H_2O

WORLD

2017

Mark Landry

Data Scientist, H2O.ai

GitHub: mlandry22, Email: mark@h2o.ai

Driver vs Driverless

An Analysis of Driverless AI Feature Creation

Overview

- Analyze two recent problems with an emphasis on feature generation
- Overview of the problem
- Discussion of my approach
- Show the features Driverless created
  - Feature creation
  - Feature representation
- Results

Who am I?

- Data scientist @ H2O since 2015, user since 2014
- 15 top 20 finishes in Kaggle, highest rank 33
- R | H2O | data.table | GBM
- The “driver”

Problem #1

How Many Attempts will a Student Make

- Online question/answer platform with computer science problems
- Predict the number of attempts a particular student will make on a particular problem
- Data
  - Student: level, ranking, highest ranking
  - Problem: type, 3-tier level, points awarded
  - Training: 124,000 attempt counts
  - Testing: 60,000 attempt counts

Regression as Classification

- Natural problem is numerical
- End user prefers buckets
- Volume
  - 1: 53%
  - 2 31%
  - 3   9%
  - 4   4%
  - 5   1%
  - 6   2%

attempts_range	No. of attempts
1	1-1
2	2-3
3	4-5
4	6-7
5	8-9
6	>=10

The Driver Approach

- Think of it like a recommender problem
  - Standard: Matrix factorization, collaborative filtering
  - GBM: use deep categorical encodings
- Frequent use of target encoding

N	F1	itx	
46471	0.3728824	Rank	
46328	0.4334152	User	
46471	0.4510425	Level	
38589	0.4616374	UserLevel	interaction
46471	0.4581436	RankLevel	interaction
44039	0.5149664	Problem	
40704	0.5283645	RankProblem	interaction

Table1: Standalone F1 rates using median-based calculation when 3 records are present per interaction level

