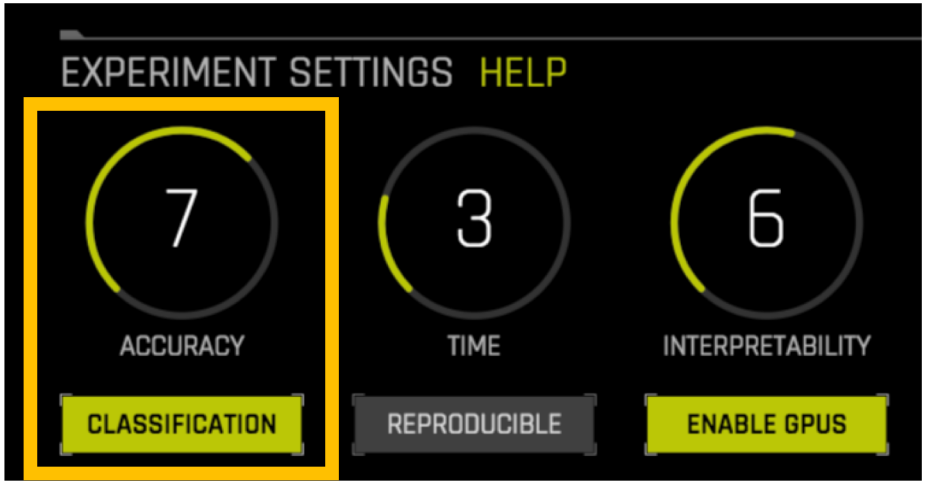
				Time Column

Internal Validation

- Default – validation data, fold column, and time column not provided
- Driverless AI internally splits the data (possibly doing cross validation depending on accuracy setting)



Experiment Settings



Accuracy

Accuracy	Max Rows x Cols	Ensemble Level	Target Transformation	Parameter Tuning Level	Num Folds	Only First Fold Model	Distribution Check
1	100K	0	False	0	3	True	No
2	1M	0	False	0	3	True	No
3	50M	0	True	1	3	True	No
4	100M	0	True	1	3-4	True	No
5	200M	1	True	1	3-4	True	Yes
6	500M	2	True	1	3-5	True	Yes
7	750M	<=3	True	2	3-10	Auto	Yes
8	1B	<=3	True	2	4-10	Auto	Yes
9	2B	<=3	True	3	4-10	Auto	Yes
10	10B	<=4	True	3	4-10	Auto	Yes

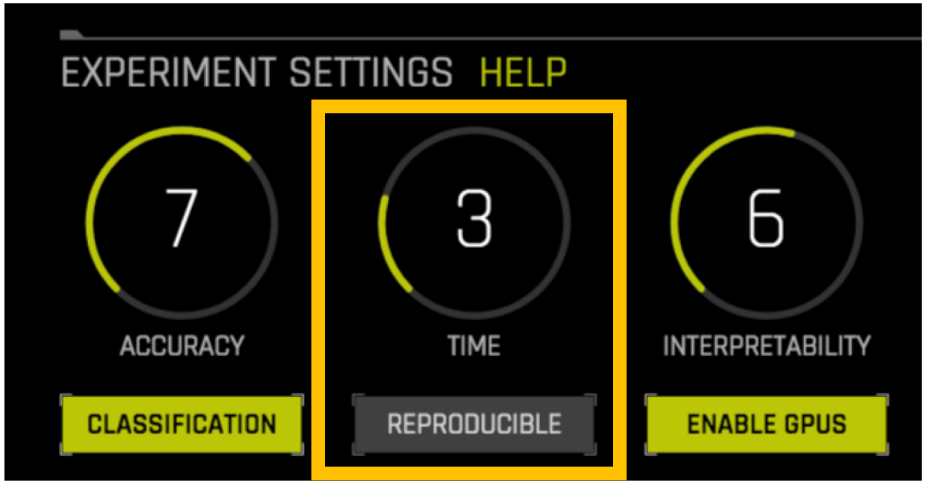
Accuracy

- Training Data Size
  - For low accuracy, data will be down-sampled
- Ensemble Level
  - How many models are used in the final ensemble?
- Target Transformation
  - Would the performance of the model improve if we transformed the target variable using a transformation determined by the interpretability setting?
- Parameter Tuning Level
  - How many models are used to tune the model parameters?
- Genetic Algorithm
  - How many feature combinations can should be explored?

Accuracy

- Cross-Validation Folds
  - How many validation splits should be used during cross-validation?
- Only First Fold Model
  - Should only first fold be used for internal validation?
- Early Stopping Rounds
  - How many rounds should be used in early stopping rule?
- Distribution Check
  - Should Driverless AI check to see if the distribution of the training data diverges from the testing data?
- Feature Selection Strategy
  - Should weak features be pruned from the model?

Experiment Settings



Time

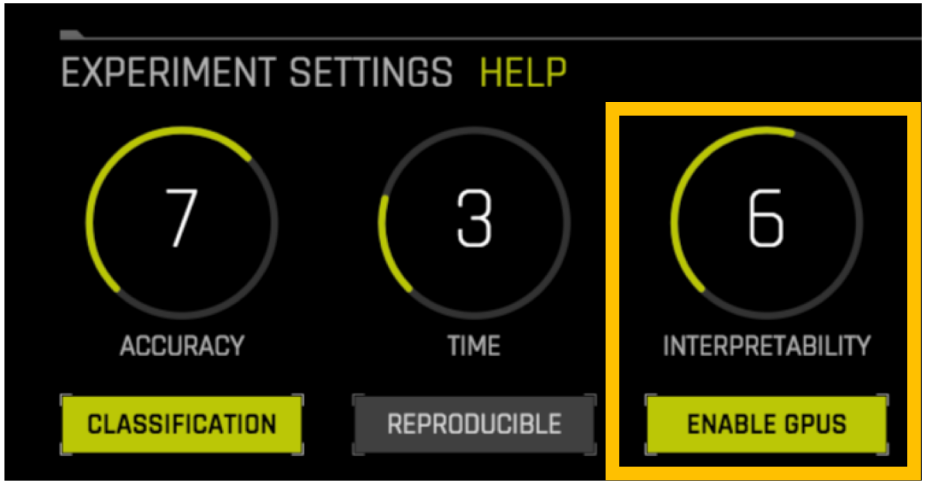
- Feature Evolution
  - How many iterations of model building should be performed to find the best set of features?

- Early Stopping
  - If Early Stopping = 5, then Driverless AI will stop performing feature engineering once the best model has not changed for the past 5 iterations

Time

Time	Iterations	Early Stopping Rounds
1	1-5	None
2	10	5
3	30	5
4	40	5
5	50	10
6	100	10
7	150	15
8	200	20
9	300	30
10	500	50

Experiment Settings



Interpretability

Interpretability	Ensemble Level	Target Transformation	Feature Engineering	Feature Pre-Pruning	Monotonicity Constraints
1 - 3	<= 3			None	Disabled
4	<= 3	Inverse		None	Disabled
5	<= 3	Anscombe	Clustering (ID, distance) Truncated SVD	None	Disabled
6	<= 2	Logit Sigmoid		Feature selection	Disabled
7	<= 2		Frequency Encoding	Feature selection	Enabled
8	<= 1	4^{th} Root		Feature selection	Enabled
9	<= 1	Square Square Root	Bulk Interactions (add, subtract, multiply, divide) Weight of Evidence	Feature selection	Enabled
10	0	Identity Unit Box Log	Date Decompositions Number Encoding Target Encoding Text (TF-IDF, Frequency)	Feature selection	Enabled

Good start

Interpretability

- Non-Time Series Feature Engineering
  - Date Decompositions
  - Text processing
    - TF-IDF
    - Frequency
  - Categorical / Integer Data
    - Number Encoding
    - Frequency Encoding
    - Cross-Validated Target Encoding
    - Cross-Validated Weight of Evidence Encoding
  - Numeric Data
    - Binary Arithmetic Operations (add, subtract, multiply, divide)
    - K-Means Clustering
      - Cluster ID
      - Distance to cluster centroid
    - Truncated SVD

Interpretability

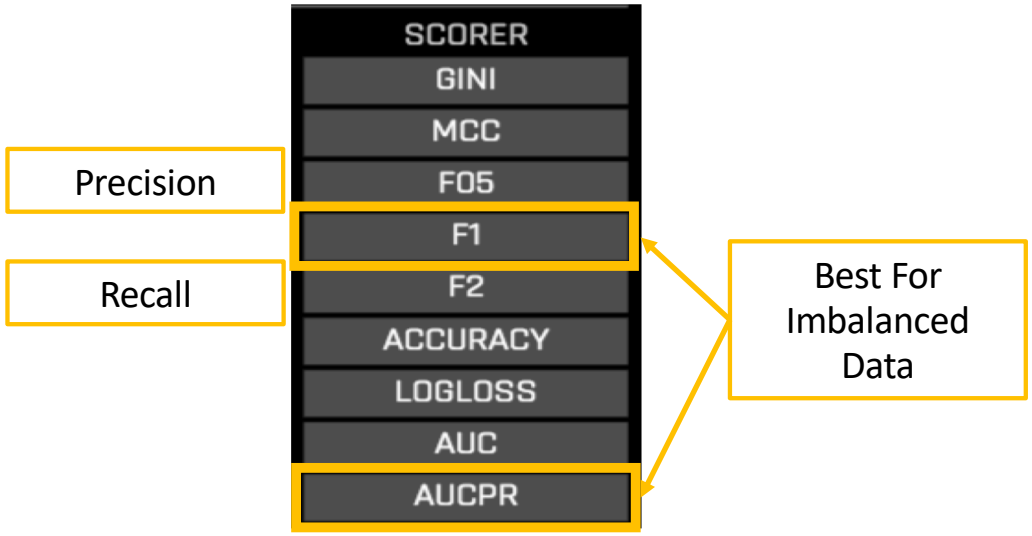
- Target Transformers
  - Only more “interpretable” target transformations are tried when interpretability knob is increased
  - High Interpretability Transformation = Identity, Log
  - Low Interpretability Transformation = Anscombe, Inverse

- Feature Pre-Pruning
  - Higher interpretability leads to more features being dropped
  - Less features means simpler model

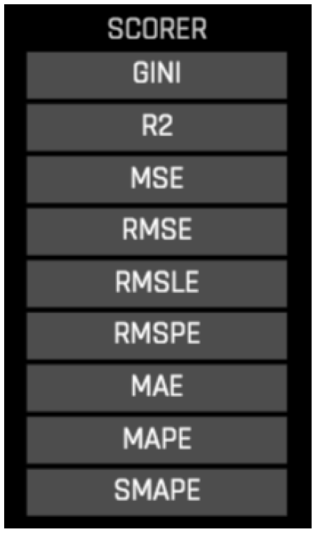
- Monotonicity
  - Do the XGBoost models built have monotonicity constraints?

