One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability

Ronny Luss*
IBM Research AI
*Speaker

Joint Work with AIX360 Team at IBM Research.

Data Council NYC, November 2019.

- **Why Explainable AI?**
- Types and Methods for Explainable AI
- AIX360
  - CEM-MAF Example
  - FICO Example (BRCG and Protodash)

AI IS NOW USED IN MANY HIGH-STAKES DECISION MAKING APPLICATIONS

Credit

Employment

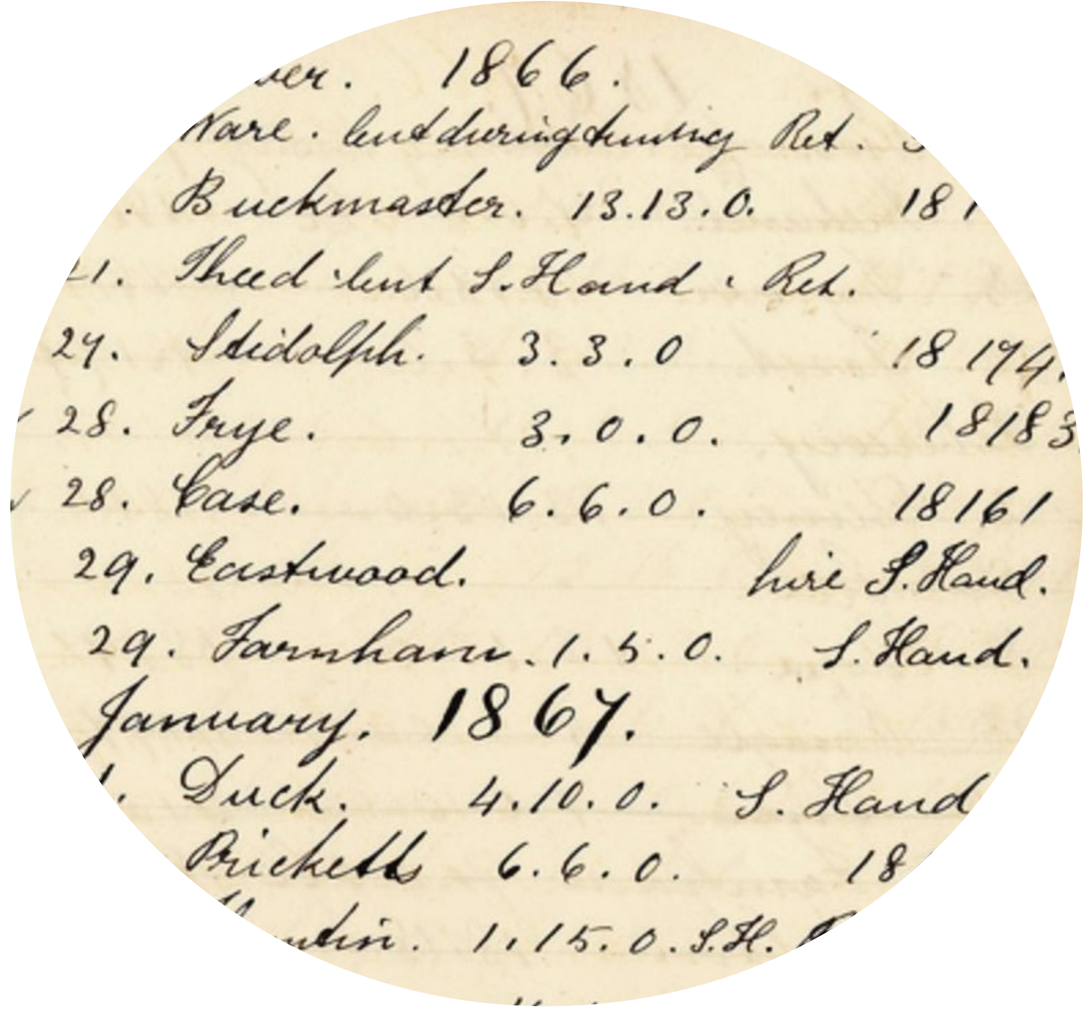
Admission

Sentencing

WHAT DOES IT TAKE TO TRUST A DECISION MADE BY A MACHINE (OTHER THAN THAT IT IS 99% ACCURATE)

Is it fair?

"Why" did it make this decision?



Is it accountable?

THE QUEST FOR “EXPLAINABLE AI”

CIO JOURNAL.

Companies Grapple With AI’s Opaque Decision-Making Process

THE WALL STREET JOURNAL.