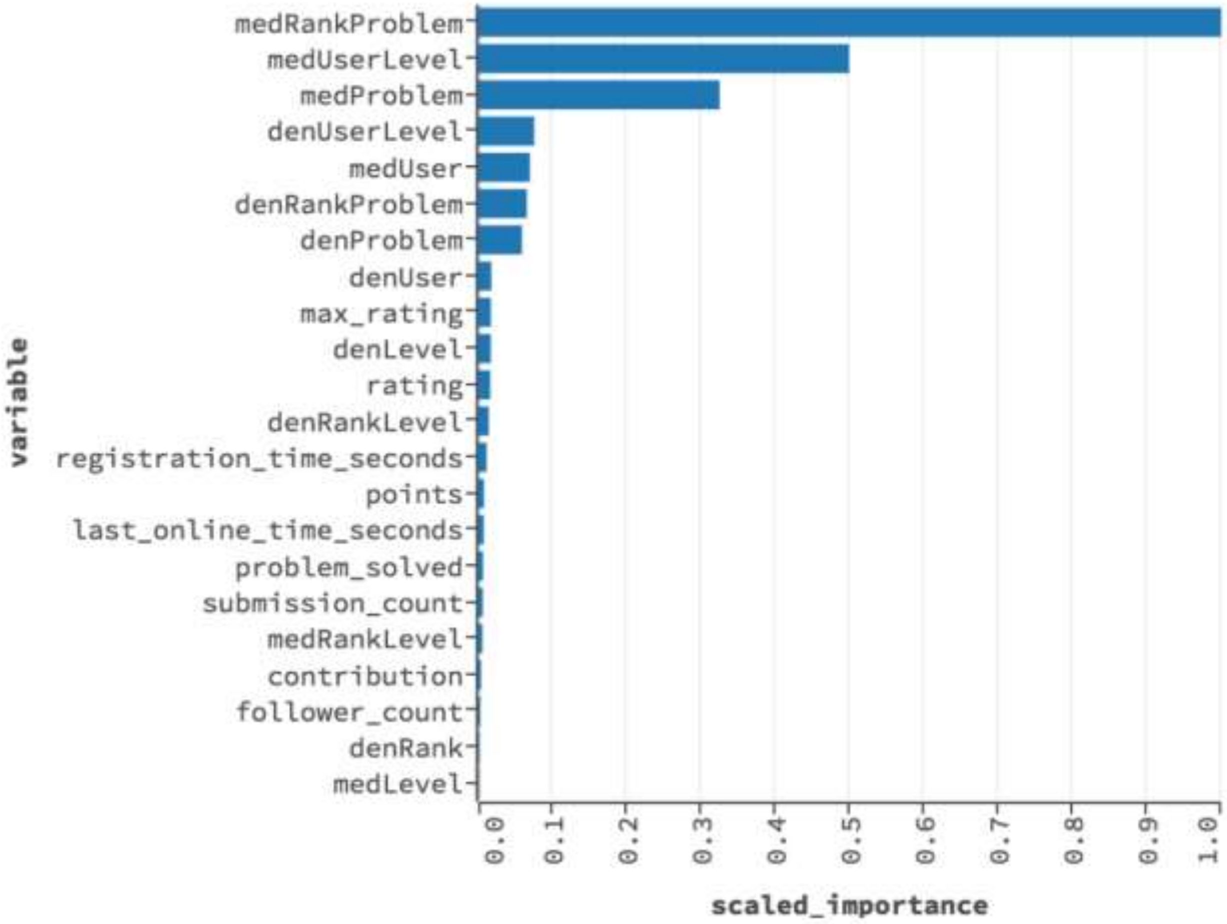
The Driver Approach

A messy chain of hierarchical target encoding and if/else statements

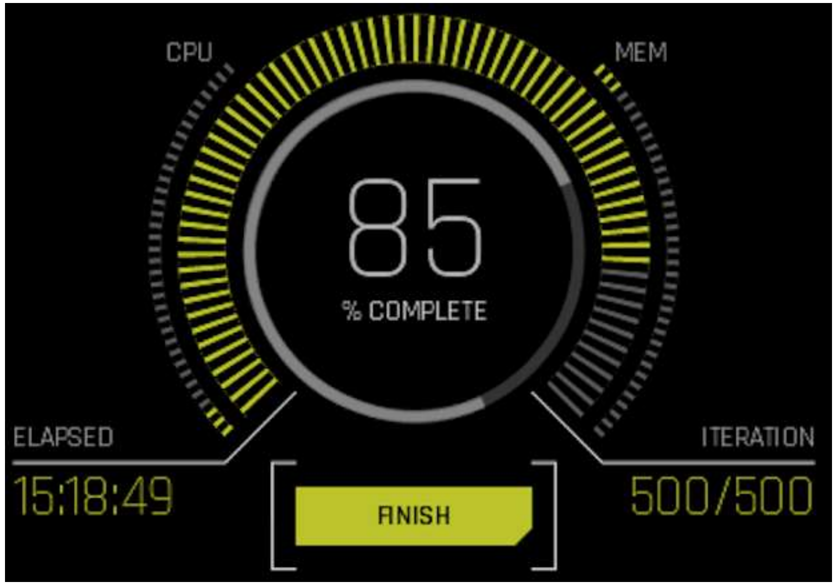
```
## RANK-LEVEL
all[,denRankLevel:=sum(isTrain),.(rank,level_type)]
all[,medLowRankLevel:=as.numeric(median(c(1,ifelse(isTrain,attempts_range,NA)),na.rm=TRUE)),.(rank,level_type)]
all[,medHiRankLevel:=as.numeric(median(c(6,ifelse(isTrain,attempts_range,NA)),na.rm=TRUE)),.(rank,level_type)]
all[,medRankLevel:=round(as.numeric(
  ifelse(!isTrain,medLowRankLevel*0.5+medHiRankLevel*0.5
         ,ifelse(denRankLevel==1,NA
                 ,ifelse(attempts_range<=medLowRankLevel,medHiRankLevel
                         ,ifelse(attempts_range>=medLowRankLevel,medHiRankLevel
                                 ,medLowRankLevel*0.5+medHiRankLevel*0.5))))))]

## RANK-PROBLEM
all[,denRankProblem:=sum(isTrain),.(rank,problem_id)]
all[,medLowRankProblem:=as.numeric(median(c(1,ifelse(isTrain,attempts_range,NA)),na.rm=TRUE)),.(rank,problem_id)]
all[,medHiRankProblem:=as.numeric(median(c(6,ifelse(isTrain,attempts_range,NA)),na.rm=TRUE)),.(rank,problem_id)]
all[,medRankProblem:=round(as.numeric(
  ifelse(!isTrain,medLowRankProblem*0.5+medHiRankProblem*0.5
         ,ifelse(denRankProblem==1,NA
                 ,ifelse(attempts_range<=medLowRankProblem,medHiRankProblem
                         ,ifelse(attempts_range>=medLowRankProblem,medHiRankProblem
                                 ,medLowRankProblem*0.5+medHiRankProblem*0.5))))))]
```