Scoring Options

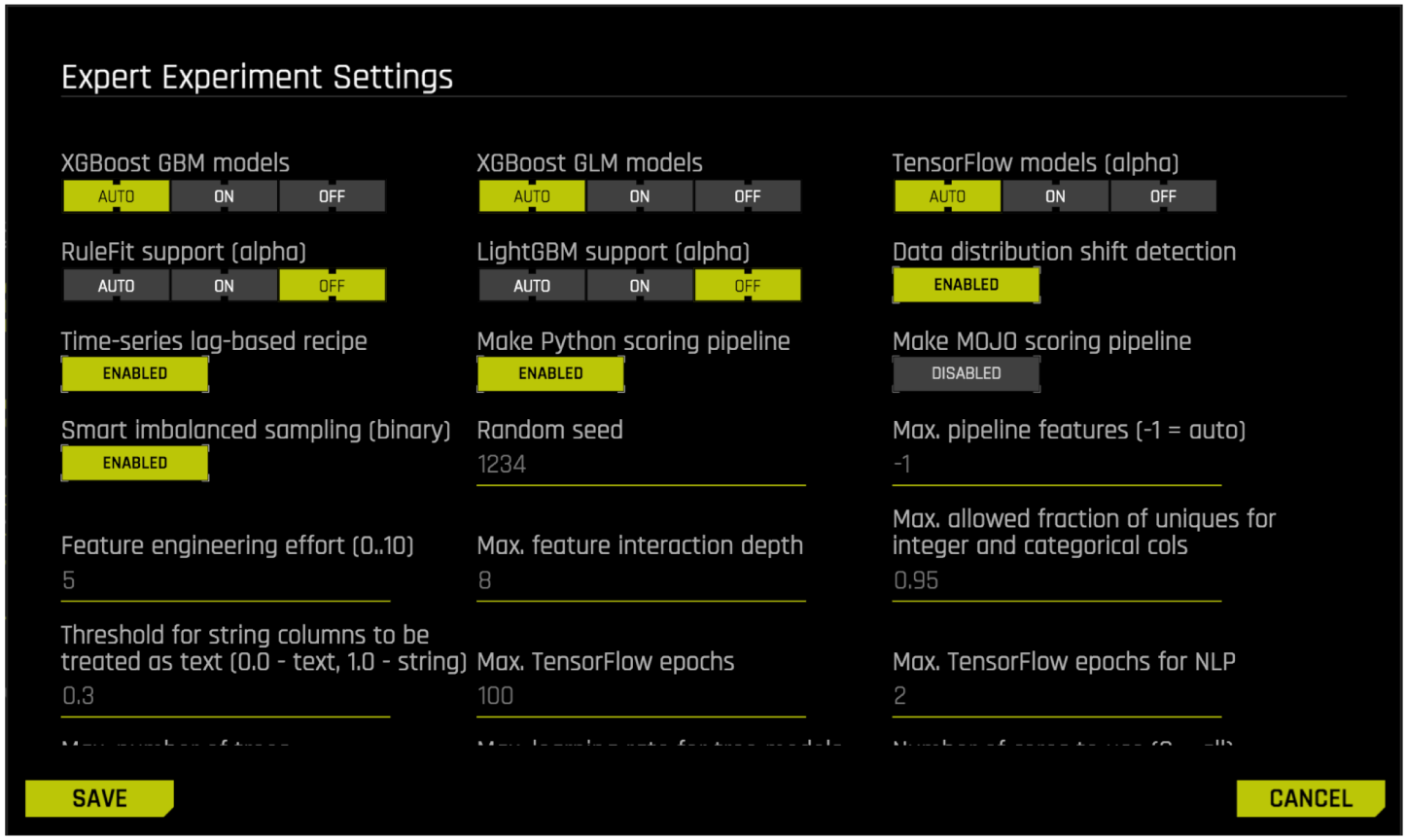
Classification



Regression



Expert Experiment Settings



Expert Experiment Settings

- GBM Model
  - XGBoost
  - LightGBM
- GLM Model
  - XGBoost
- TensorFlow Model
  - Train a neural network using TensorFlow
- Rule Fit Model
  - Train a scikit-learn GBM model
  - Decompose each tree in the GBM model into a set of rules
  - Train a GLM model on the original data plus the rules from the GBM model

Expert Experiment Settings

- Data Distribution Shift Detection
- Time Series Lag Features
- Make Python Scoring Pipeline
- Make MOJO Scoring Pipeline
- Smart Imbalance Sampling (Binary Classification)
  - Keep all rows from minority class
  - Fill out remaining rows with majority class
- Random Seed
  - When Reproducible button is enabled

Expert Experiment Settings

- Max Pipeline Features
- Feature Engineering Effort (0..10)
  - Higher values generally lead to more time (and memory) spent in feature engineering.
- Max Feature Interaction Depth
  - Used for interaction features like grouping for target encoding, weight of evidence and other likelihood estimates.
- Max Allowed Fraction of Uniques for Integer and Categorical Cols
- Threshold for String Columns to be Treated as Text

Credit Card Experiment

The Data

- Credit Card Data
  - Contains: information on default payments, demographic factors, credit data, history of payment, etc.
  - Date Range: August 2005 – September 2005
  - Reference: https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset
  - File System: /data/Kaggle/CreditCard/CreditCard-train.csv
- Our Goal: Predict whether someone will default on their credit card payment.

The Data

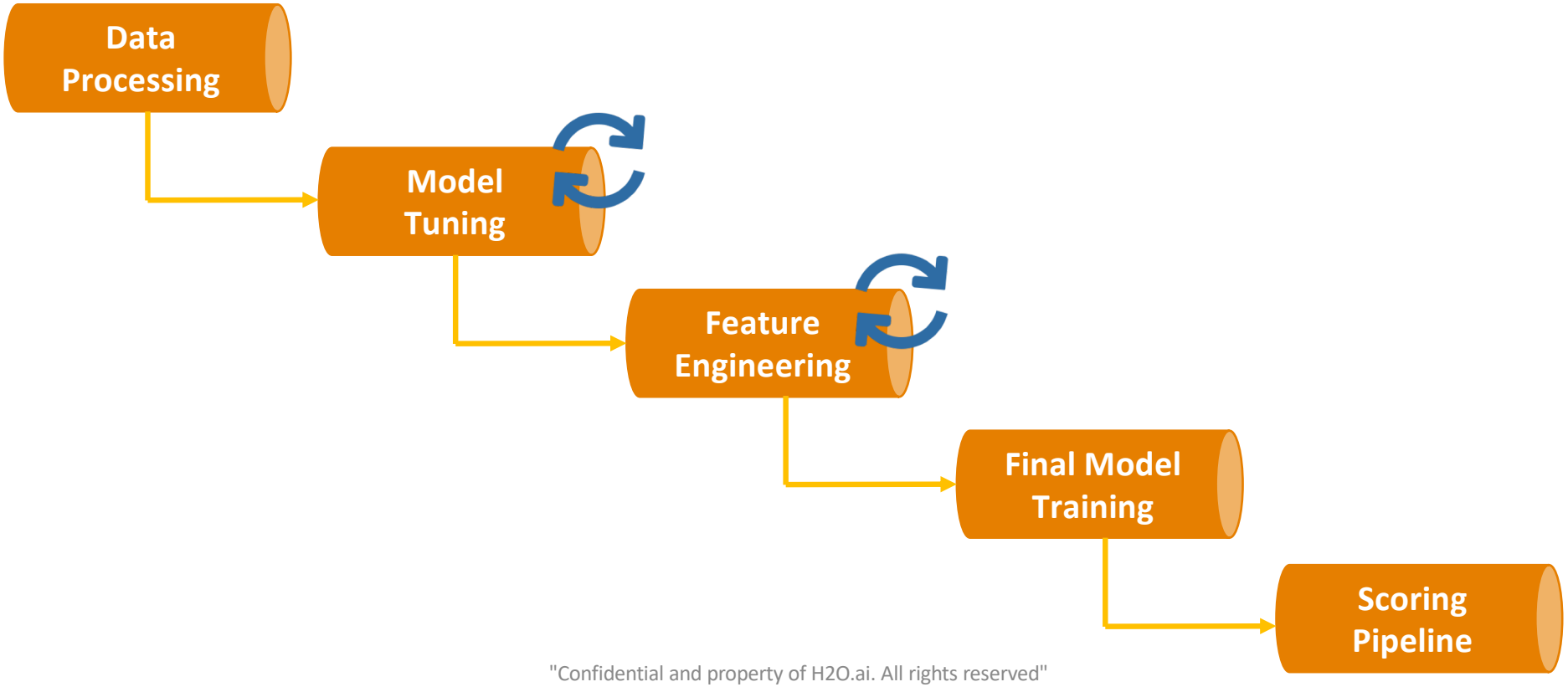
Column Name	Description
ID	ID of each client
LIMIT_BAL	Amount of given credit in NT dollars (includes individual and family/supplementary credit)
SEX	Gender (1=male, 2=female)
EDUCATION	(1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
MARRIAGE	Marital status (1=married, 2=single, 3=others)
AGE	Age in years
PAY_x {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)

The Data

LIMIT BAL	EDUCATION	AGE	PAY_1	PAY_2	BILL_AMT1	PAY_AMT1	DEFAULT PAYMENT NEXT MONTH
120,000	university	26	-1	2	2,682	0	**1**
90,000	university	34	0	0	29,239	1,418	**0**
50,000	university	37	1	0	46,990	2,000	**0**
50,000	university	37	2	0	8,617	2,000	**0**
50,000	graduate	57	3	0	64,400	2,500	**0**

