Detecting Money Laundering FP with AI

Building Machine Learning Recipes to Detect Malicious Behaviours

Ashrith Barthur

H_2O.ai

September 17, 2019

Contents

Thanks

1. Audience
2. Our Customers
3. AI4 Team
4. H2O Team
5. Patrick Hall - Thanks for the beamer template.

Speaker

1. Principal Security Scientist at H2O.AI.
2. Researcher in the area of anomalies specific to malicious behaviour, graduated with a PhD from Purdue.
3. Areas - Cybersecurity, Electronic Fraud, Money Laundering, and Global Malicious Migratory Patterns.
4. When I am not working, I am usually biking, or taking pictures.
5. @ashrith (github) @cyberbaggage (twitter) @ba(h2o.ai)

Question

How many of us attending today's webinar analyse data and build models?

Overview

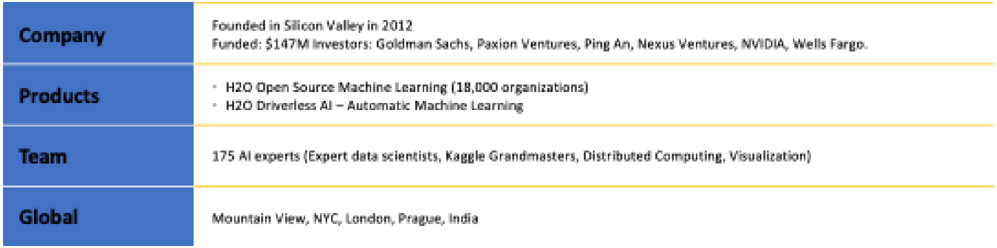
This presentation introduces us to three topics. These topics are:

1. An introduction to Detecting False Positive in Money Laundering Alerts using feature recipes.
2. A simple, starter guide to building your own feature recipes.
3. A 'peek' introduction to detecting malicious behaviours. Malicious Domains in cybersecurity.

Question

How many of the people attending today have known about H2O.AI?

H2O.AI Overview



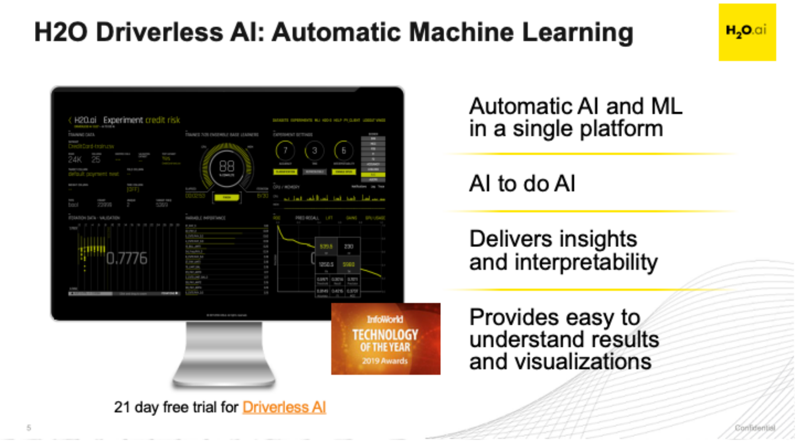
Company	Founded in Silicon Valley in 2012 Funded: $147M Investors: Goldman Sachs, Paxion Ventures, Ping An, Nexus Ventures, NVIDIA, Wells Fargo.
Products	• H2O Open Source Machine Learning (18,000 organizations) • H2O Driverless AI – Automatic Machine Learning
Team	175 AI experts (Expert data scientists, Kaggle Grandmasters, Distributed Computing, Visualization)
Global	Mountain View, NYC, London, Prague, India

Industry Footprint

Question

How many of the people attending today have known, or used Driverless AI?

Driverless AI



H2O Driverless AI: Automatic Machine Learning

InfoWorld
TECHNOLOGY OF THE YEAR
2019 Awards

Automatic AI and ML in a single platform

AI to do AI

Delivers insights and interpretability

Provides easy to understand results and visualizations

21 day free trial for Driverless AI

5

Confidential

Anti Money Laundering

What is money laundering?
"the concealment of the origins of illegally obtained money, typically by means of transfers involving foreign banks or legitimate businesses."