The Driver Approach

h2o.gbm feature importance – primarily using three target encodings

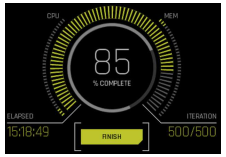


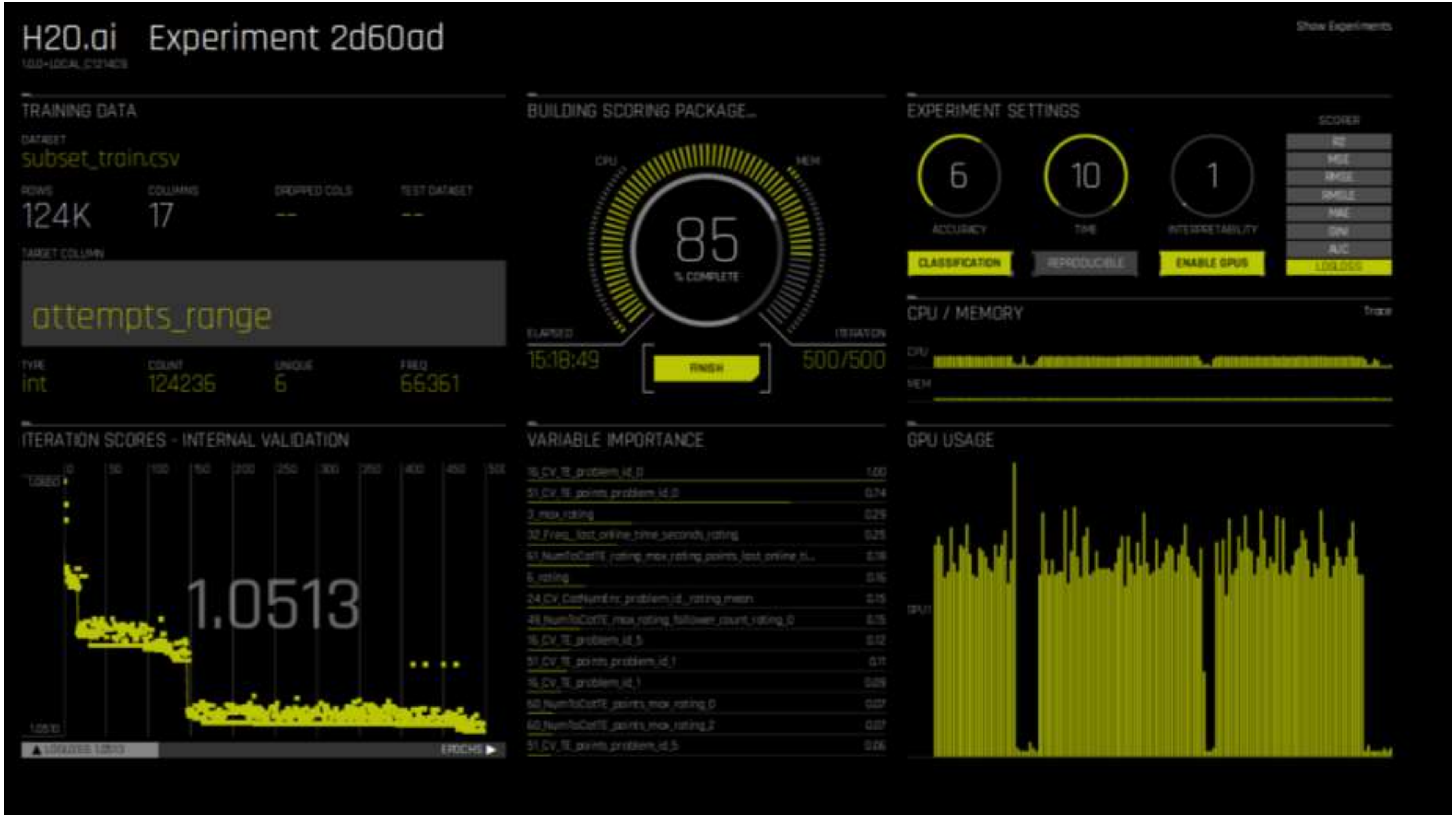
The Driverless Approach



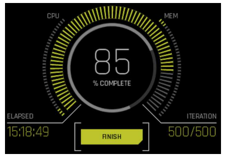
Staples® Easy Button™

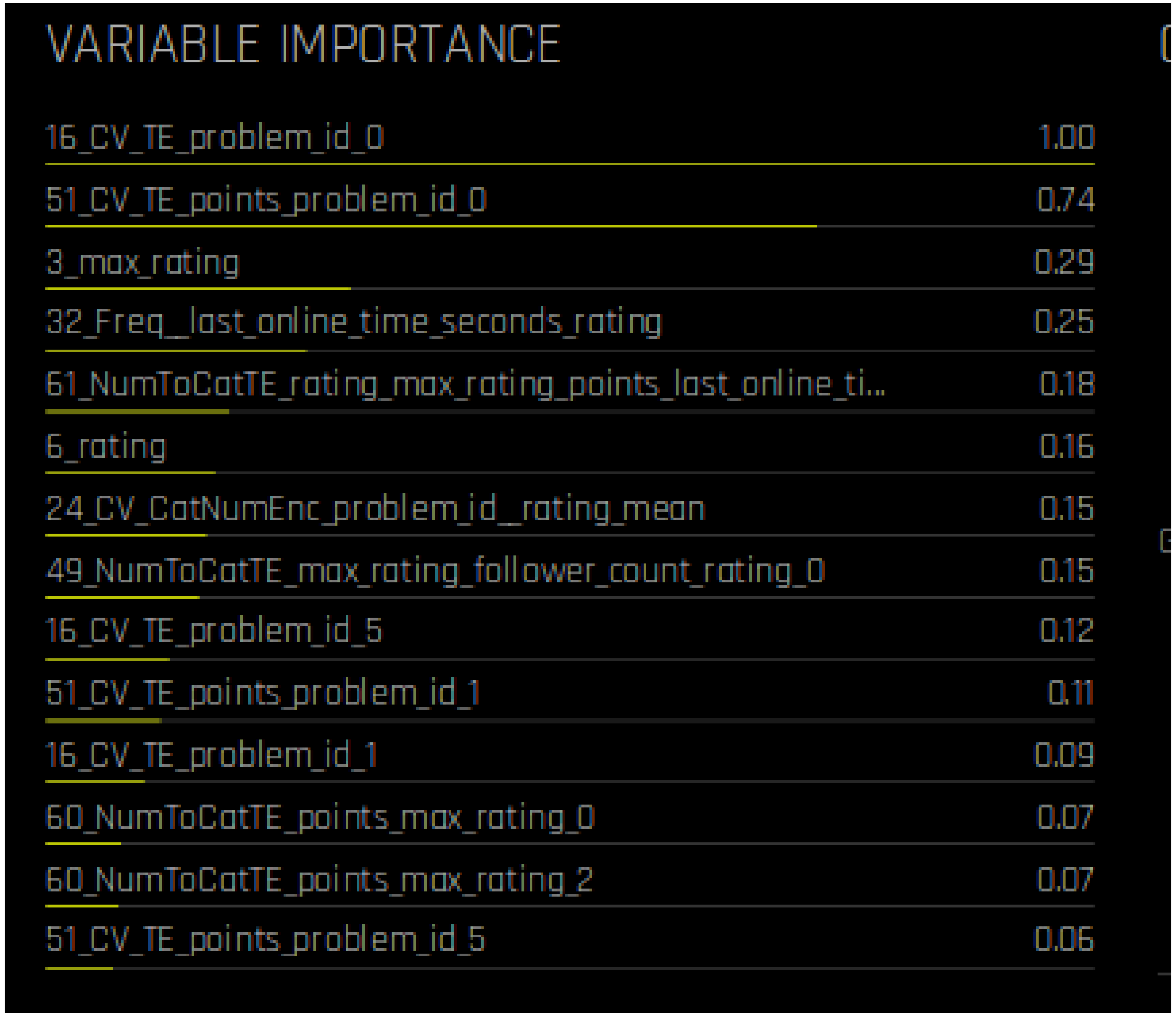
The Driverless Approach



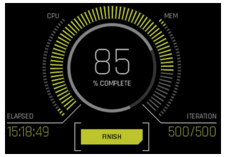


The Driverless Approach





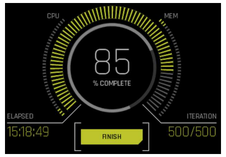
The Driverless Approach



Top 5 Features: divided into components

- {16} {CV TE} {problem_id} {0}
- {51} {CV TE} {points * problem_id} {0}
- {3}   {max_rating}
- {32} {freq} {last_online_time_seconds * rating}
- {61} {NumToCatTE} {rating * max_rating * points * last_online_time_seconds} {0}

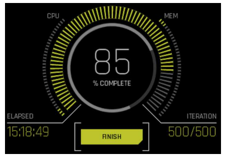
The Driverless Approach



Feature #1: same base target encoding I found to be the best

- {16} {CV TE} {problem_id} {0}
  - 16: indicator of base features – 16 is later used twice more in the top 15
  - CV TE: Cross-Fold target encoding
  - Problem_id: feature used as basis for target encoding
  - 0: the target; multinomial, so the two other uses are for class 1 & 5
- {51} {CV TE} {points * problem_id} {0}
- {3}   {max_rating}
- {32} {freq} {last_online_time_seconds * rating}
- {61} {NumToCatTE} {rating * max_rating * points * last_online_time_seconds} {0}

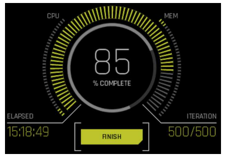
The Driverless Approach



Feature #2: deeper interaction

- {16} {CV TE} {problem_id} {0}
- {51} {CV TE} {points * problem_id} {0}
  - 51: base feature ID
  - CV TE: out of sample target encoding result
  - points * problem: interaction of two different features, both related to the problem; it is subdividing the problem further