Driverless AI Experiment

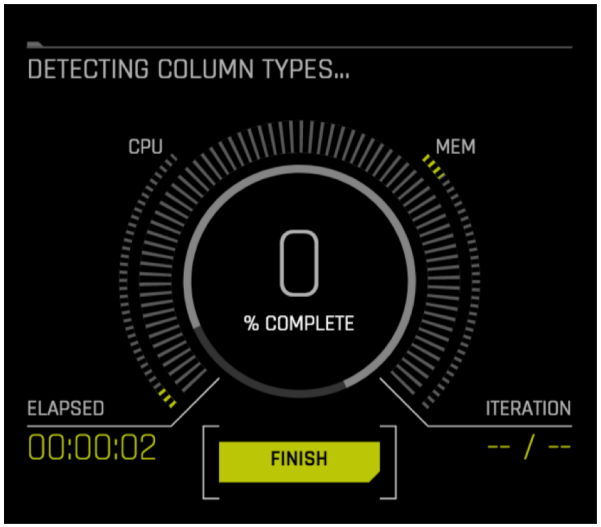
Driverless AI Workflow

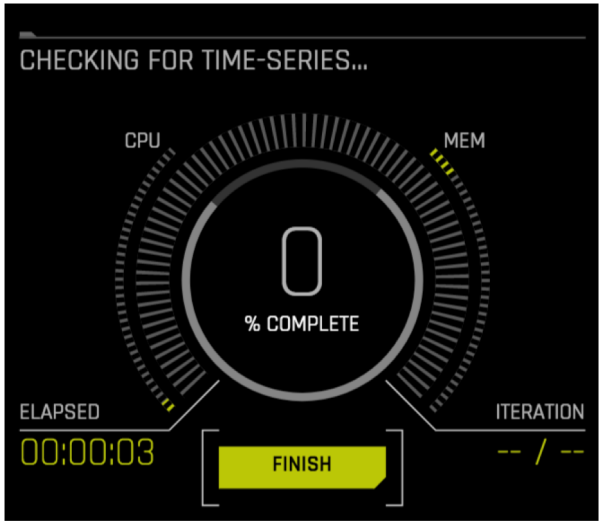


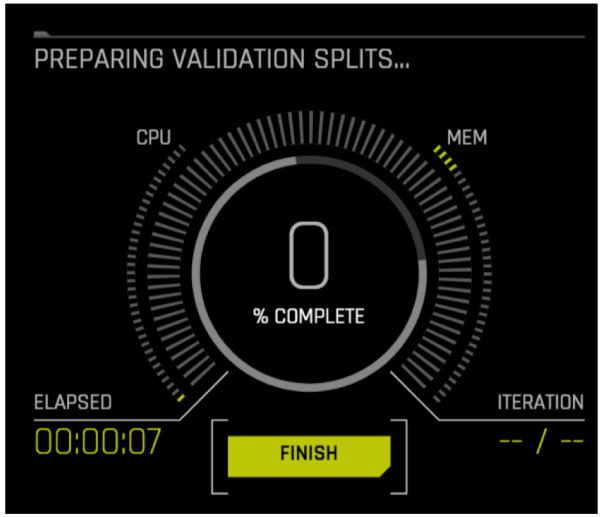
Data Processing

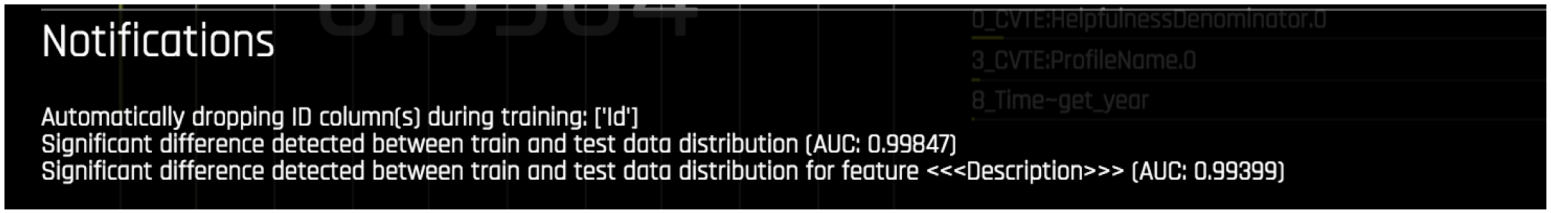
- Data Filtering
  - Columns with all missing values or all constant values are removed
  - Rows where the Target Column is NA are removed
- Detect Column Types
  - Columns can be Date, Date Time, ID, Categorical, Numeric, or Text
- Detecting Time Correlation
  - Determine if the data should be split by time or randomly

Data Processing





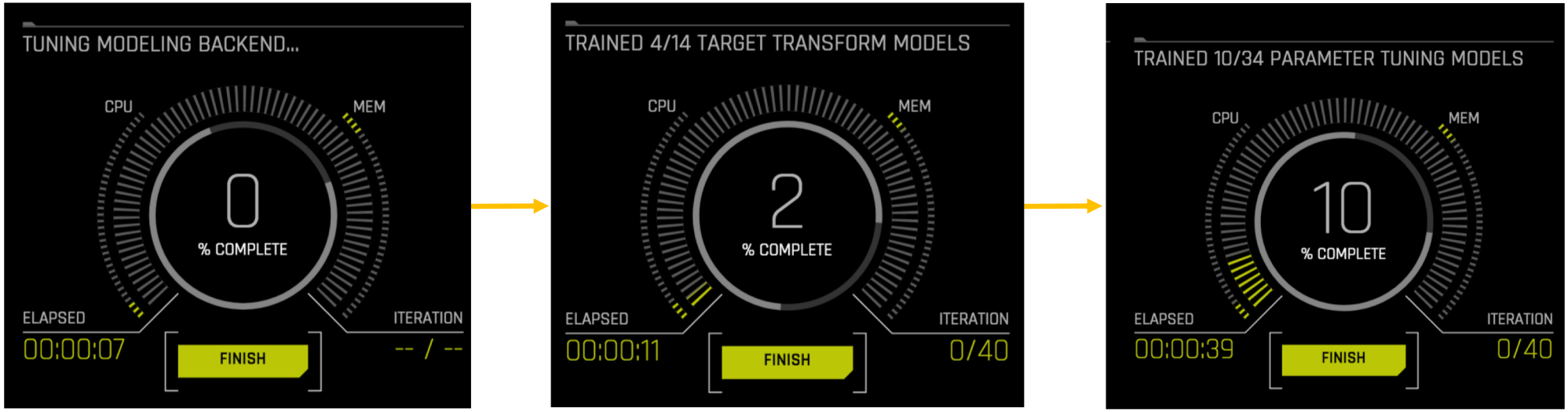




Model Tuning

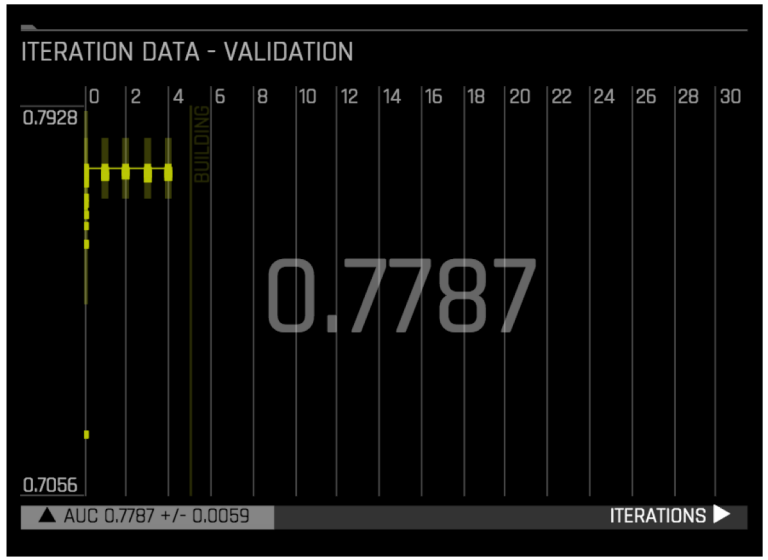
- Model Backend Tuned
  - First simple model trained – determine if GPU can be used
- Target Column Transformed
  - Target column will be transformed to see if that improves accuracy of model
- Model Parameters Tuned
  - Grid search run on model parameters

Model Tuning



Feature Engineering

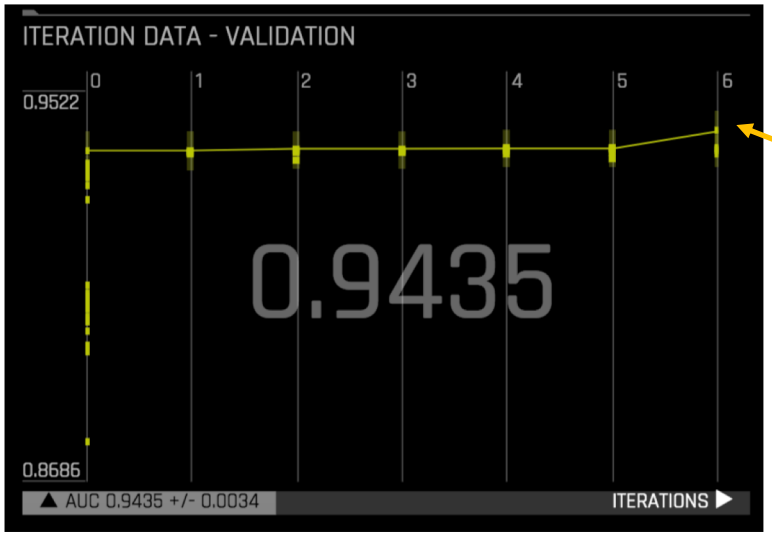
- New features are created from the data
- Models are trained on a combination of the original data and new features
- Models are evaluated on the internal holdout data



Feature Engineering

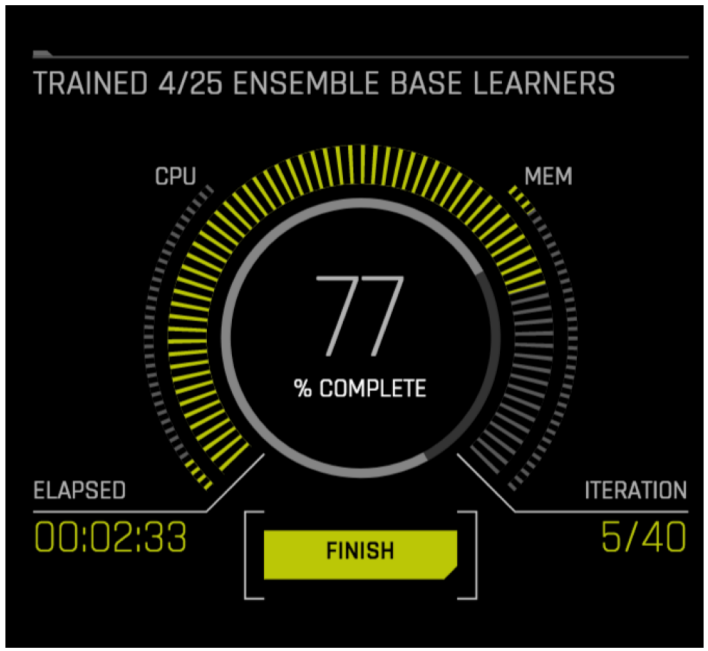
Final Model Training

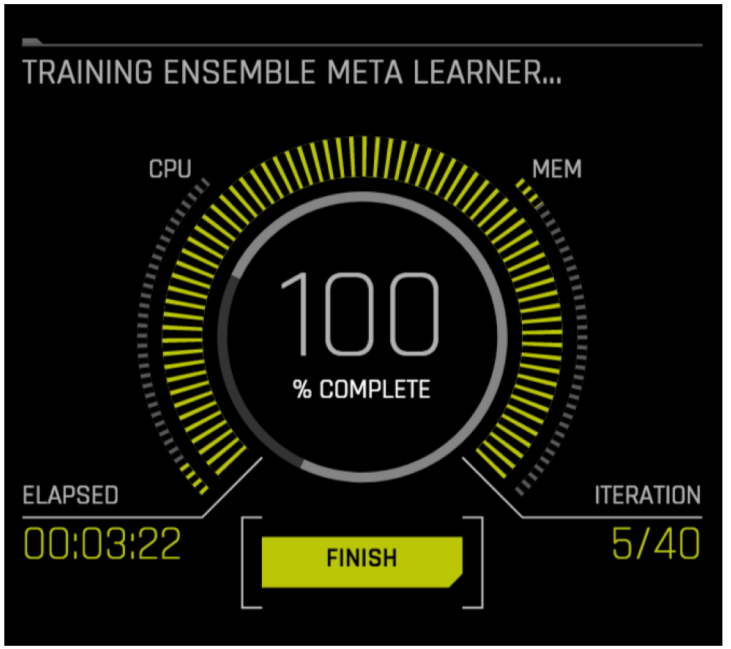
- Final Model Trained – can be an ensemble of multiple models
  - Ensemble models trained on the same data but use different parameters