Why Explainable AI Will Be the Next Big Disruptive Trend in Business

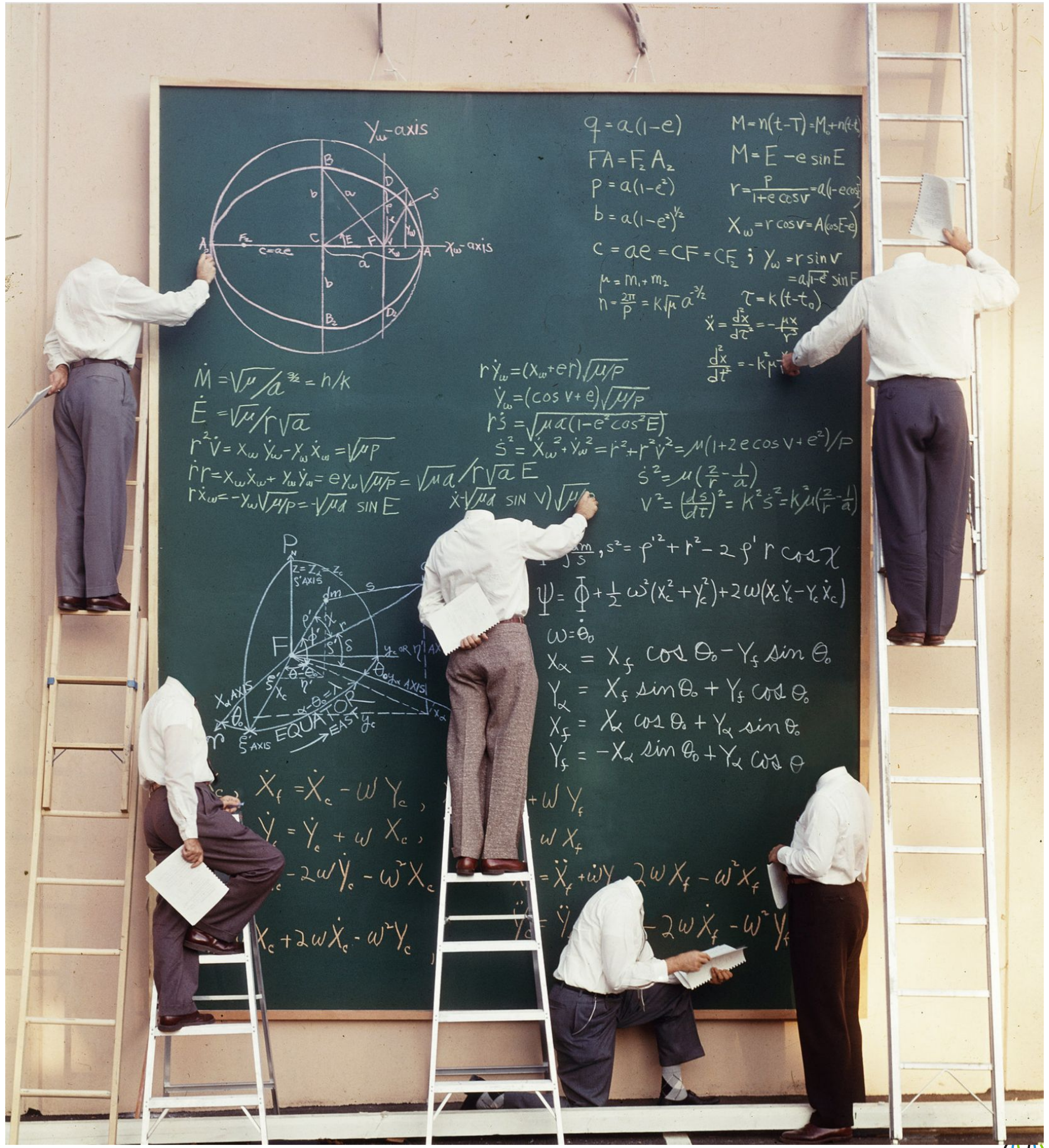
AlleyWatch

When a Computer Program Keeps You in Jail

The New York Times

Don't Trust Artificial Intelligence? Time To Open The AI 'Black Box'

Forbes



The General Data Protection Regulation (GDPR)

- Limits to decision-making based solely on automated processing and profiling (Art.22)
- Right to be provided with meaningful information about the logic involved in the decision (Art.13 (2) f and 15 (1) h)

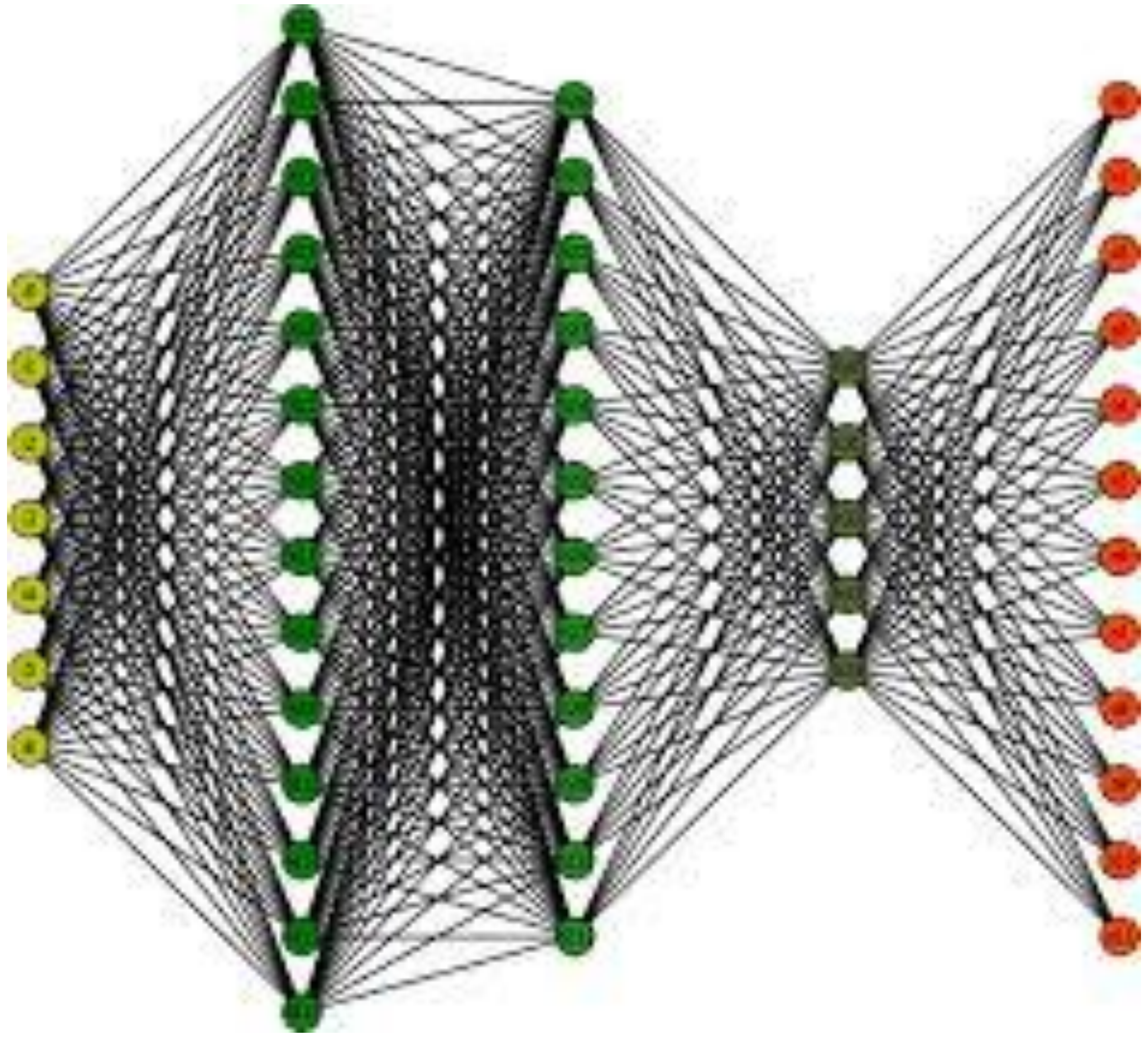
"meaningful"

???

Paul Nemitz, *Principal Advisor, European Commission*
Talk at IBM Research, Yorktown Heights, May, 4, 2018

Simplification

Understanding what’s truly happening can help build simpler systems.



Insight

Check if code has comments

Debugging

Can help to understand what is wrong with a system.

Self driving car slowed down but wouldn't stop at red light???

Existence of Confounders

Can help to identify spurious correlations.

Pneumonia

~~Diabetes~~

Fairness

Is the decision making system fair?

Robustness and Generalizability

Is the system basing decisions on the correct features?

Wide Spread Adoption

- Why Explainable AI?
- **Types and Methods for Explainable AI**
- AIX360
    - CEM-MAF Example
    - FICO Example (BRCG and Protodash)

AIX360: COMPETITIVE LANDSCAPE

Toolkit	Data Explanations	Directly Interpretable	Local Post-hoc	Global Post-hoc	Custom Explanation	Metrics
IBM AIX360	2	2	3	1	1	2
Seldon Alibi			✓	✓		
Oracle Skater		✓	✓	✓		
H2o		✓	✓	✓		
Microsoft Interpret		✓	✓	✓		
Ethical ML				✓		
DrWhyDalEx				✓		

All algorithms of AIX360 are developed by IBM Research
AIX360 also provides demos, tutorials, and guidance on explanations for different use cases.
Paper: One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques.
https://arxiv.org/abs/1909.03012v1

One explanation does not fit all: There are many ways to explain things.

directly interpretable vs. **post hoc interpretation**

Decision rule sets and trees are simple enough for people to understand. Supervised learning of these models is directly interpretable.

Probe a black-box with a companion model. The black box model provides actual predictions while the interpretation is thru the companion model

Global (model-level) vs. **Local (instance-level)**

Shows the entire predictive model to the user to help them understand it (e.g. a small decision tree, whether obtained directly or in a post hoc manner).

Only show the explanations associated with individual predictions (i.e. what was it about this particular person that resulted in her loan being denied).

static vs. **interactive (visual analytics)**

The interpretation is simply presented to the user.

The user can interact with interpretation.