  - 0: again, rate of class 0 as the target
- {3}   {max_rating}
- {32} {freq} {last_online_time_seconds * rating}
- {61} {NumToCatTE} {rating * max_rating * points * last_online_time_seconds} {0}

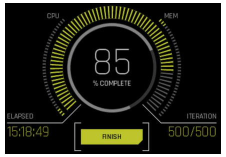
The Driverless Approach



Feature #3: no transformation

- {16} {CV TE} {problem_id} {0}
- {51} {CV TE} {points * problem_id} {0}
- {3}   {max_rating}
  - max rating
    - used as is – no alternate encoding; was first natural feature in my model as well
    - this is the first variable of the student dimension
    - a 4-digit number with close to a normal distribution
- {32} {freq} {last_online_time_seconds * rating}
- {61} {NumToCatTE} {rating * max_rating * points * last_online_time_seconds} {0}

The Driverless Approach

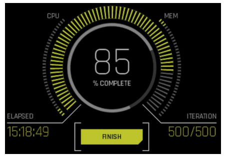


Feature #4: frequency encoding

- {16} {CV TE} {problem_id} {0}
- {51} {CV TE} {points * problem_id} {0}
- {3}   {max_rating}
- {32} {freq} {last_online_time_seconds * rating}
  - Counting the occurrences of two fields
  - Last online & rating are both numerics in the student dimension
- {61} {NumToCatTE} {rating * max_rating * points * last_online_time_seconds} {0}

The Driverless Approach

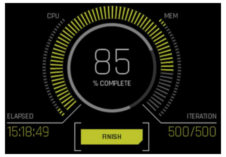
Feature 5: four-way interaction w/ target encoding



- {16} {CV TE} {problem_id} {0}
- {51} {CV TE} {points * problem_id} {0}
- {3} {max_rating}
- {32} {freq} {last_online_time_seconds * rating}
- {61} {NumToCatTE} {rating * max_rating * points * last_online_time_seconds} {0}
    - Target encoding of class 0
    - Finding the rate for each value of the result of a four way interaction