Final Improvement Due to Ensembling

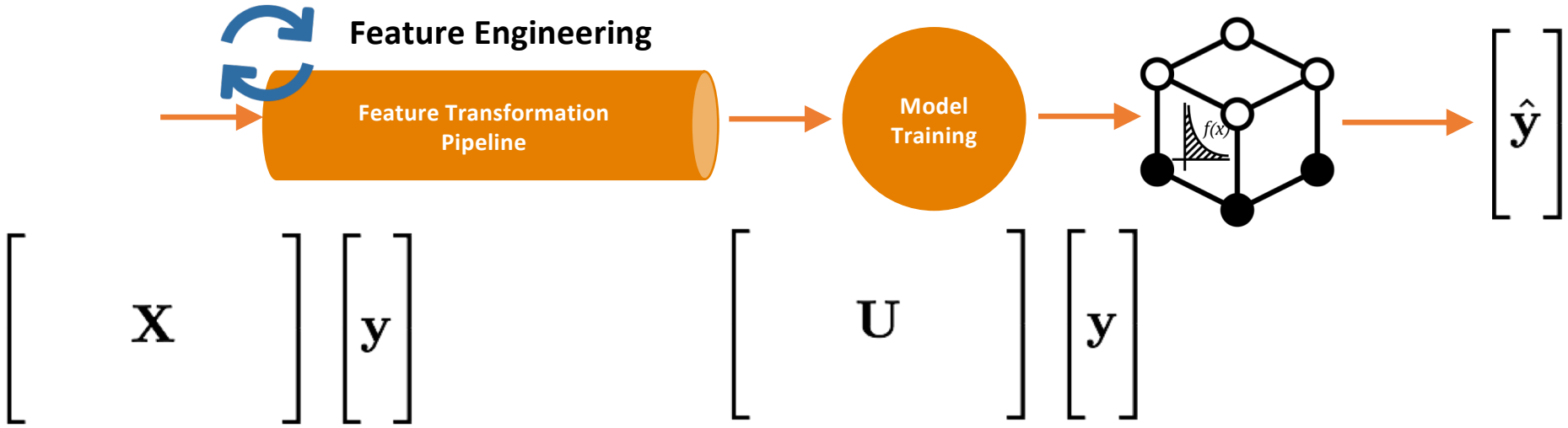
Final Model Training



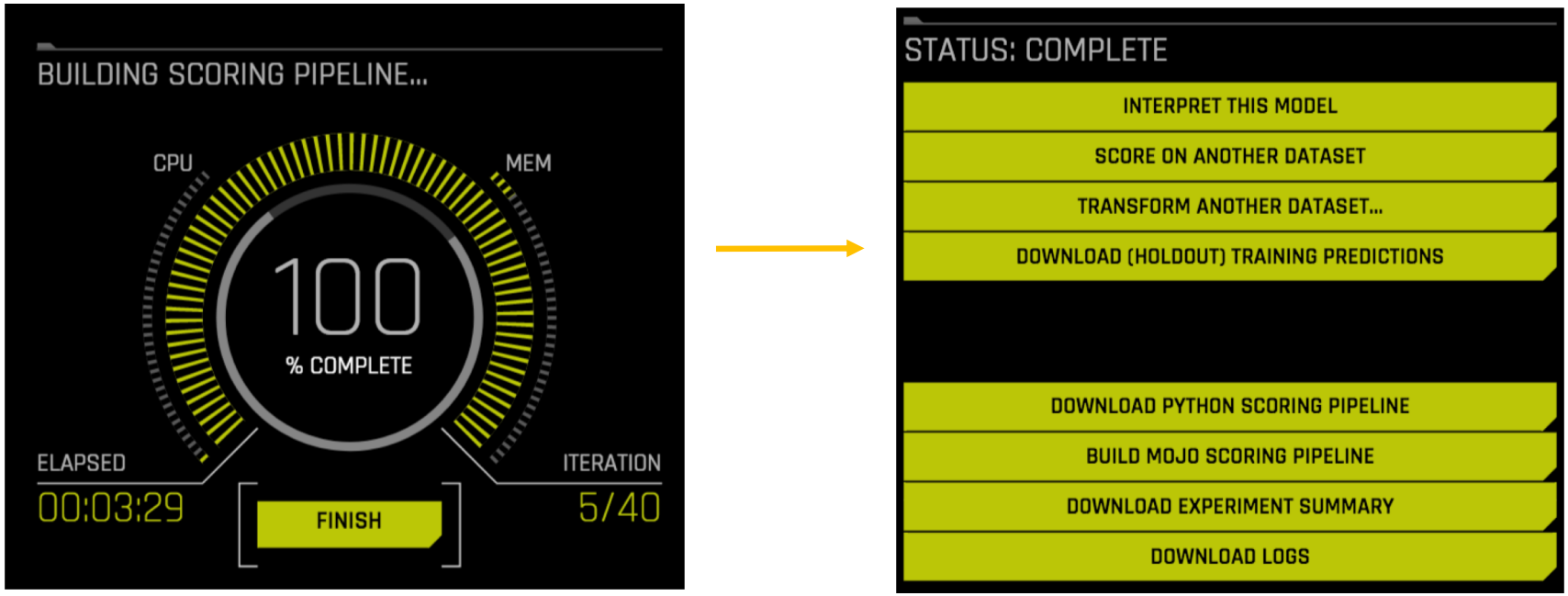


Scoring Pipeline

- Scoring pipeline created
  - Used independently of Driverless AI to score on a new record or frame of data

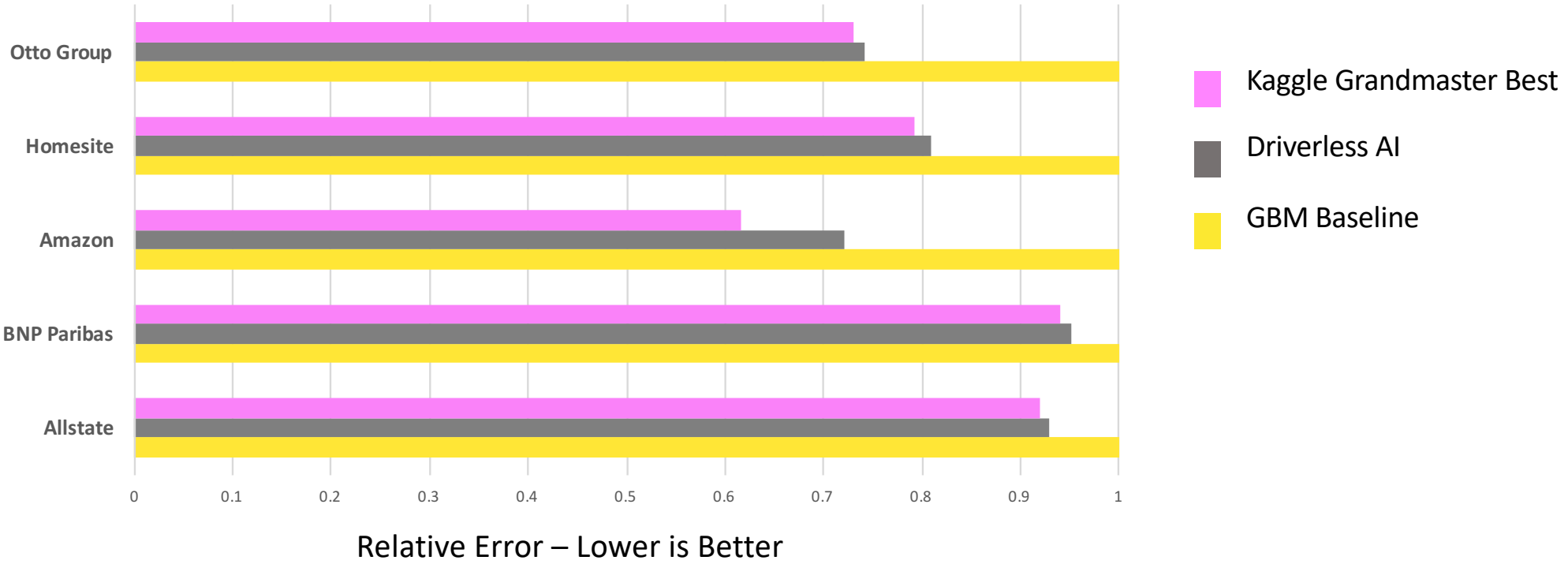


Scoring Pipeline



Feature Engineering

How Does Feature Engineering Effect Accuracy?



Target Mean Encoding

What?

Replace categorical variables with the mean of the response

Why?

Categorical variables increase the number of features (dummy encoding) and can cause us to overfit

Target Mean Encoding

Pay 1	Default Payment
Up To Date	0
Up To Date	0
Up To Date	0
Missed 1 Mo	1
Missed 1 Mo	0
Missed 1 Mo	0
Missed 5 Mo	1

Target Mean Encoding

Pay 1	Default Payment	Mean Target Encoding
Up To Date	0	0
Up To Date	0	0
Up To Date	0	0
Missed 1 Mo	1	0.33
Missed 1 Mo	0	0.33
Missed 1 Mo	0	0.33
Missed 5 Mo	1	1

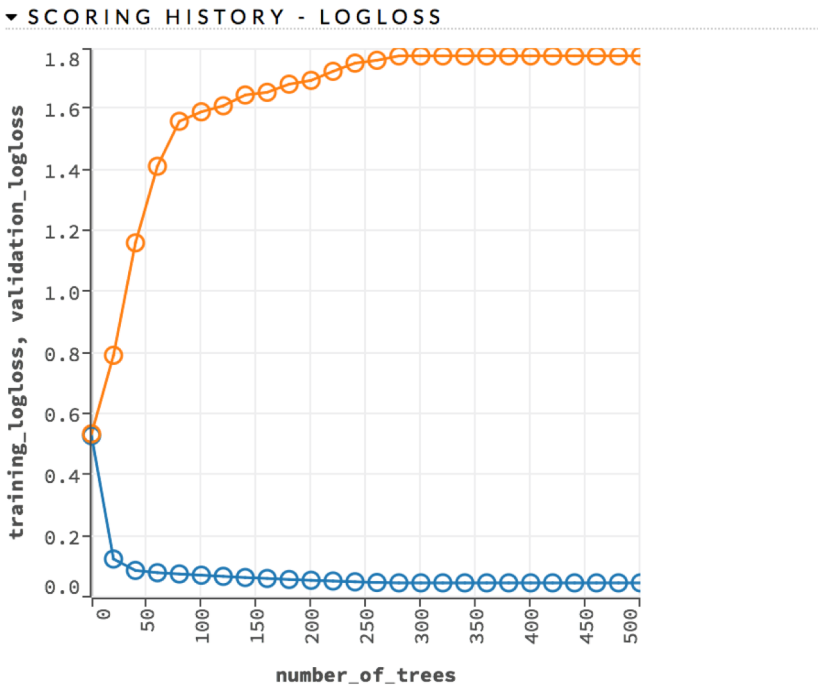
Target Mean Encoding

- Mean Target Encoding is based on the response column of the rows
- The lower the number of rows in the group, the more it reveals the response column value