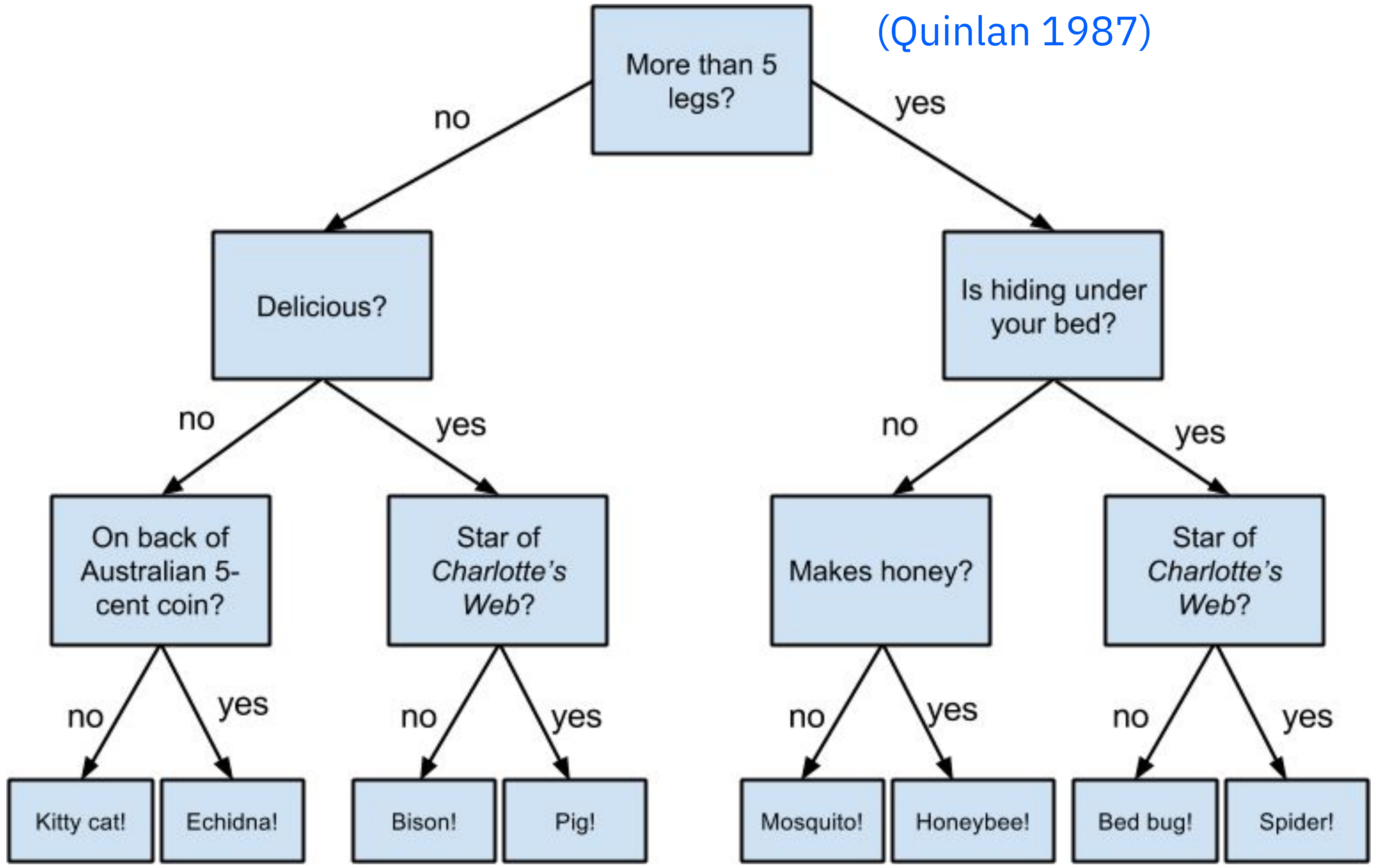
- EXPLAINABILITY TAXONOMY

Directly (global) interpretable

Decision rule sets and trees are simple enough for people to understand.

Decision Tree

(Quinlan 1987)



Rule List

(Wang and Rudin 2016)

if	capital-gain>$7298.00	then probability to make over 50K	= 0.986
else if	Young,Never-married,	then probability to make over 50K	= 0.003
else if	Grad-school,Married,	then probability to make over 50K	= 0.748
else if	Young,capital-loss=0,	then probability to make over 50K	= 0.072
else if	Own-child,Never-married,	then probability to make over 50K	= 0.015
else if	Bachelors,Married,	then probability to make over 50K	= 0.655
else if	Bachelors,Over-time,	then probability to make over 50K	= 0.255
else if	Exec-managerial,Married,	then probability to make over 50K	= 0.531
else if	Married,HS-grad,	then probability to make over 50K	= 0.300
else if	Grad-school,	then probability to make over 50K	= 0.266
else if	Some-college,Married,	then probability to make over 50K	= 0.410
else if	Prof-specialty,Married,	then probability to make over 50K	= 0.713
else if	Assoc-degree,Married,	then probability to make over 50K	= 0.420
else if	Part-time,	then probability to make over 50K	= 0.013
else if	Husband,	then probability to make over 50K	= 0.126
else if	Prof-specialty,	then probability to make over 50K	= 0.148
else if	Exec-managerial,Male,	then probability to make over 50K	= 0.193
else if	Full-time,Private,	then probability to make over 50K	= 0.026
else	(default rule)	then probability to make over 50K	= 0.066.

Directly (global) interpretable

Boolean Decision Rules via Column Generation (BRCG):

(Dash et. al. 2018)