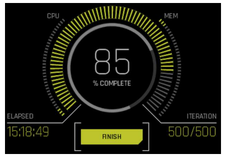
The Driverless Approach

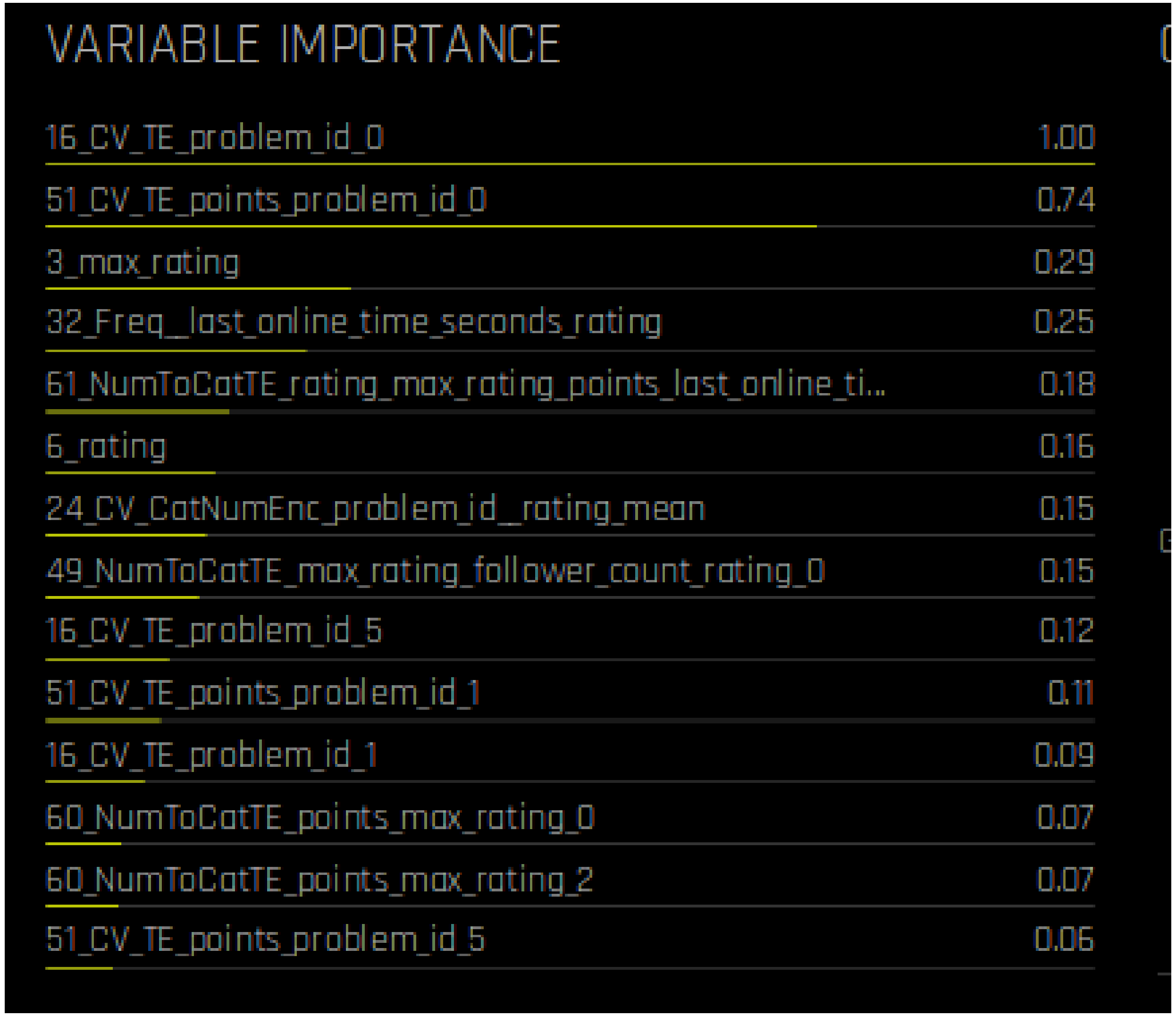
Top 5 Features: did I try?



- YES {16} {CV TE} {problem_id} {0}
- NO {51} {CV TE} {points * problem_id} {0}
- YES {3} {max_rating}
- NO {32} {freq} {last_online_time_seconds * rating}
- NO {61} {NumToCatTE} {rating * max_rating * points * last_online_time_seconds} {0}