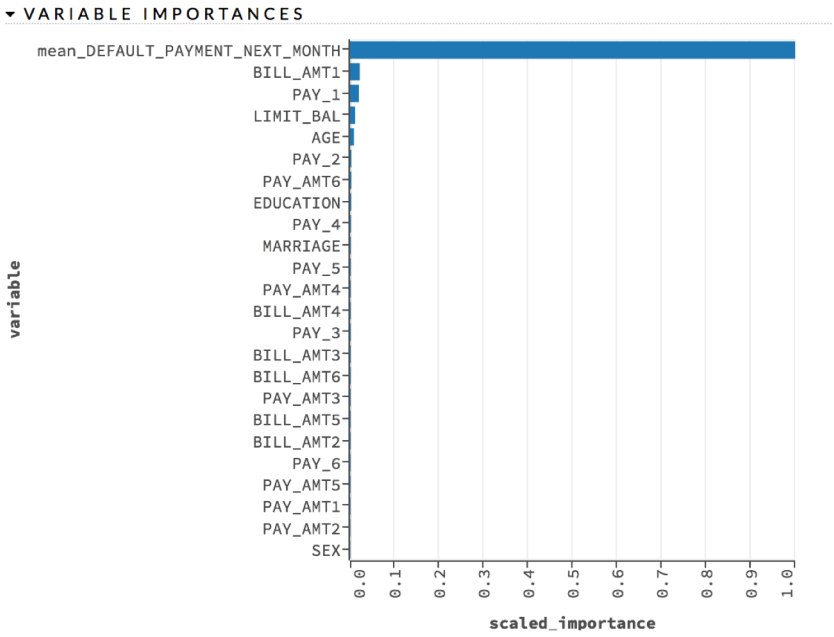
Pay 1	Default Payment	Mean Target Encoding
Missed 5 Mo	1	1

Worst Case Scenario: Response Column = Mean Target Encoding

Effects of Data Leakage



Scoring History: Training vs Testing



Data Leakage Feature is the only important feature

Cross Validation Target Encoding

Fold	Pay 1	Default Payment
1	Up To Date	0
2	Up To Date	0
3	Up To Date	0
2	Missed 1 Mo	1
1	Missed 1 Mo	0
3	Missed 1 Mo	0
1	Missed 5 Mo	1

Cross Validation Target Encoding

Fold	Pay 1	Default Payment
2	Up To Date	0
3	Up To Date	0
2	Missed 1 Mo	1
3	Missed 1 Mo	0

Fold	Pay 1	Default Payment
1	Up To Date	0
1	Missed 1 Mo	0
1	Missed 5 Mo	1

Cross Validation Target Encoding

Fold	Pay 1	Default Payment	Mean Target Encoding
2	Up To Date	0	0
3	Up To Date	0	0
2	Missed 1 Mo	1	0.5
3	Missed 1 Mo	0	0.5

Fold	Pay 1	Default Payment
1	Up To Date	0
1	Missed 1 Mo	0
1	Missed 5 Mo	1

Cross Validation Target Encoding

Fold	Pay 1	Default Payment	Mean Target Encoding
2	Up To Date	0	0
3	Up To Date	0	0
2	Missed 1 Mo	1	0.5
3	Missed 1 Mo	0	0.5

Fold	Pay 1	Default Payment	CV Target Encoding
1	Up To Date	0	0
1	Missed 1 Mo	0	
1	Missed 5 Mo	1	

Cross Validation Target Encoding

Fold	Pay 1	Default Payment	Mean Target Encoding
2	Up To Date	0	0
3	Up To Date	0	0
2	Missed 1 Mo	1	0.5
3	Missed 1 Mo	0	0.5

Fold	Pay 1	Default Payment	CV Target Encoding
1	Up To Date	0	0
1	Missed 1 Mo	0	0.5
1	Missed 5 Mo	1	

Cross Validation Target Encoding

Fold	Pay 1	Default Payment	Mean Target Encoding
2	Up To Date	0	0
3	Up To Date	0	0
2	Missed 1 Mo	1	0.5
3	Missed 1 Mo	0	0.5

Fold	Pay 1	Default Payment	CV Target Encoding
1	Up To Date	0	0
1	Missed 1 Mo	0	0.5
1	Missed 5 Mo	1	NA

Weight Of Evidence Encoding

What?

In binary classification, replace categorical variables with

$$WOE_{ja} = \ln \frac{P(X_j = a|Y = 1)}{P(X_j = a|Y = 0)}$$

Why?

Leverage rich history in information theory and Bayesian statistics to manage overfitting of high cardinality variables

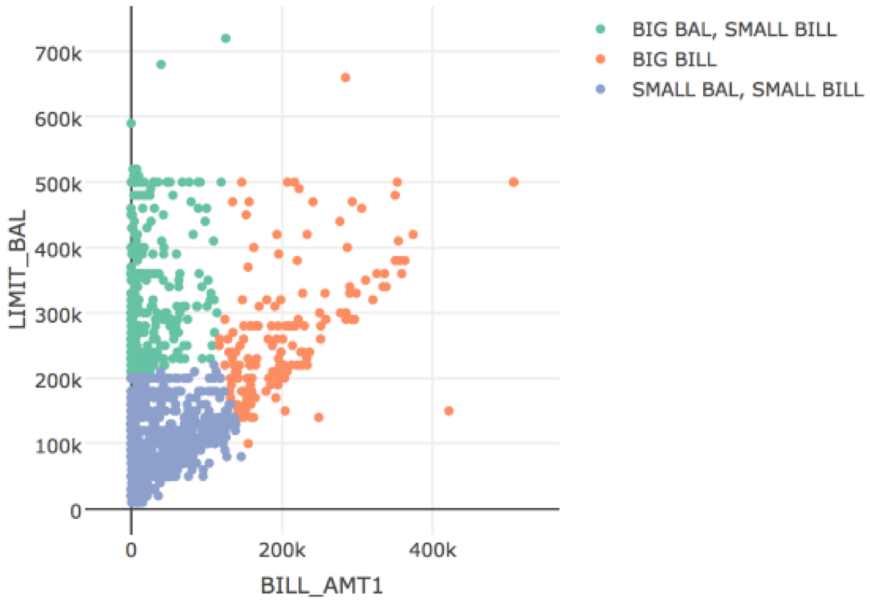
Weight Of Evidence Encoding

Pay 1	Default Payment	% 0s	% 1s	(% 1s) / (% 0s)	WOE
Up To Date	0	60 %	0 %	0	- Inf
Up To Date	0	60 %	0 %	0	- Inf
Up To Date	0	60 %	0 %	0	- Inf
Missed 1 Mo	1	40 %	50 %	1.25	0.223
Missed 1 Mo	0	40 %	50 %	1.25	0.223
Missed 1 Mo	0	40 %	50 %	1.25	0.223
Missed 5 Mo	1	0 %	50 %	Inf	Inf

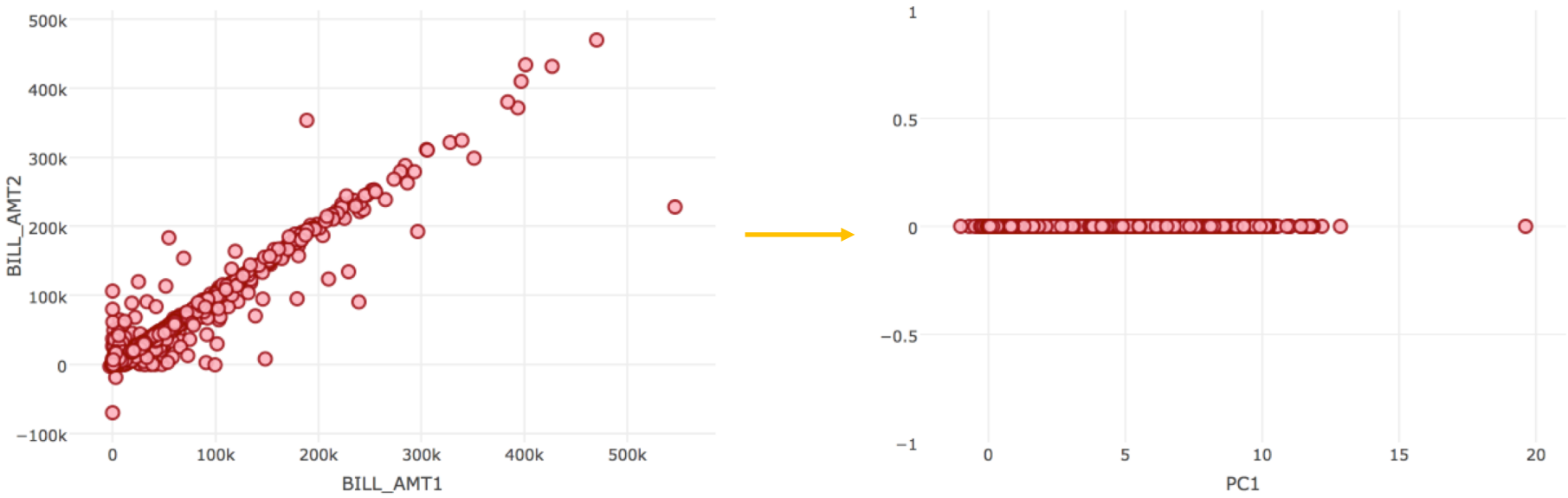
Clustering

Cluster Transformations

- Distance to a specific cluster
- Cross Validation Target Encoding by Cluster ID



Truncated SVD



Model Interpretability

Model Interpretability Techniques

Final Model (Global)

Goal: Get a general understanding of how the model works

Example: What were the most important features to predict default?