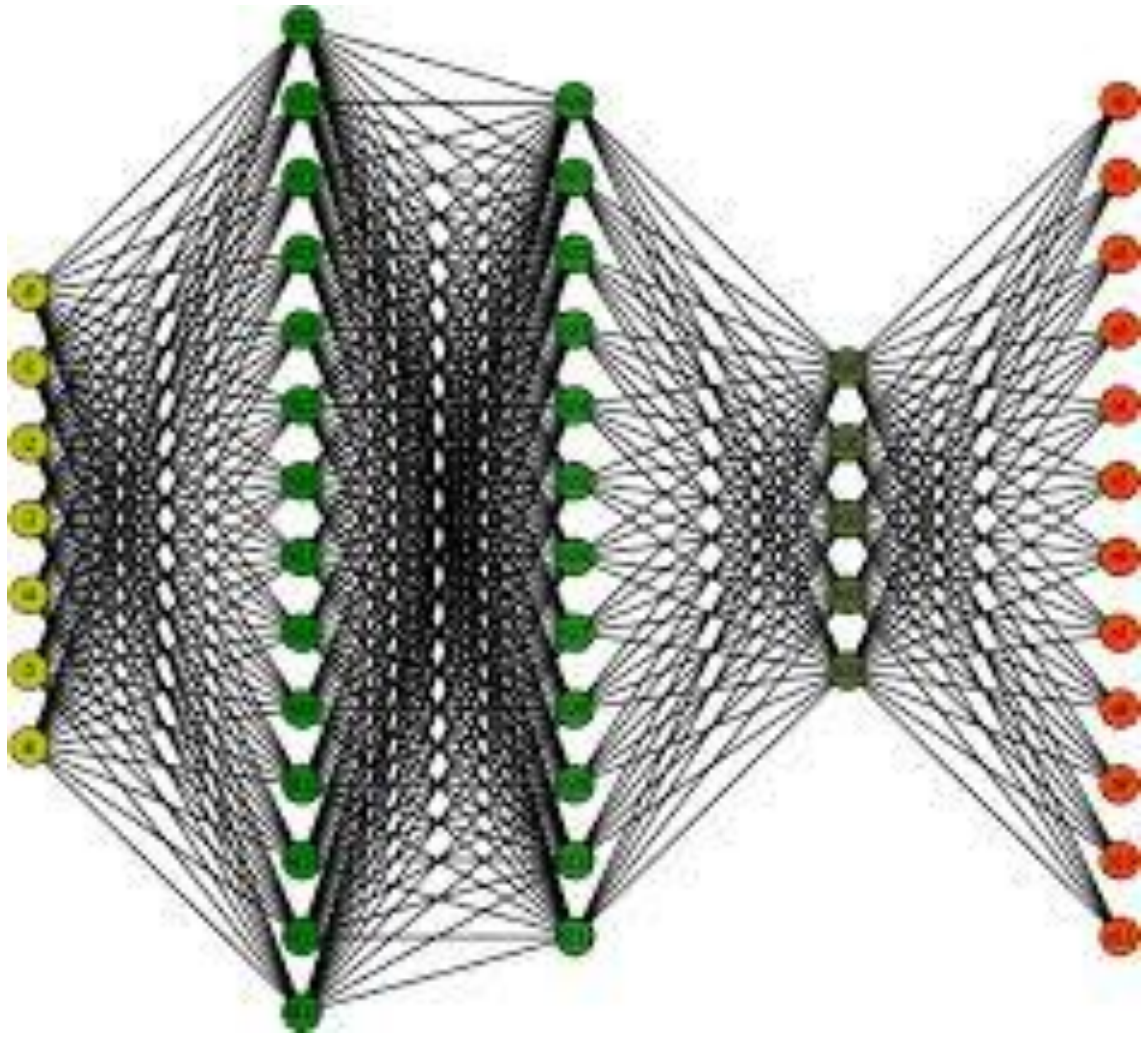
- DNF formulas (OR of ANDs) with small clauses to predict a binary target.
- Exponential number of possible clauses
- Fitting with DNFs as a Mixed Integer Program.
- Column Generation
  - Use few clauses to start with – solve the MIP.
  - Use a Pricing Problem on dual variables to identify the best clauses that still increase prediction accuracy – efficient step.
  - Iterate - stop when nothing more can be added.
- Scales to datasets of size ~ 10000 samples.

A variant is in AIX360. This technique won the **NeurIPS '18 FICO xML Challenge** !!

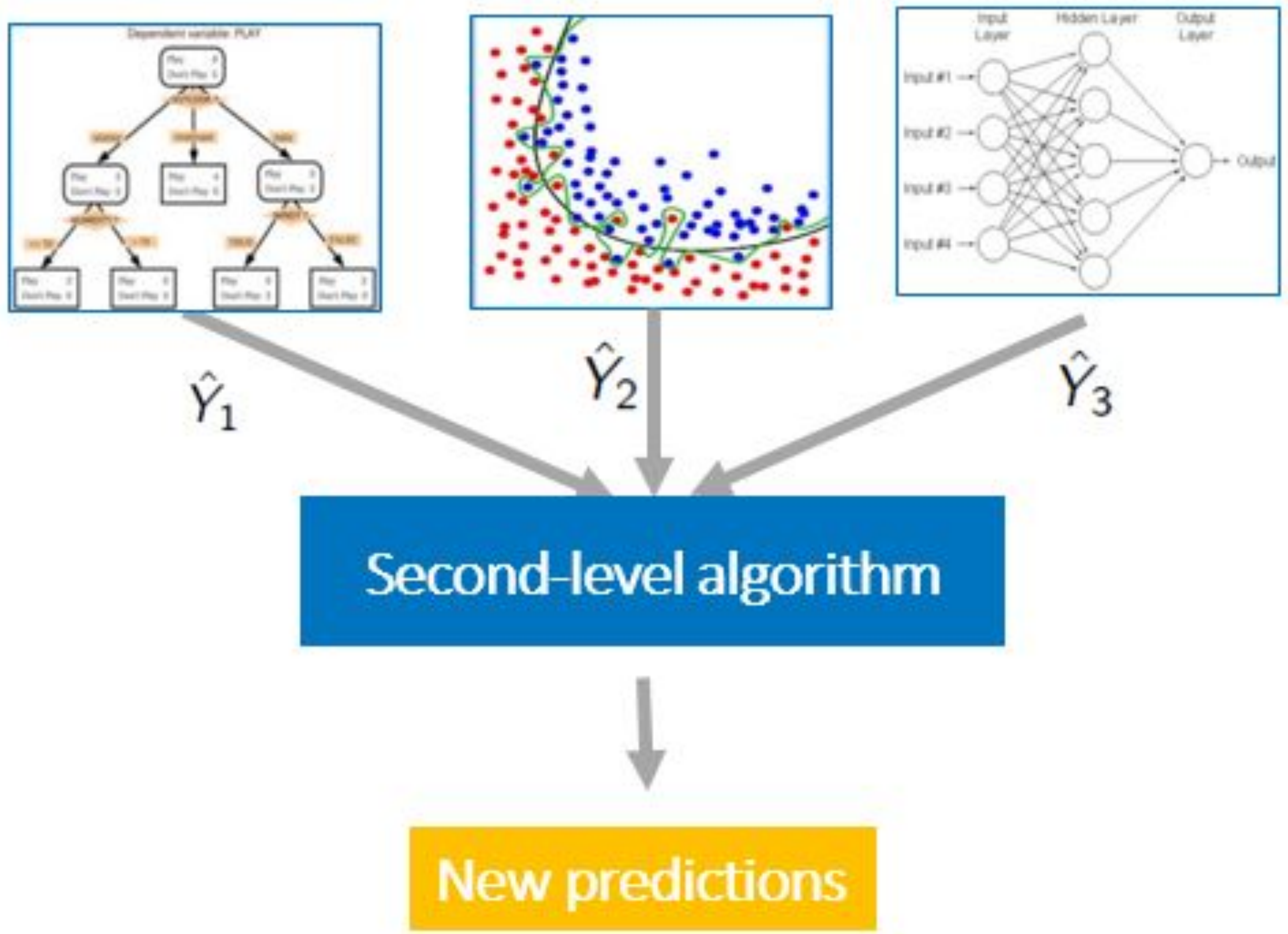
Post hoc interpretation

Start with a black box model and probe into it with a companion model to create interpretations.

