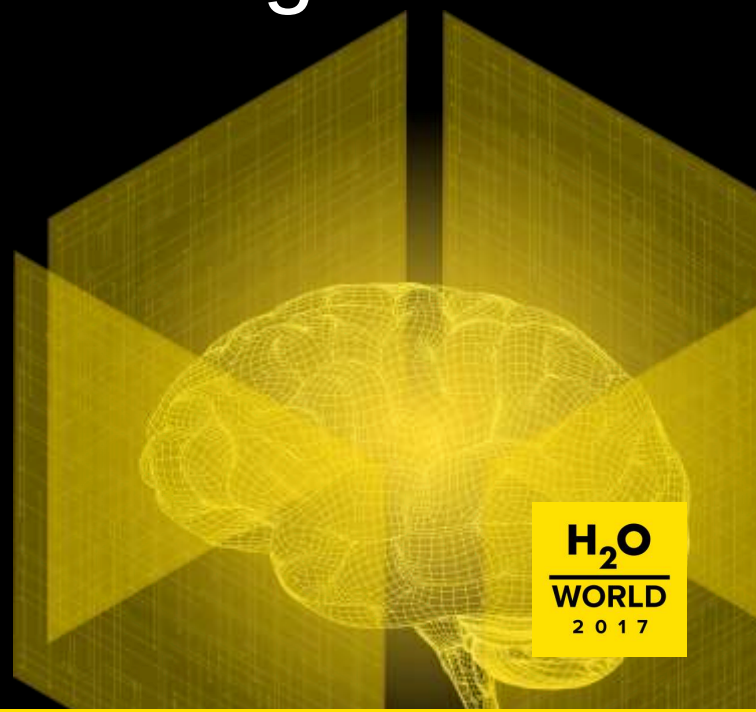

H2O
WORLD
2017

Leakage in Meta Modeling & Its Connection to HCC Target-Encoding

Mathias Müller

faron@h2o.ai
kaggle.com/mmueller - github.com/Far0n



H2O WORLD 2017

Background

- Born & raised in Berlin
- Diplom in Computer Science from Humboldt University of Berlin
- Joined **H2O** two month ago
  - Data Scientist
  - Development of **Driverless AI**

kaggle

Faron

Data Scientist at H2O.ai

Deutschland

Joined 3 years ago · last seen in the past day

http://kagglenizer.com/

Followers 412

Competitions Grandmaster

Home | Competitions (26) | Kernels (9) | Discussion (377) | Datasets (0) | ••• | Edit Profile

Competitions Grandmaster

Current Rank	Highest Rank
5 of 66,213	4

14 | 4 | 3

Homesite Quote Conversion · 2 years ago · Top 1%	1st	of 1764
Truly Native? · 2 years ago · Top 1%	1st	of 274
Two Sigma Connect: Renta... · 7 months ago · Top 1%	2nd	of 2488

Kernels Expert

Current Rank	Highest Rank
22 of 109,345	8

3 | 3 | 3

Road-2-0.4+ · a year ago	122	votes
F1-Score Expectation Maxi... · 4 months ago	105	votes
Stacking Starter · a year ago	84	votes

Discussion Master

Current Rank	Highest Rank
12 of 40,838	6

26 | 43 | 187

2nd Place Solution · 7 months ago	87	votes
Get Expected F1-Score in O... · 4 months ago	78	votes
Faron's 3rd Place Solution · a year ago	53	votes