Techniques:

- Feature Importance
- Shapley

Surrogate Models (Local)

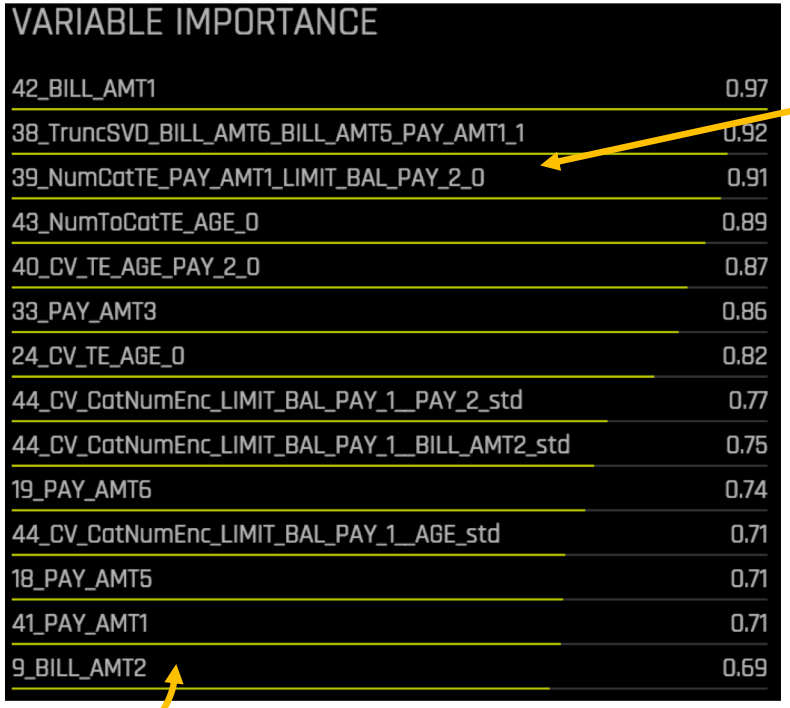
Goal: Understand why a particular record received that prediction

Example: Why did the model predict such a a high probability of default for this particular customer?

Techniques:

- Local Interpretable Model-Agnostic Explanations with Clustering (K-LIME)
- Partial Dependency Plots
- Leave One Covariate Out (LOCO)

The Challenge



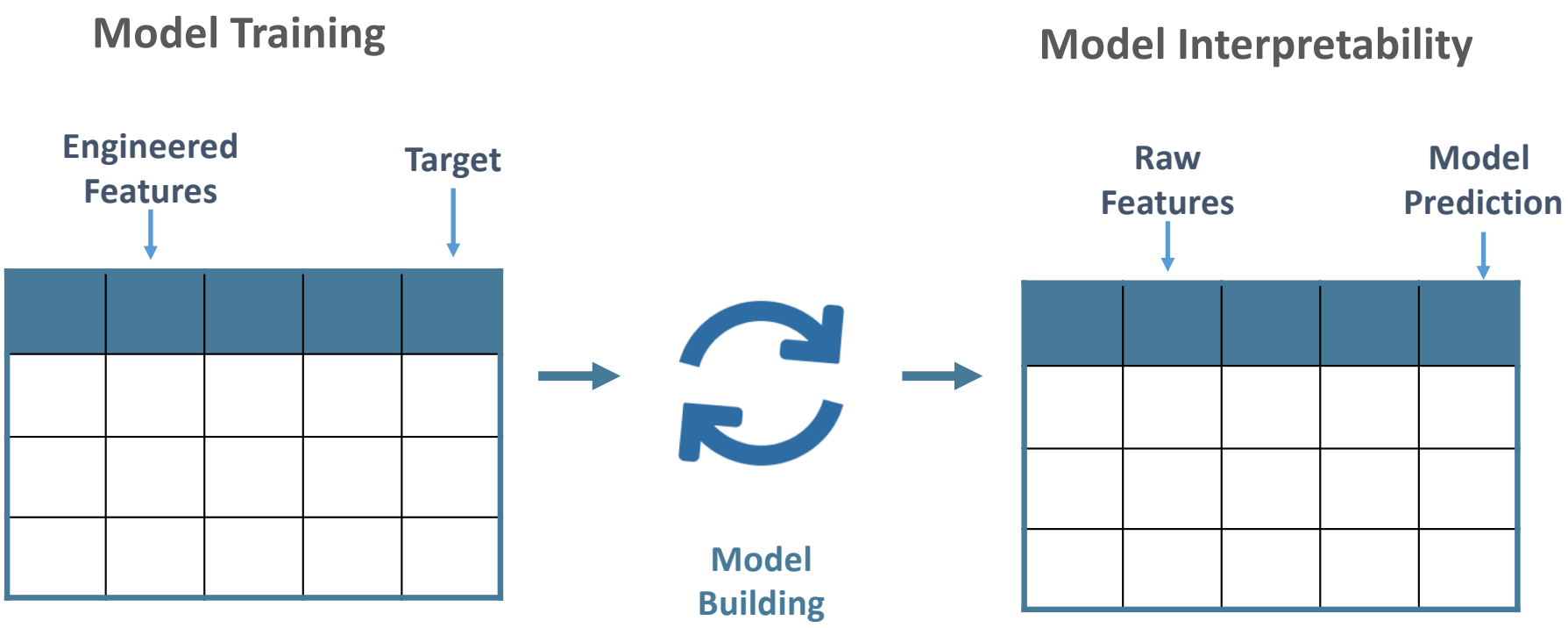
Generated Features

Original Features

How can we understand the meaning behind features like:

- ***TruncSVD_BILL_AMT6_BILL_AMT5_PAY_AMT1***
- ***CV_CatNumEnc_LIMIT_BAL_PAY1__PAY_2_std***

The Solution



The Solution

DEFAULT	TE_PAY_1	LIMIT_BAL	TE_EDUCATION
YES	0.53	$20,000	0.32
NO	0.19	$90,000	0.32
NO	0.15	$50,000	0.19

1. Train a complex machine learning model on generated features

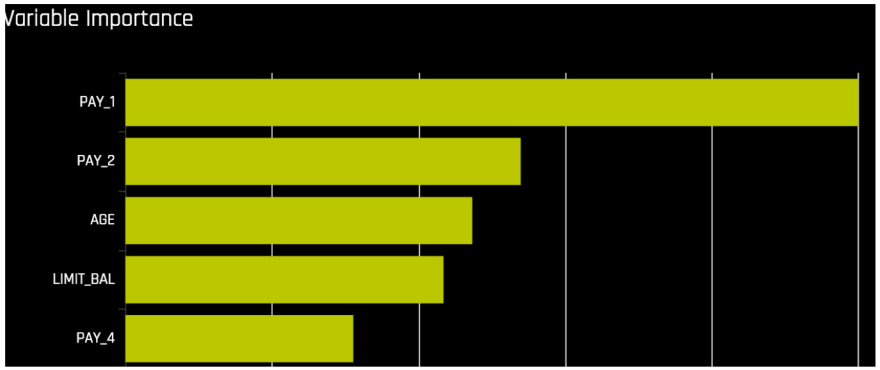
Prob DEFAULT	PAY_1	LIMIT_BAL	EDUCATION
81%	Missed 2 Mo	$20,000	university
32%	Up to Date	$90,000	university
21%	Up to Date	$50,000	graduate

2. Train a complex machine learning model on raw features and the predicted target values of our original model

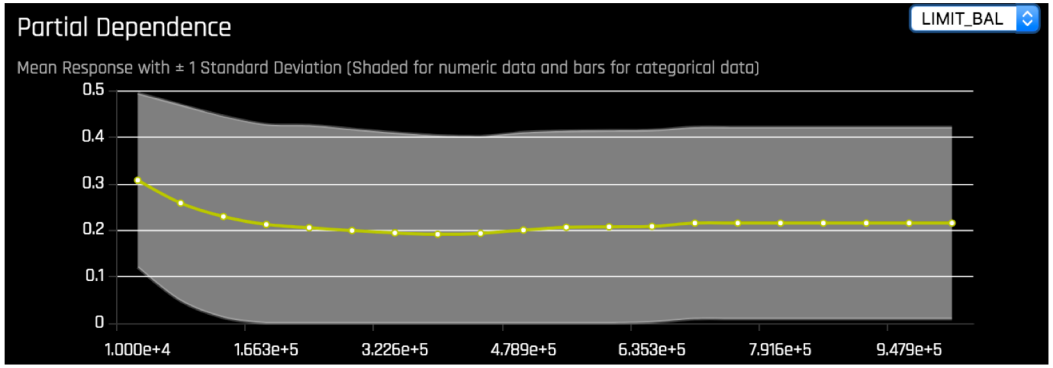
Understanding the Model Globally

2. Train a complex machine learning model on raw features and the predicted target values of our original model

Variable Importance



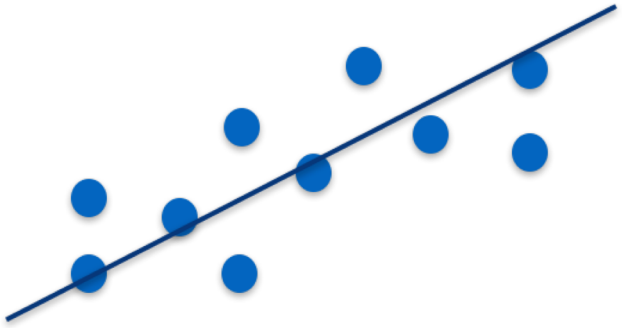
Partial Dependency Plots



LIMIT BAL

Challenge

Our Model



Benchmarks

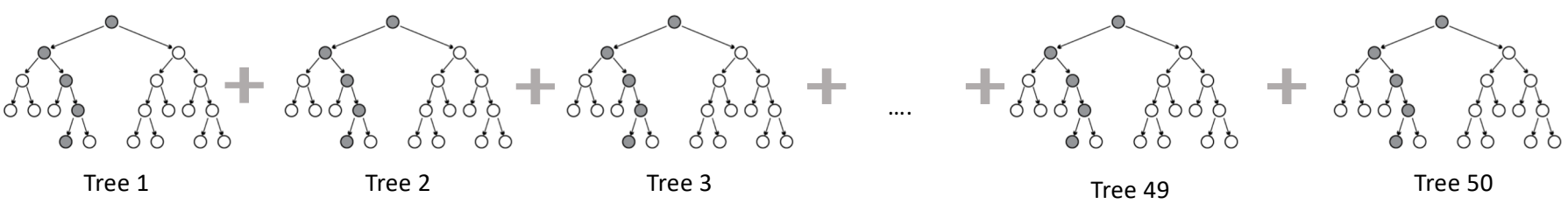
AUC	0.71
Max Accuracy	84%

Default Interpretability

Can we understand the model overall?	Yes
Do we know exactly why a prediction was made for each record?	Yes