(Deep) Neural Network

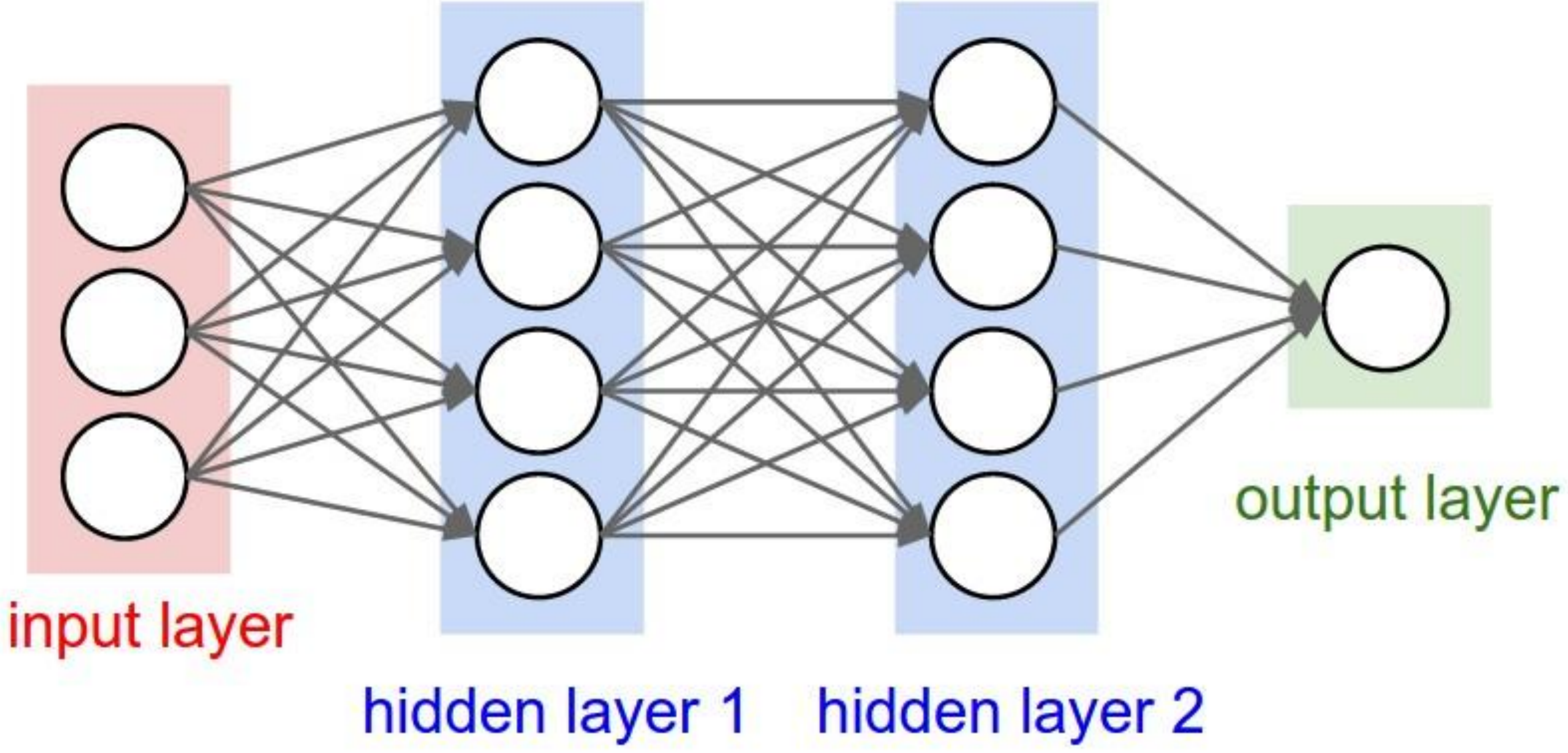


Ensembles



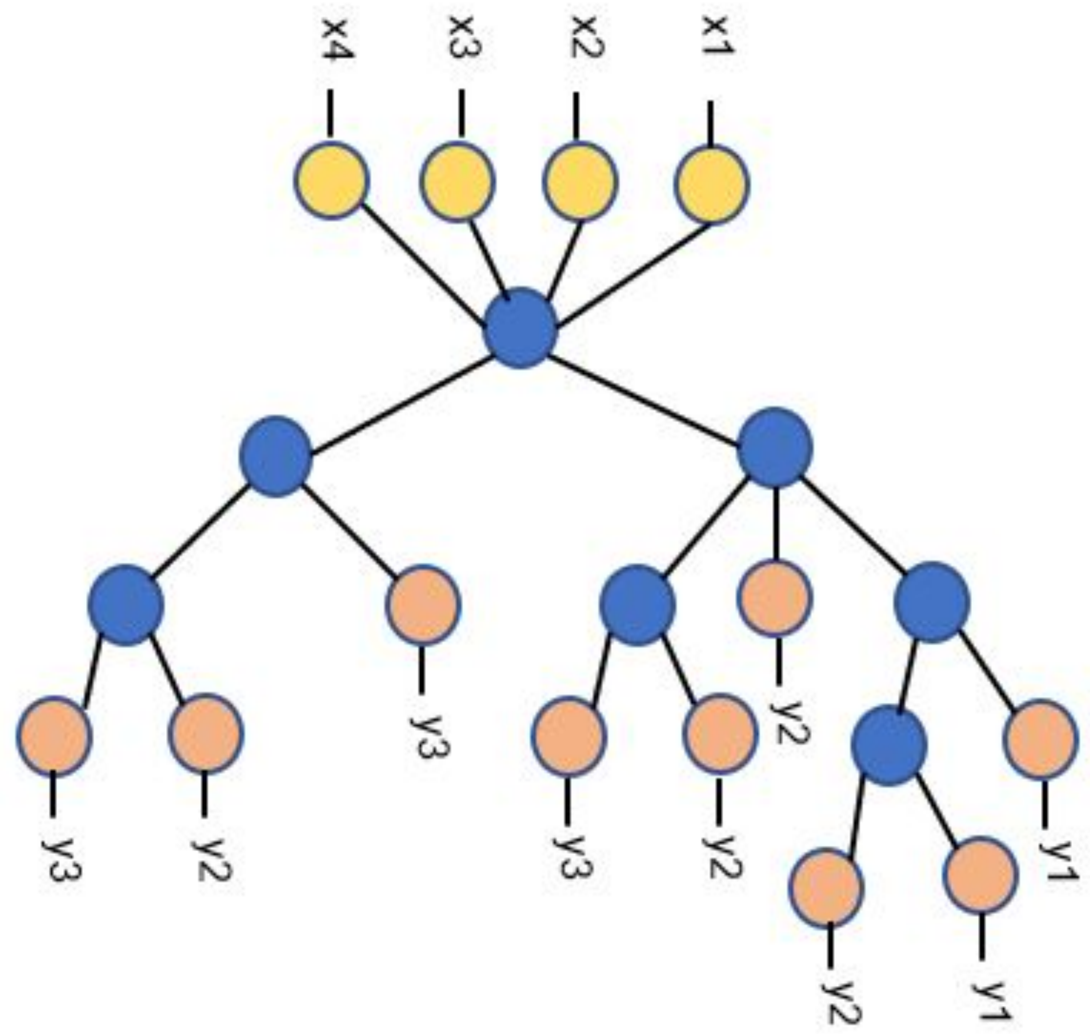
Post hoc (global) interpretation

Complex Model (Deep Neural Network)



Can you transfer information from a pre-trained neural network to this simple model ?

Simple Model (Decision Tree, Random forests, smaller neural network)



Post hoc (global) interpretation

Knowledge Distillation (Hinton et. al. 2015)

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right)$$

Re-train a simple model with temperature scaled soft scores of complex model.