The Driverless Approach





Problem #2

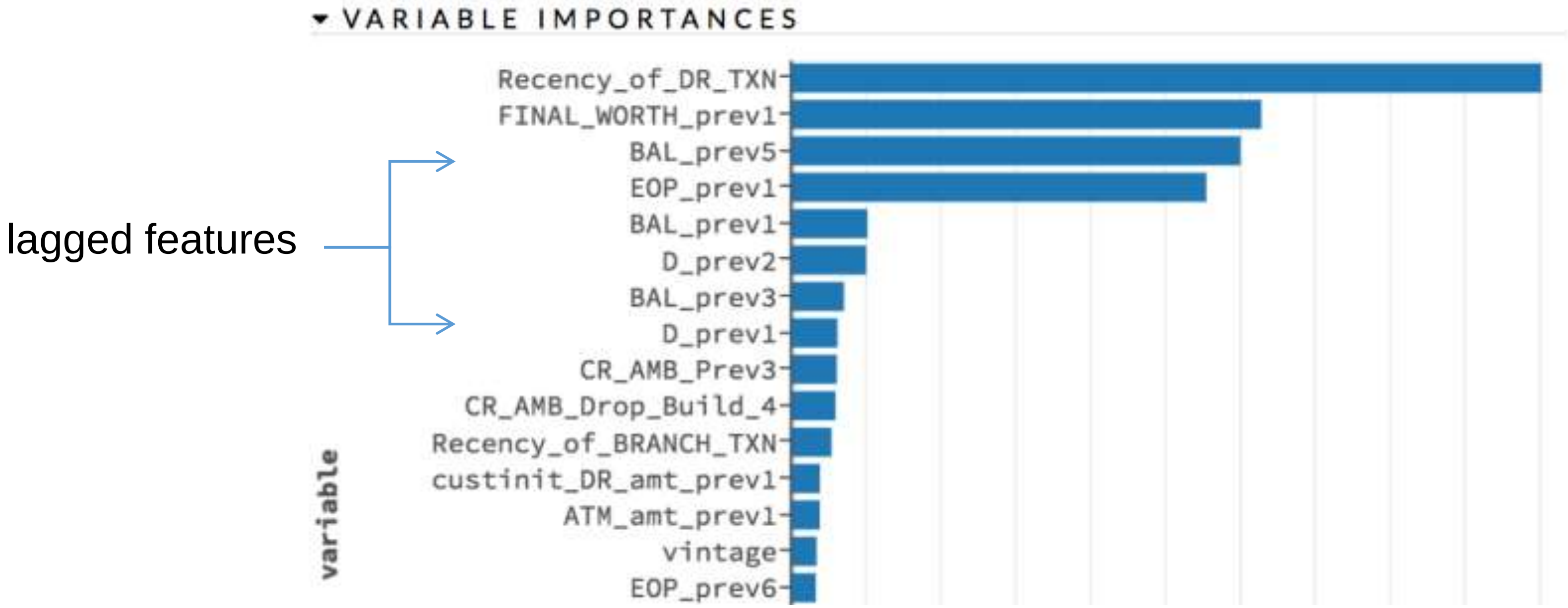
Bank Customer Churn

- Identify customers likely to churn balances in the next quarter by 50%
- Data
  - 300,00 training rows; 200,000 testing rows
  - 377 columns
  - Customer: age, gender, demographics
  - Reported assets, liabilities
  - Monthly balance history

The Driver Approach

Exploit before Explore

- 377 columns made [quick] manual investigation harder
- Rather than iterate: {analyze > model > analyze > … },
  I changed to {model > analyze > model > … }



The Driver Approach

After lagging, try differences and ratios

- Lagging features present the balance features at several time steps.
- But often, the interesting part is not the raw balance itself, but whether it is growing or shrinking
- Decision trees have a hard time “seeing” this so it is wise to engineer mathematical features: + - * /

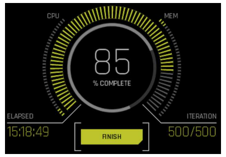
The Driver Approach

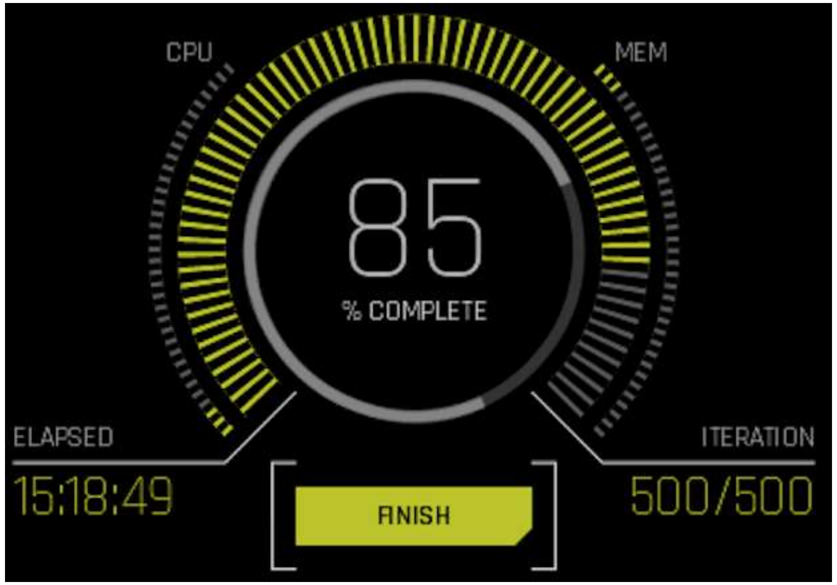
After lagging, try differences and ratios

- I used the leading monthly feature from the model and created new features representing month-over-month differences and a binary indicator
- One field, one specific length (1 month), two calculations

```
trainHex$diff_BAL_5_4<-trainHex$BAL_prev5-trainHex$BAL_prev4
trainHex$diff_BAL_4_3<-trainHex$BAL_prev4-trainHex$BAL_prev3
trainHex$diff_BAL_3_2<-trainHex$BAL_prev3-trainHex$BAL_prev4
trainHex$diff_BAL_2_1<-trainHex$BAL_prev2-trainHex$BAL_prev1

trainHex$rt_BAL_5_4<-trainHex$BAL_prev5/trainHex$BAL_prev4
trainHex$rt_BAL_4_3<-trainHex$BAL_prev4/trainHex$BAL_prev3
trainHex$rt_BAL_3_2<-trainHex$BAL_prev3/trainHex$BAL_prev4
trainHex$rt_BAL_2_1<-trainHex$BAL_prev2/trainHex$BAL_prev1
```