Challenge

Our Model



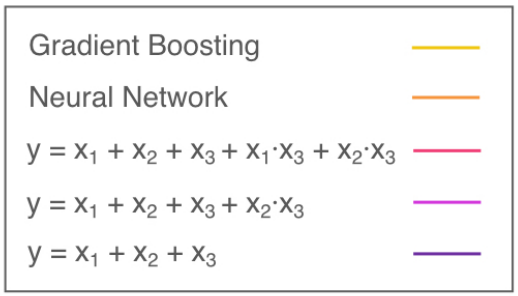
Benchmarks

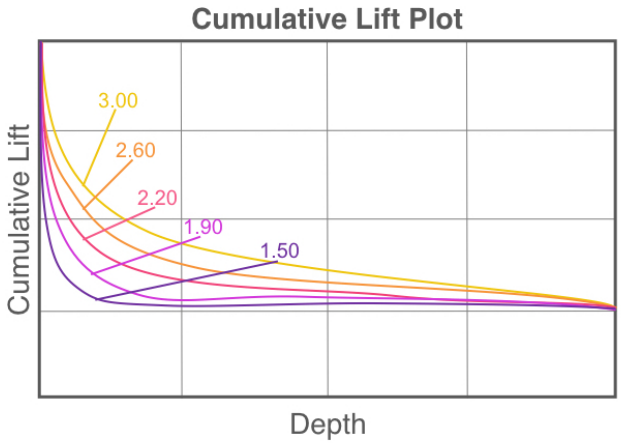
AUC	0.77
Mean Per Class Error	82%

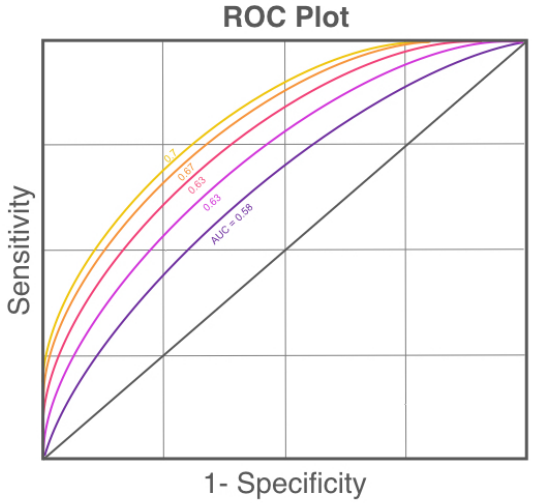
Default Interpretability

Can we understand the model overall?	No
Do we know exactly why a prediction was made for each record?	No

Challenge

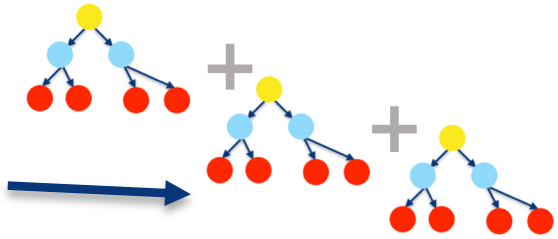






Surrogate Models

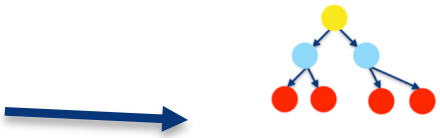
DEFAULT	TE_PAY_1	LIMIT_BAL	TE_EDUCATION
YES	0.53	$20,000	0.32
NO	0.19	$90,000	0.32
NO	0.15	$50,000	0.19



Gradient Boosting Machine

1. Train a complex machine learning model on generated features

Prob DEFAULT	PAY_1	LIMIT_BAL	EDUCATION
81%	Missed 2 Mo	$20,000	university
32%	Up to Date	$90,000	university
21%	Up to Date	$50,000	graduate



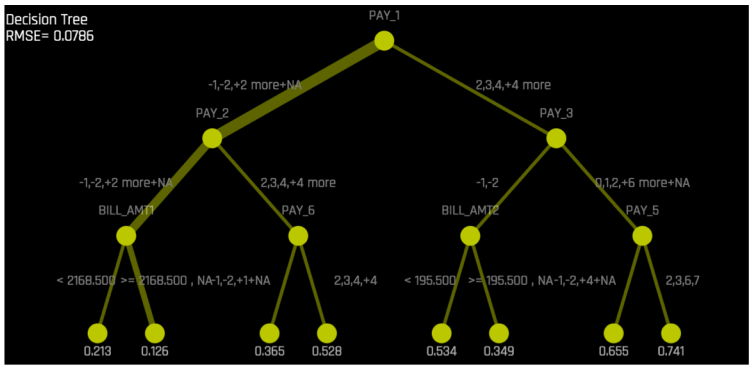
Interpretable Decision Tree

Interpretable Linear Model

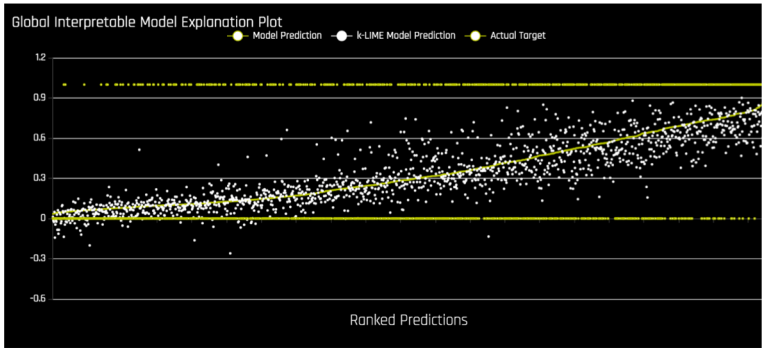
2. Train an interpretable model on raw features and the predicted target values of our original model

Surrogate Models

Surrogate Decision Tree

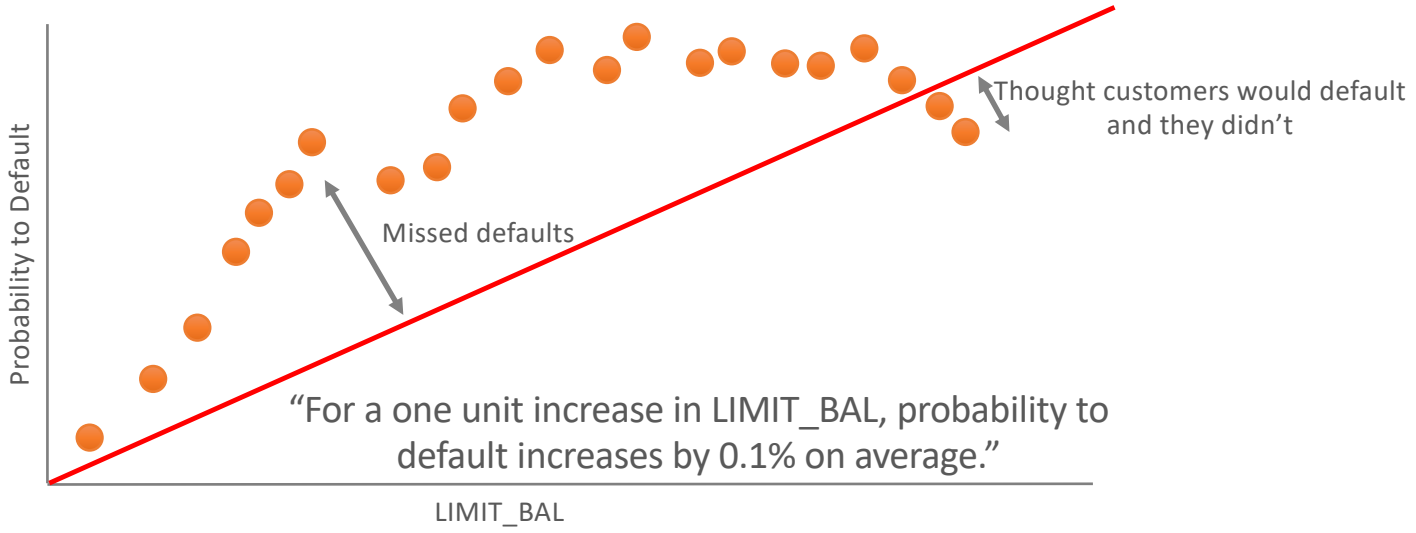


K-Lime Linear Models



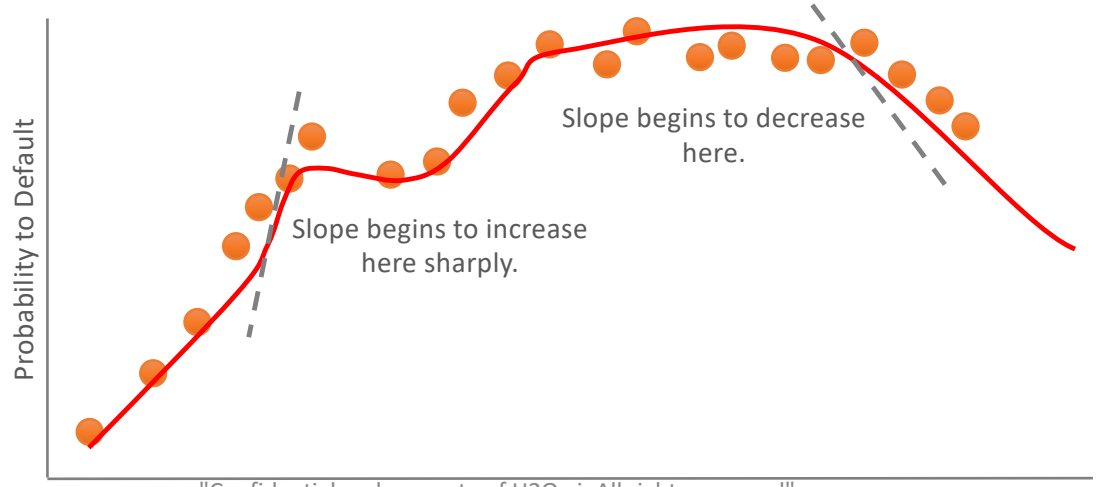
Linear Models

Exact explanations for approximate models.



Machine Learning

Approximate explanations for exact models.



Reason Codes

Local Reason Codes

k-LIME Local Attributions	Variable	with value	is associated with DEFAULT_PAYMENT_NEXT_MONTH	
Top Positive Local Attributions				
	PAY_1	2	increase of	0.34
	PAY_5	2	increase of	0.06
	PAY_3	2	increase of	0.06
Skipped 8 additional attributions, click to view all ...				
Top Negative Local Attributions				
	PAY_AMT3	3000	decrease of	0.01
	BILL_AMT5	24930	decrease of	0.01
	BILL_AMT1	21024	decrease of	0.01

Why will someone Default?

The fact that they haven't paid in 2 months **increases their likelihood by 34%**

Their Pay Amount is $3,000 **decreases their likelihood by 1%**

Amazon Reviews Experiment

The Data

- Amazon Fine Food Reviews
  - Contains: product and user information, ratings, and plain text review
  - Date Range: August 2011 – July 2012
  - Reference: https://www.kaggle.com/snap/amazon-fine-food-reviews
  - File System: /data/Kaggle/AmazonFineFoodReviews/AmazonFineFoodReviews-train-26k.csv
- Our Goal: Predict whether someone has left a positive review.