Works Well

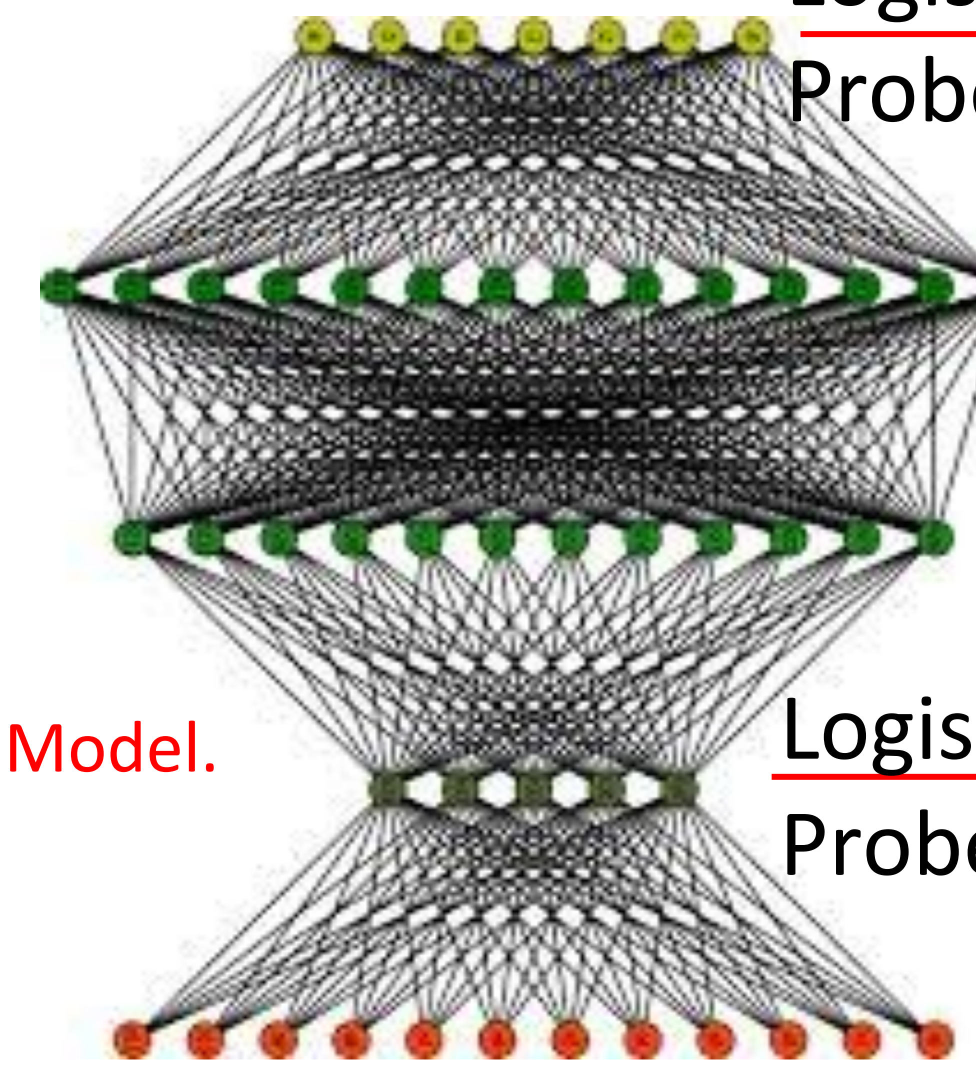
When Simple Model's complexity is comparable to Complex Model –ideal for compression

When Simple Model complexity is very small compared to Complex Model.

Prof-Weight (Dhurandhar et. al. 2018)

Re-train a simple model by weighing samples. Weights obtained by looking at inner layers of Complex Model.

Weight= $(p_1+ p_2+p_3+p_4)/4$



High -> Easy sample

Low->Difficult sample

Post hoc (local) interpretation

Saliency Maps

(Sinmoyan et. al. 2013)