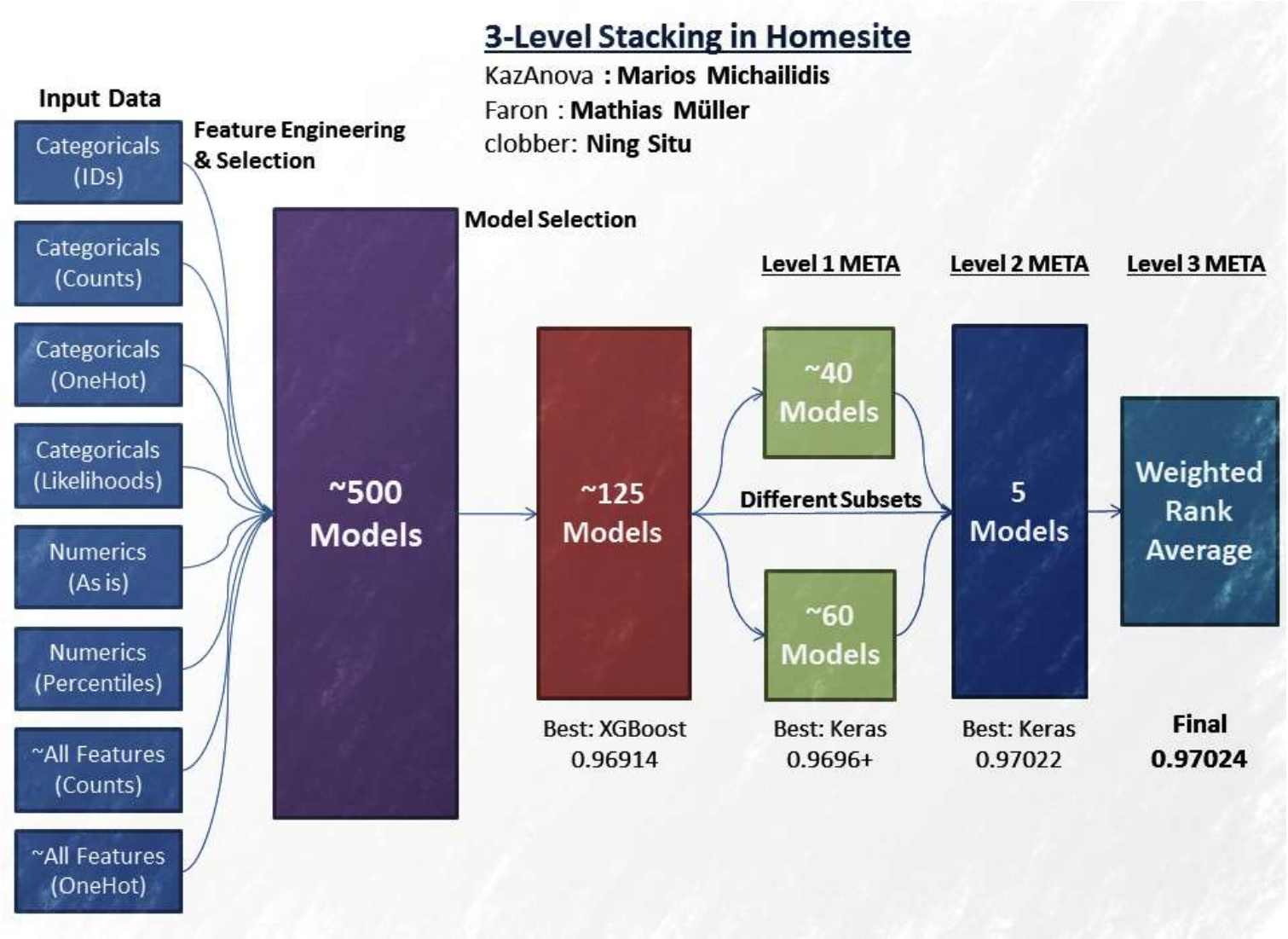
H2O WORLD 2017

Leakage

“Data Leakage is the creation of unexpected additional information in the training data, allowing a model or machine learning algorithm to make unrealistically good predictions.”
kaggle.com/wiki/leakage

- Many different sources
  - ID-Leaks
  - Leaking future information into past
  - Validating models on already seen data
  - **Leaking target information into Feature Matrices**
  - Feedback loops / adaptive data analysis
  - …
- Caused damage varies from case to case

Meta Modeling / Stacking



Leakage in Meta Modeling

- Suppose a 3-fold split of our training data into *(A,B,C)*
- Creating of out-of-fold predictions *(A2,B2,C2)*
  - ***train****((A,B))* followed by ***predict****(C)* to get *C2*
  - ***train****((A,C))* followed by ***predict****(B)* to get *B2*
  - ***train****((B,C))* followed by ***predict****(A)* to get *A2*

Base Level	1st Meta Level	Leaked Target Information
AB -> C2	A2B2 -> C3	(B**C**)(A**C**) -> **C**3
AC -> B2	A2C2 -> B3	(**B**C)(A**B**) -> **B**3
BC -> A2	B2C2 -> A3	(**A**C)(**A**B) -> **A**3

HCC Target Encoding

- In general, tree based models like XGBoost, LightGBM, RF, etc. struggle with (non-ordinal) High Cardinal Categoricals (HCC) features
- Order of mapped HCC values determines the required amount of splits to get “useful” data partitions
- Idea: Replace HCC values by their likelihoods to get a “good order”

K-Fold Target Encoding - Example

- We want to replace the categorical values **blue** and **red** by their likelihoods in a k-fold cross-validated fashion:

X	y	Fold	AB -> C	AC -> B	BC -> A	X_lhood_cv
blue	1	A			p(y | X = blue) = 0	**0**
red	1				p(y | X = red) = 1.0	**1**
blue	0	B		p(y | X = blue) = 0.5		**0.5**
blue	0					**0.5**
blue	0	C	p(y | X = blue) = 0.333			**0.333**
red	1		p(y | X = red) = 1.0			**1**

Recap: Leakage in Meta Modeling

Base Level	1st Meta Level	Leaked Target Information
AB -> C2	A2B2 -> C3	(B**C**)(A**C**) -> **C**3
AC -> B2	A2C2 -> B3	(**B**C)(A**B**) -> **B**3
BC -> A2	B2C2 -> A3	(**A**C)(**A**B) -> **A**3

K-Fold Target Encoding - Example

X	y	Fold	AB -> C	AC -> B	BC -> A	X_lhood_cv
blue	1	A			p(y | X = blue) = 0	**0**
red	1				p(y | X = red) = 1.0	**1**
blue	0	B		p(y | X = blue) = 0.5		**0.5**
blue	0					**0.5**
blue	0	C	p(y | X = blue) = 0.333			**0.333**
red	1		p(y | X = red) = 1.0			**1**

- **X_lhood_cv** values are basically “out-of-fold” predictions of a maximum likelihood estimator
- Using **X_lhood_cv** as feature is pretty much the same procedure as stacking
- Same leakage issue .. but fails more often than strong model stacking, because of no regularization

Counter-Measures

- Using a fixed holdout set to calculate likelihoods / to generate out-of-fold predictions
  - Loss of training data at later stages
- Using a 2-fold scheme with fixed seed
  - Not ideal regarding bias-variance-tradeoff
- Adding Noise to likelihoods / out-of-fold predictions
  - Hard to get the noise level right (heavily dataset dependent)
- Avoiding target leakage by nested cross validation
  - Order of magnitude higher complexitity: $O(k) \Rightarrow O(k_{outer} * k_{inner})$

Nested Cross Validation

Base Level	1st Meta Level	No Leaked Target Information
AB -> C2 A -> B2 B -> A2	A2B2 -> C3	(B)(A) -> C3
AC -> B2 A -> C2 C -> A2	A2C2 -> B3	(C)(A) -> B3
BC -> A2 B -> C2 C -> B2	B2C2 -> A3	(C)(B) -> A3

Thank you for your attention!

Any Questions?