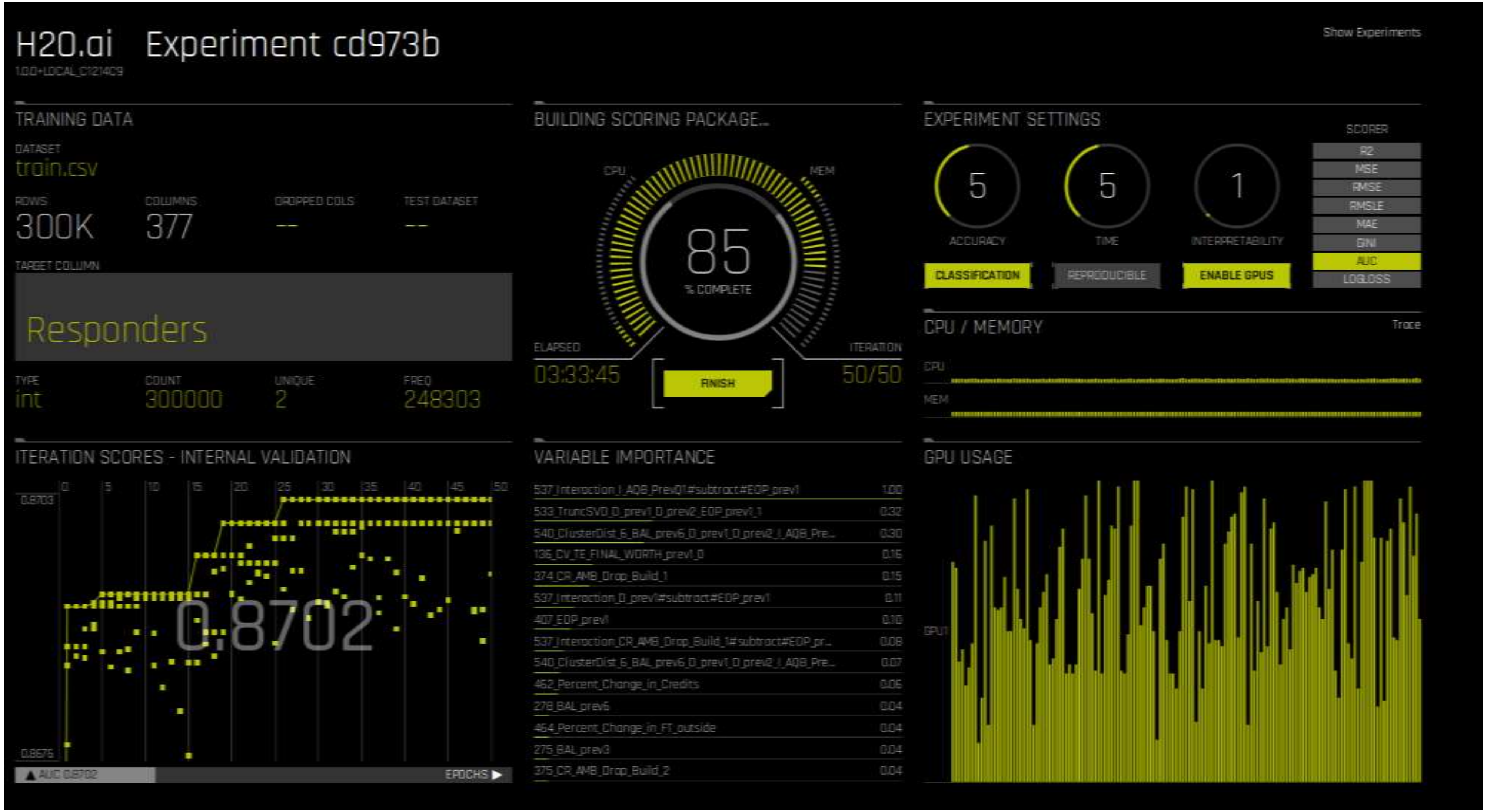
The Driverless Approach





Staples® Easy Button™

The Driverless Approach



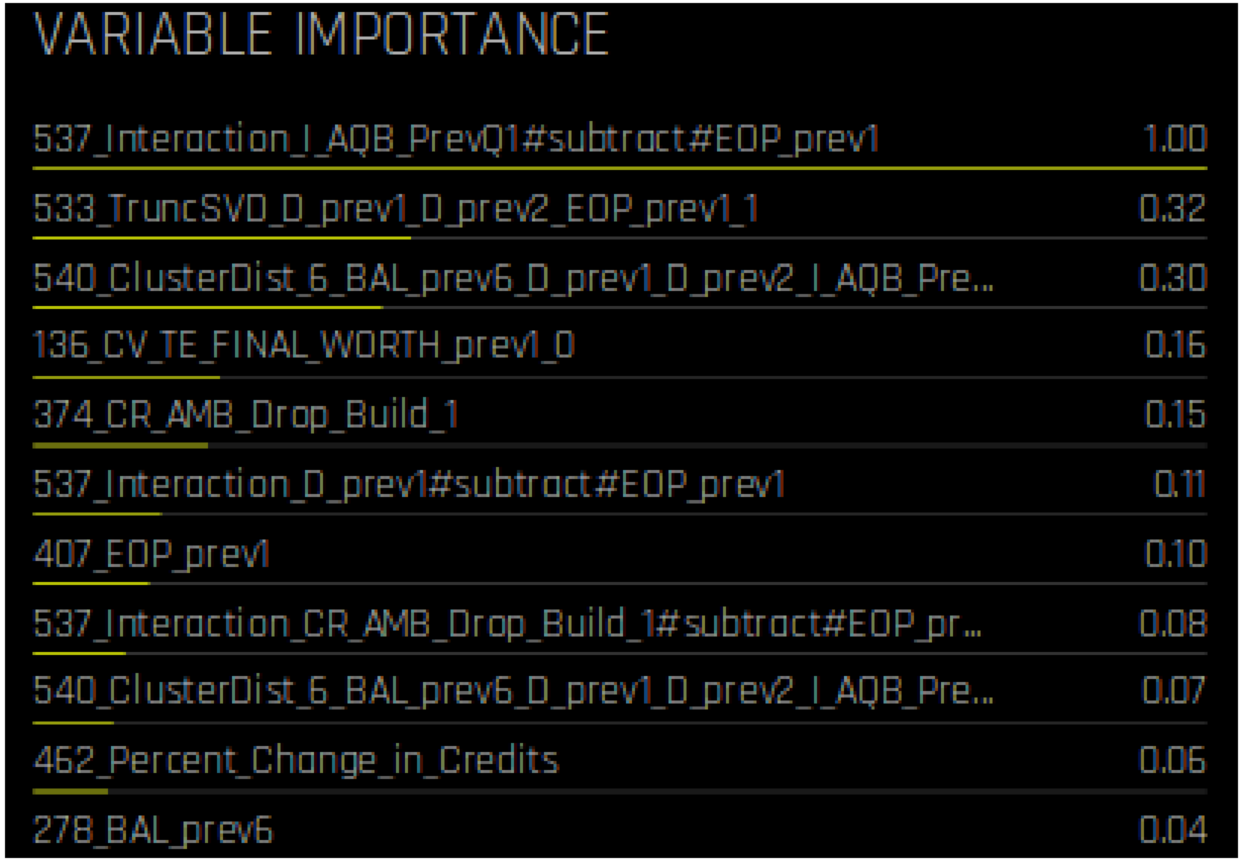
The Driverless Approach

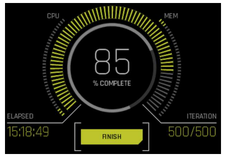
It knows math!

subtraction

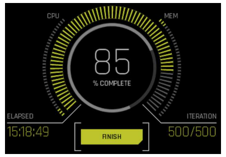
subtraction

subtraction





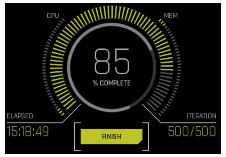
The Driverless Approach



Top 10 Features: divided into categories

- (3) Subtraction: #1, #6, #8
- (1) Truncated SVD components: #2
- (2) Cluster Distances: #3, #9
- (1) Target encoding: #4
- (3) Direct features: #5, #7, #10

The Driverless Approach

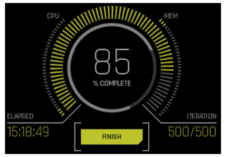


Lagged balances also used in clusters & SVD

- Distance to cluster #1 after segmenting columns into 6 clusters
  - BAL_prev6
  - D_prev1
  - D_prev2