$$w = \left. \frac{\partial S_c}{\partial I} \right|_{I_0}.$$

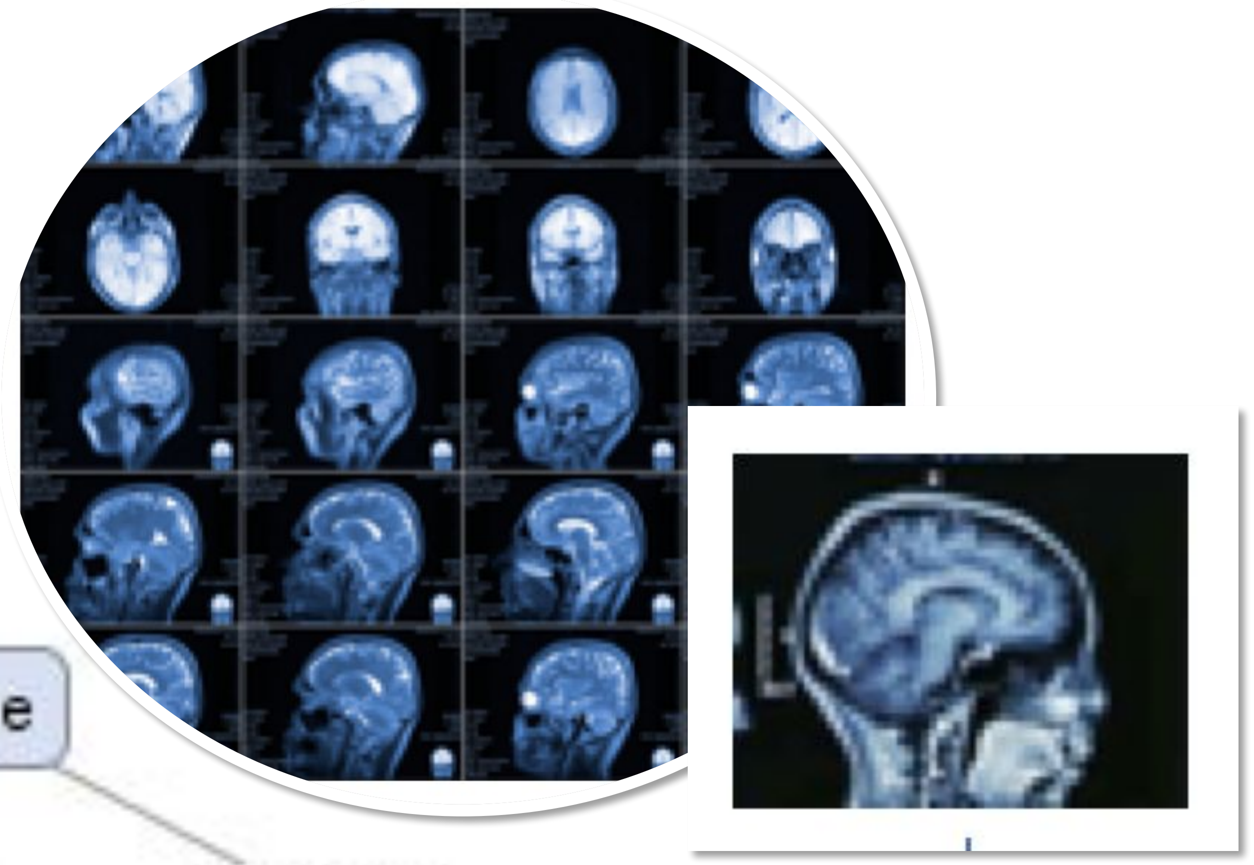
Post hoc (local) interpretation

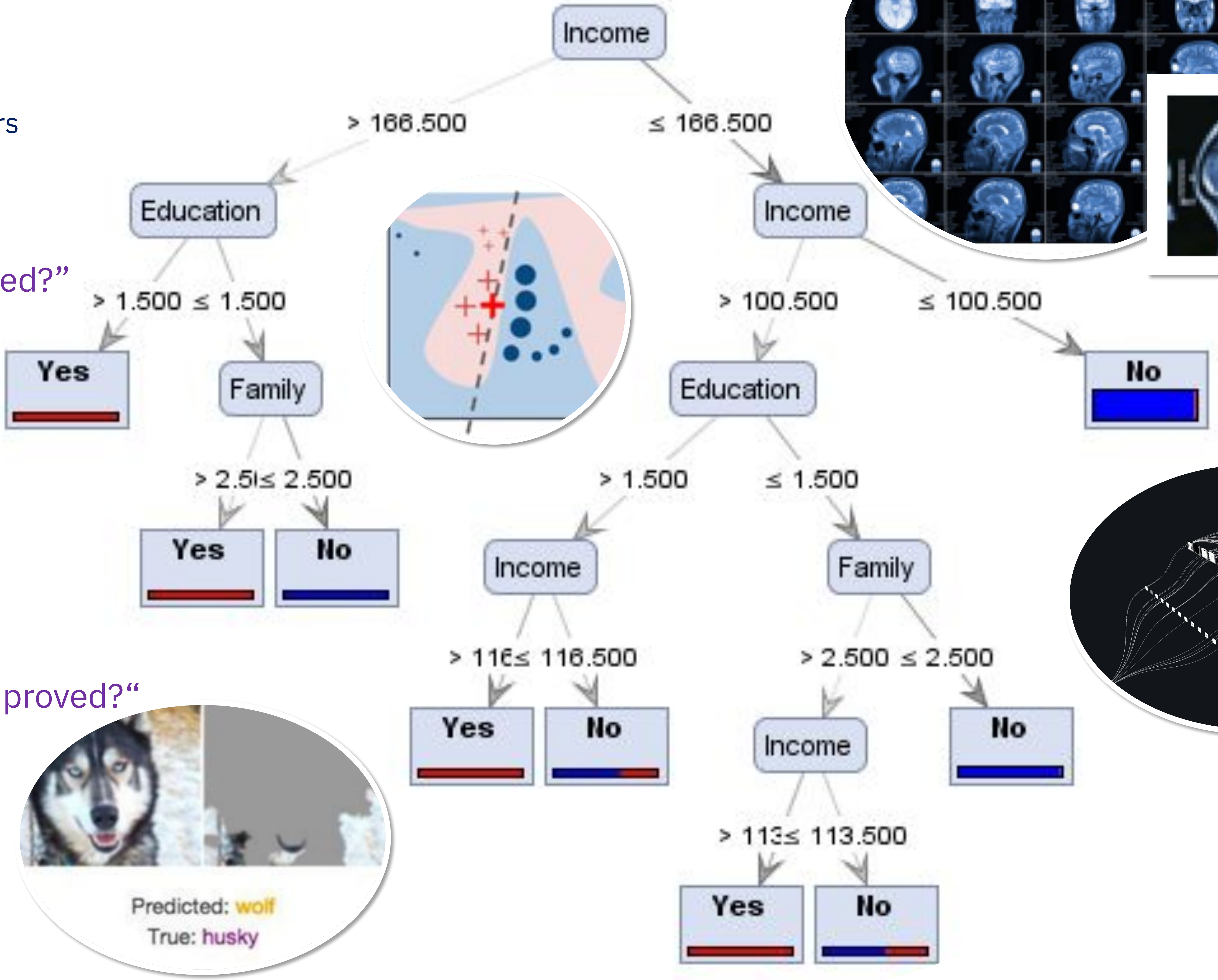
Contrastive Explanations – "Pertinent Negatives" (CEM-MAF):

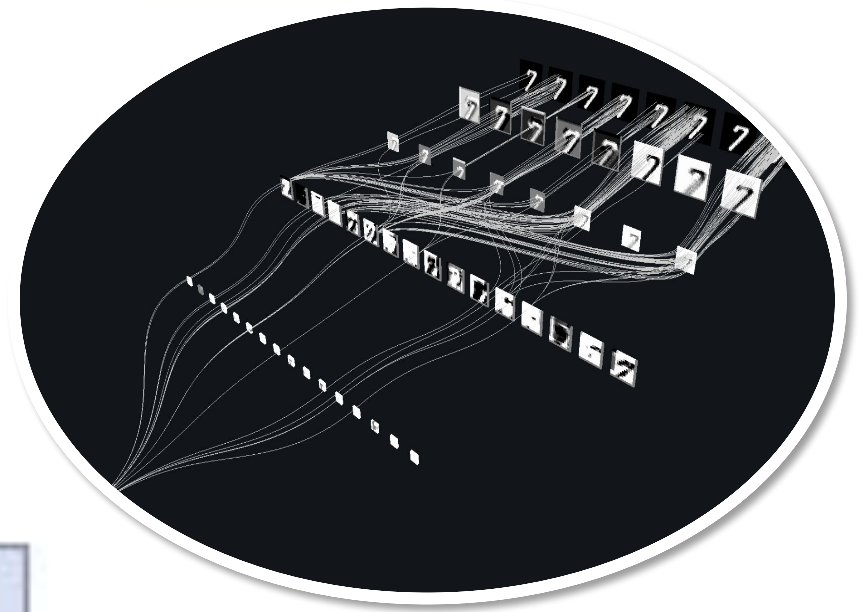
(Dhurandhar et. al. 2018)

Original Class Pred	yng, ml, smlg	yng, fml, smlg	yng, fml, not smlg
Original			
Pert. Neg. Class Pred	**old**, **fml**, smlg	**old**, fml, smlg	yng, **ml**, not smlg
Pertinent Negative			
Pert. Neg. Expla -nations	+brwn hr, +makeup, +bangs	+oval face	+single hair clr, -bangs

Different stakeholders require explanations for different purposes and with different objectives. Explanations will have to be tailored to their needs.







End users

"Why did you recommend this treatment?"

Who: Physicians, judges, loan officers, teacher evaluators

Why: trust/confidence, insights(?)

Affected users

"Why was my loan denied? How can I be approved?"

Who: Patients, accused, loan applicants, teachers

Why: understanding of factors

Regulatory bodies

"Prove that your system didn't discriminate."

Who: EU (GDPR), NYC Council, US Gov't, etc.

Why: ensure fairness for constituents

AI system builders/stakeholders

"Is the system performing well? How can it be improved?"

Who: EU (GDPR), NYC Council, US Gov't, etc.

Why: ensure or improve performance

Agenda

- Why Explainable AI?
- Types and Methods for Explainable AI
- **AIX360**
    - CEM-MAF Example
    - FICO Example (BRCG and Protodash)

AI EXPLAINABILITY 360 (V0.1.0)

AI Explainability 360 (v0.1.0)

build passing | docs passing | pypi package 0.1.0

The AI Explainability 360 toolkit is an open-source library that supports interpretability and explainability of datasets and machine learning models. The AI Explainability 360 Python package includes a comprehensive set of algorithms that cover different dimensions of explanations along with proxy explainability metrics.

The AI Explainability 360 interactive experience provides a gentle introduction to the concepts and capabilities by walking through an example use case for different consumer personas. The tutorials and example notebooks offer a deeper, data scientist-oriented introduction. The complete API is also available.

There is no single approach to explainability that works best. There are many ways to explain: data vs. model, directly interpretable vs. post hoc explanation, local vs. global, etc. It may therefore be confusing to figure out which algorithms are most appropriate for a given use case. To help, we have created some guidance material and a chart that can be consulted.