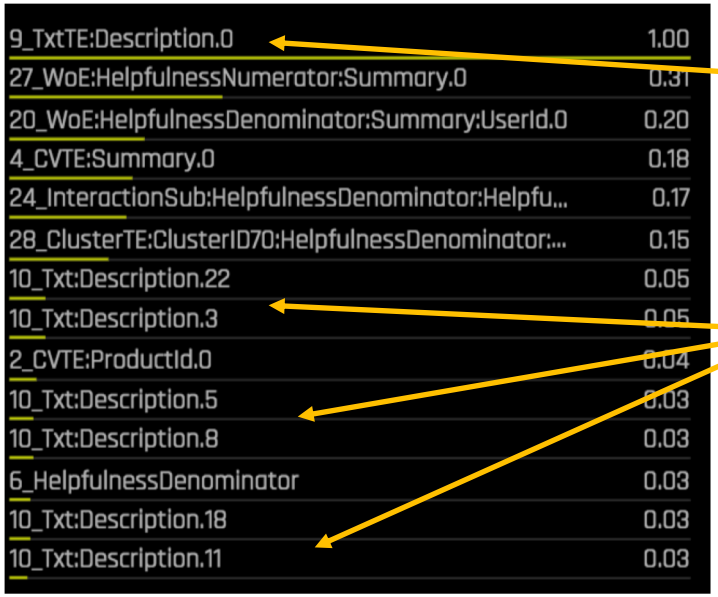
Positive Review 

The Data

Column	Example
User ID	A1UQRSCLF8GW1T
Product ID	B006K2ZZ7K
ID	142228
Summary	Great taffy
Score	5
Helpfulness Denominator	1
Profile Name	Michael Scott
Helpfulness Numerator	1
Time	1350777600
Description	*"Great taffy at a great price. There was a wide assortment of yummy taffy. Delivery was very quick. If your a taffy lover, this is a deal."*
Positive Review	**1**

Text Feature Engineering

Text Features



TxtTE – Train a linear model on the text components from TF-IDF

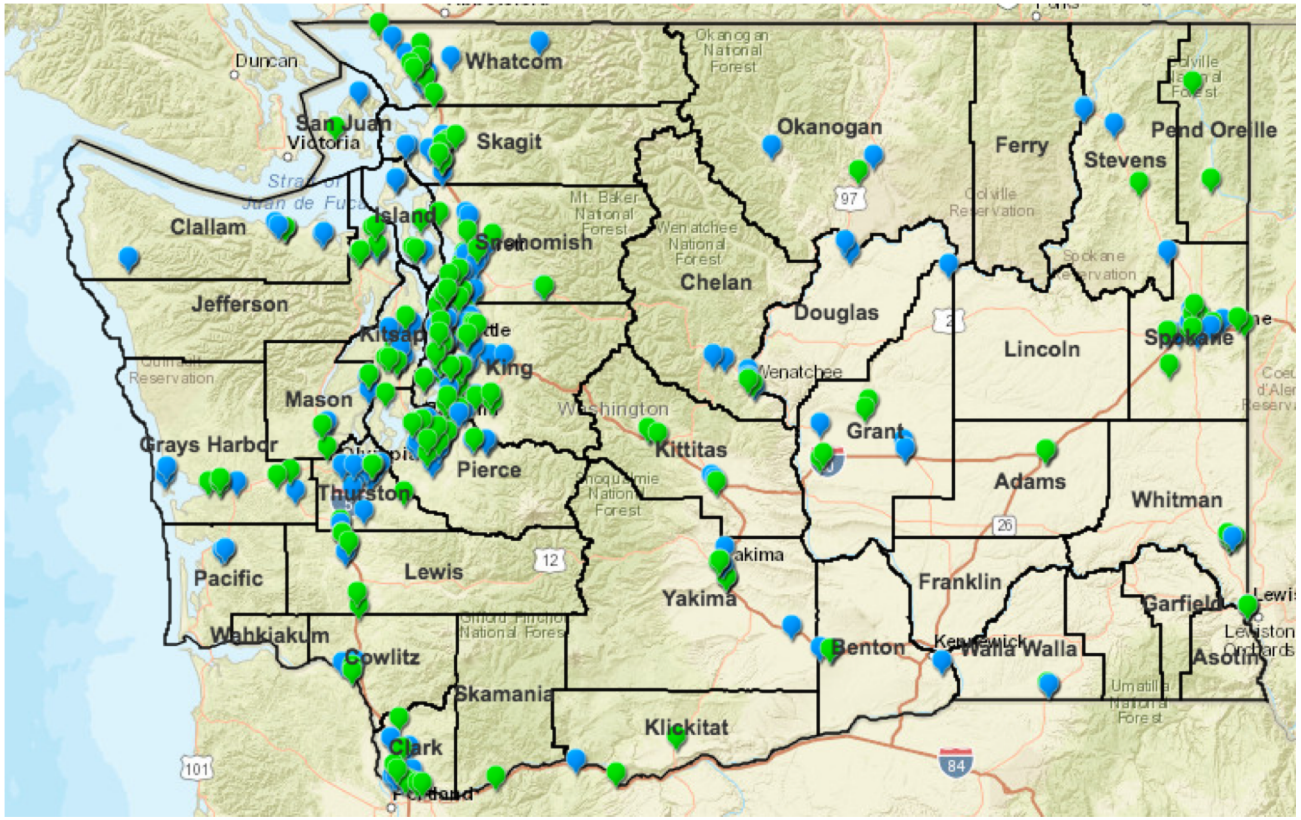
Txt – Components from a TF-IDF Matrix

Performance Comparison

Model	AUC
H2O-3 AutoML	0.64
H2O-3 AutoML with Word Embeddings from Word2Vec	0.88
Driverless AI	0.91

Washington State
Daily Cannabis Sales Experiment

Washington State Cannabis Retail Locations



The Data

- Washington State Cannabis Sales
  - Contains: daily sales metrics for different organizations in Washington state
  - Date Range: 2015-11-08 to 2017-03-04
  - Reference: https://data.lcb.wa.gov/dataset/Dashboard-Usable-Sales-w-Weight-Daily/9wz2-qma2
  - s3://h2o-public-test-data/smalldata/wa_cannabis/WA_Cannabis_Aggregated_Usable_Sales_Daily.csv

- Our Goal: Forecast sales for
  - The next week
  - The next month
  - The next quarter

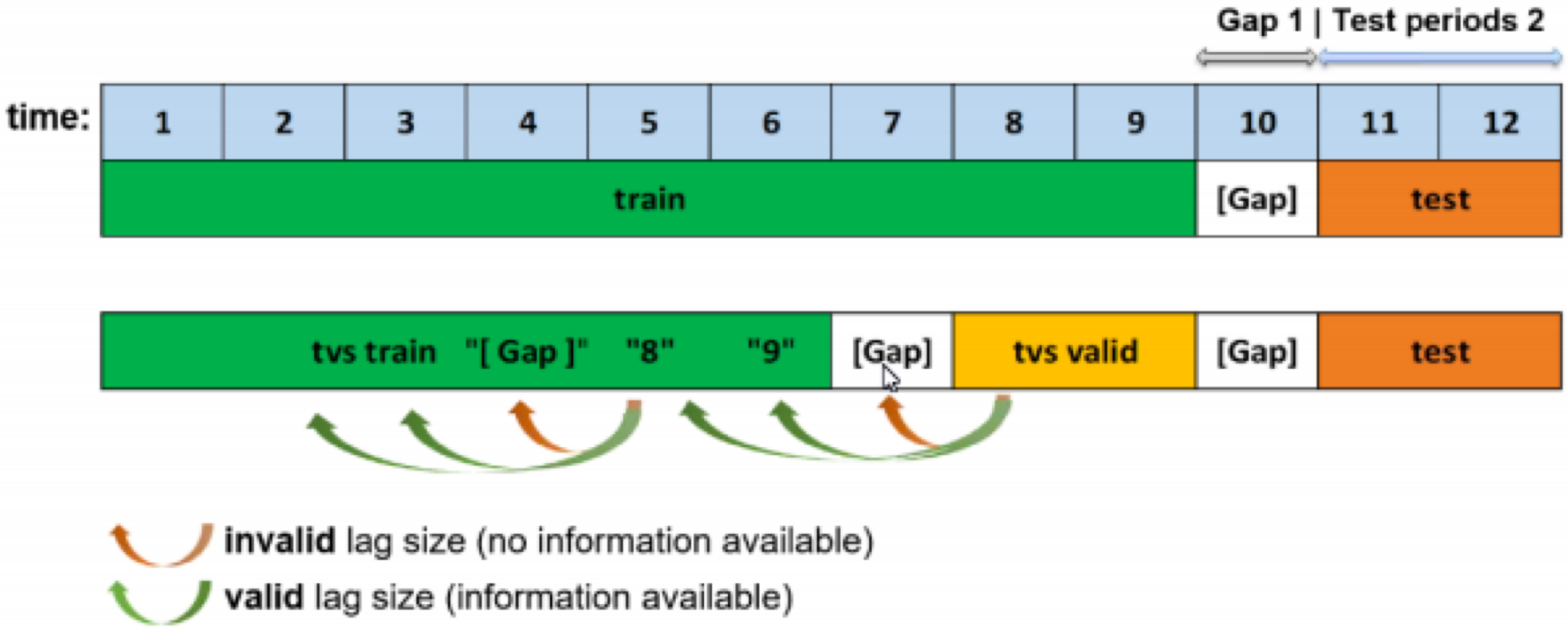
The Data

Column Name	Description
SalesDate	Sales date
Organization	Organization name
RowWeights	1 if ordinary, 365 if day of interest (optional for upweighting DOIs)
DayOfInterest	Day of interest name (DOI name)
DayOfInterestCode	Day of interest code (DOI code)
DayOfWeek	Day of week name
DayOfWeekCode	Day of week code (Sunday = 1, Friday & Saturday = 3, Otherwise = 2)
IsOrdinaryDay	Is ordinary day, i.e. a non-day of interest?
LastWeekIQRLogSalesPrice*	Interquartile range for last week's sales on the natural log scale
LastWeekLog1pNumReturns*	Log1p(number of negative sales transactions from last week)
LastWeekLog1pDemandInThou*	Log1p(demand from last week in thousands)
LastWeekSkewLogSalesPrice*	Skewness statistic for last week's sales on the natural log scale
Log1pDemandInThou	Log1p(demand in thousands)

The Data

DayOfInterest	DayOfInterestCode	RowWeight
FourTwenty	2	365
FourTwentySeven	-2	365
PreJuly4th	1	365
ThanksgivingMinusOne	1	365
ThanksgivingPlusSix	-1	365
ChristmasMinusTwo	1	365
ChristmasMinusOne	2	365
Christmas	-3	365
ChristmasPlusOne	-1	365
NewYearsDay	2	365
NewYearsDayPlusSix	-2	365
N/A	0	1

Time Series Lag Features



Exponentially Weighted Moving Average

- Create new features by exponentially weighting a fixed number of lags (k) at common intervals, e.g. for daily data use weekly lags

$$EWMA_t = \begin{cases} y_{t-il}, & i = k \\ \alpha y_{t-l} + (1-\alpha) * EWMA_{t-l} & t > l \end{cases}$$

Questions?