Anti Money Laundering

How are we solving Anti Money Laundering up until now?

1. We use Rule-Based systems that detect instances of money laundering.

Then what remains to be a problem?

1. The Rule-Based systems are slow to evolve.
2. Rule gaps, and complexities exist.
3. The have a high false positive rate - 75% to 99%

Anti Money Laundering

In this presentation we reduce the false positive rate of Anti Money Laundering alerts. How do we do this?

1. We show case a pre-built recipe for AML.
2. The feature recipe, comprehensively covers many different features that are potentially applicable for Anti-Money Laundering.
3. Through the feature recipe, we use the strengths of Driverless AI to build us the best model possible.

DEMO

Question

How many of us are familiar with Custom Transformers in Driverless AI?

How Did We Build This?

Driverless AI provides an extension.
This is a class ‘CustomTransformer‘

```
class ExampleLogTransformer(CustomTransformer):
```

How Did We Build This?

The class has:

1. Parameters that need to be provided.
2. These parameters are specific to the type of feature recipe that you are building.
3. It also has four methods which primary handle your feature engineering transformation.

Parameters - Basic

```
class ExampleLogTransformer(CustomTransformer):
_regression = True
_binary = True
_multiclass = True
```

Parameters - Advanced

```
class ExampleLogTransformer(CustomTransformer):
_regression = True
_binary = True
_multiclass = True
_numeric_output = True
_is_reproducible = True
_excluded_model_classes = ['tensorflow']
_modules_needed_by_name = ["custom_package==1.0.0"]
```

Acceptance Method

```
class ExampleLogTransformer(CustomTransformer):
_regression = True
_binary = True
_multiclass = True
_numeric_output = True
_is_reproducible = True
_excluded_model_classes = ['tensorflow']
_modules_needed_by_name = ["custom_package==1.0.0"]

@staticmethod
def do_acceptance_test():
return True
```

Input Data

```
...
@staticmethod
def do_acceptance_test():
return True

@staticmethod
def get_default_properties():
return dict(col_type = "numeric", min_cols = 1, max_cols = 1,
relative_importance = 1)
```

Input Data Types

```
a. "all"         - all column types
b. "any"         - any column types
c. "numeric"     - numeric int/float column
d. "categorical" - string/int/float column considered a categorical for
feature engineering
e. "numcat"      - allow both numeric or categorical
f. "datetime"    - string or int column with raw datetime such as
'%Y/%m/%d %H:%M:%S' or '%Y%m%d%H%M'
```

Input Data Types

```
g. "date"        - string or int column with raw date such as
'%Y/%m/%d' or '%Y%m%d'
h. "text"        - string column containing text
(and hence not treated as categorical)
i. "time_column" - the time column specified at the start of
the experiment (unmodified)
```

Fit Function

```
@staticmethod
def get_default_properties():
return dict(col_type = "numeric", min_cols = 1, max_cols = 1,
relative_importance = 1)

def fit_transform(self, X: dt.Frame, y: np.arry = None):
X_pandas = X.to_pandas()
X_p_log = np.log10(X_pandas)
return X_p_log
```

Transform Function

```
def fit_transform(self, X: dt.Frame, y: np.arry = None):
X_pandas = X.to_pandas()
X_p_log = np.log10(X_pandas)
return X_p_log

def transform(self, X: dt.Frame):
X_pandas = X.to_pandas()
X_p_log = np.log10(X_pandas)
return X_p_log
```

Machine Generated Domains Detection - Recipe

In this presentation we look at how a recipe built to detect machine generated domains works, and performs.

DEMO

Advantages

1. Feature engineering process standardised by:
   1.1 preset parameters
   1.2 preset methods
2. Effort minimisation leads to minimisation in time spent.
3. Build only once - Feature engineering is carried over from training/testing to production.
4. DAI automatically, runs multiple models on various sets of features to get the best model.
5. All the requirements are handled internally by DAI.

References

How to build a recipe

https://github.com/h2oai/driverlessai-recipes/tree/master/how_to_write_a_recipe

Questions

Thanks