  - I_AQB_PrevQ1
- Component #1 of truncated SVD of
  - D_prev1
  - D_prev2
  - EOP_prev1_1

The Driverless Approach

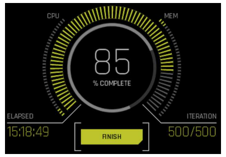
Final Analysis



- On first iteration, Driverless AI had surpassed my manual modeling
- Features were well beyond what I would have ever attempted
- Accuracy was stable: Driverless AI self-reported scores within 1% of competition submission

The Driverless Approach

Competition Results

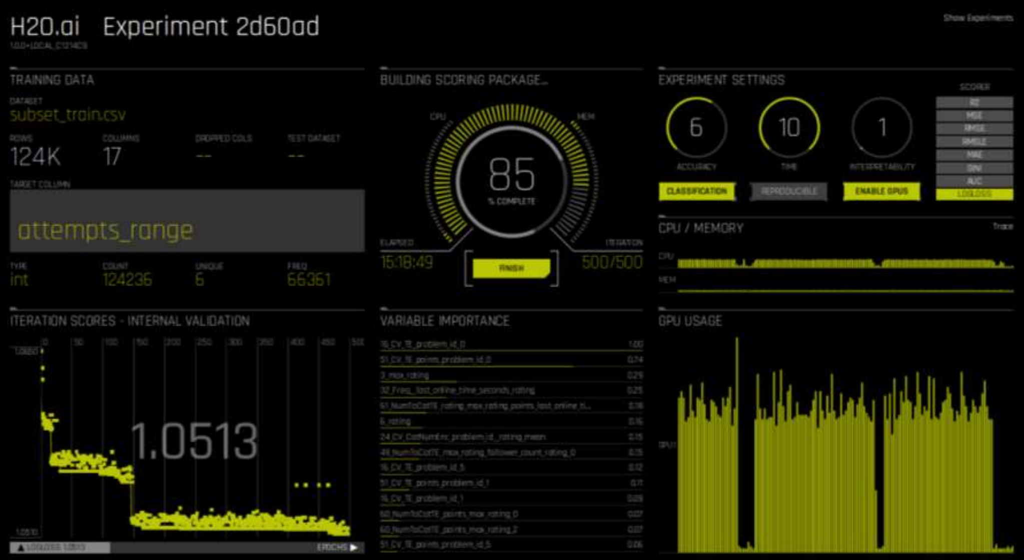


Private Leaderboard

	#	Name	Score
	1	Team Billa	0.680800
	2	ankit2106	0.678672
Kaggle Grandmaster	3	SRK	0.678595
	3	sadz2201	0.678595
100% Driverless AI	5	mark12	0.678440
	5	numb3r303	0.678440
Kaggle Grandmaster	7	Rohan Rao	0.677937
Also Driverless AI	8	ruben_diaz	0.677860

The End

Thank You



H2O WORLD 2017