We have developed the package with extensibility in mind. This library is still in development. We encourage the contribution of your explainability algorithms and metrics. To get started as a contributor, please join the AI Explainability 360 Community on Slack by requesting an invitation here. Please review the instructions to contribute code here.

AI EXPLAINABILITY 360 (V0.1.0)

aix360	lime and shap	15 days ago
docs	lime and shap	15 days ago
examples	Merge pull request #41 from vijay-arya/master	14 days ago
tests	lime and shap	14 days ago
.gitignore	lime integration	17 days ago
.readthedocs.yml	doc issues	3 months ago
.travis.yml	lime and shap	15 days ago
CONTRIBUTING.md	Update CONTRIBUTING.md	3 months ago
LICENSE	Initial commit	4 months ago
MAINTAINERS.md	Update MAINTAINERS.md	3 months ago
README.md	Merge pull request #31 from sadhamanus/master	2 months ago
setup.py	lime and shap	15 days ago

AI EXPLAINABILITY 360 (V0.1.0)

contrastive	first version	3 months ago
dipvae	first version	3 months ago
lime	lime and shap	15 days ago
metrics	first version	3 months ago
profwt	Changes to the ProfWt notebook fixing directory refs	3 months ago
protodash	first version	3 months ago
rbm	first version	3 months ago
shap	lime and shap	15 days ago
tutorials	Merge pull request #39 from IBM/ted-notebook-update	17 days ago
README.md	add miss dot	2 months ago

Explaining Neural Network Decisions on Data that have High-level Attributes

CEM_MAFImageExplainer from AIX360 can be used to obtain contrastive explanations on data that have pre-defined high-level attributes, such as facial images that are annotated with features such as smile, high cheekbones, makeup, etc.

The goal of this tutorial is to demonstrate the use of CEM_MAFImageExplainer, which offers two-part explanations based on a pertinent positive and a pertinent negative. The pertinent positive explanation outputs the minimal set of high-level features that must be present in order for the classification of a sample to remain the same, so that if any one of the output features was missing from the sample, the classification would be different. The pertinent negative explanation outputs a set of features that would cause a change to the classification if they were added to the sample.

Import statements

```
import tensorflow as tf
import sys
import os
import numpy as np
import random
import matplotlib.pyplot as plt
from zipfile import ZipFile

from aix360.algorithms.contrastive import CEM_MAFImageExplainer
from aix360.algorithms.contrastive import CELEBAModel
from aix360.algorithms.contrastive import KerasClassifier
from aix360.algorithms.contrastive.dwnld_CEM_MAF_celebA import dwnld_CEM_MAF_celebA
from aix360.datasets.celeba_dataset import CelebADataset

dwnld = dwnld_CEM_MAF_celebA()
```


A Tensorflow session is required to run this example

```
sess = tf.InteractiveSession()
random.seed(120)
np.random.seed(1210)
sess.run(tf.global_variables_initializer())
```

Load CelebA model to be explained. Model must first be downloaded.

```
# Download pretrained celebA model
local_path_models = '../../aix360/models/CEM_MAF'
celebA_model_file = dwnld.dwnld_celebA_model(local_path_models)
```

```
celebA model file downloaded:
['../../aix360/models/CEM_MAF/celebA']
```

```
# Load the downloaded celebA model
model_file = '../../aix360/models/CEM_MAF/celebA'
loaded_model = CELEBAModel(restore=model_file, use_softmax=False).model
```

```
Load: ../../aix360/models/CEM_MAF/celebA
```

Wrap the CelebA model into a framework independent class structure

```
mymodel = KerasClassifier(loaded_model)
```

AI EXPLAINABILITY 360 (V0.1.0): CONTRASTIVE EXPLANATIONS VIA CEM-MAF

Download a sample image. Note: img_id must be from the following list: [2, 3, 4, 9, 11, 13, 15, 16, 18, 20]. These images are stored publicly and are downloaded here using the function dwnld.dwnld_celebA_data. The second argument is a list of the image ids to be downloaded.¶

```
img_id = 15
local_path_img =  '../../aix360/data/celeba_data'
img_files = dwnld.dwnld_celebA_data(local_path_img, [img_id])
```

```
Image files downloaded:
['../../aix360/data/celeba_data/15_img.npy', '../../aix360/data/celeba_data/15img.png', '../../aix360/da
ta/celeba_data/15_latent.npy']
```

Load the image and its latent representations, both to be used to generate a pertinent negative for the sample image. Then process the image and plot.

```
dataset_obj = CelebADataset(local_path_img) # use the CelebA dataset class
input_img = dataset_obj.get_img(img_id)
input_latent = dataset_obj.get_latent(img_id)

# images are processed according to needs for model being explained
input_img = np.clip(input_img/2, -0.5, 0.5)

plt.axis("off")
plt.imshow(input_img[0,:,:,:]+0.5)
plt.show()
```

AI EXPLAINABILITY 360 (V0.1.0): CONTRASTIVE EXPLANATIONS VIA CEM-MAF

Predict sample image using 8-class classifier based on 3 binary attributes: young (0 for old, 1 for young), smiling (0 for not smiling), 1 for smiling, and sex (0 for female, 1 for male).

```
orig_prob, orig_class, orig_prob_str = mymodel.predict_long(input_img)
# Compute classes
young_flag = orig_class % 2
smile_flag = (orig_class // 2) % 2
sex_flag = (orig_class // 4) % 2

arg_img_name = os.path.join(local_path_img, "{}_img.png".format(img_id))
print("Image:{}, pred:{}".format(arg_img_name, orig_class))
print("Male:{}, Smile:{}, Young:{}".format(sex_flag, smile_flag, young_flag))
orig_img = input_img
target_label = [np.eye(mymodel._nb_classes)[orig_class]]
```

```
Image:../../aix360/data/celeba_data/15_img.png, pred:4
Male:1, Smile:0, Young:0
```

AI EXPLAINABILITY 360 (V0.1.0): CONTRASTIVE EXPLANATIONS VIA CEM-MAF

Set up a CEM_MAF explainer object with respect to the trained CelebA model and high-level attributes.

```
aix360_path = '../../aix360' # needed to find paths to attribute files
explainer = CEM_MAFImageExplainer(mymodel, attributes, aix360_path)
```

Obtain the pertinent negative explaination

```
# parameter values for the pertinent negative
arg_mode = 'PN'
arg_kappa = 5
arg_gamma = 1
arg_binary_search_steps = 1
arg_max_iterations = 250
arg_initial_const = 10
arg_attr_reg = 100.0
arg_attr_penalty_reg = 100.0
arg_latent_square_loss_reg = 1.0
```

```
(adv_pn, attr_pn, info_pn) = explainer.explain_instance(sess, input_img,
                    input_latent, arg_mode, arg_kappa, arg_binary_search_steps,
                    arg_max_iterations, arg_initial_const, arg_gamma, None,
                    arg_attr_reg, arg_attr_penalty_reg,
                    arg_latent_square_loss_reg)

print(info_pn)
```

AI EXPLAINABILITY 360 (V0.1.0): CONTRASTIVE EXPLANATIONS VIA CEM-MAF

```
Loaded model for Black_Hair from disk
Loaded model for Blond_Hair from disk
Loaded model for Brown_Hair from disk
Loaded model for Gray_Hair from disk
Loaded model for Wearing_Lipstick from disk
Loaded model for Heavy_Makeup from disk
Loaded model for High_Cheekbones from disk
Loaded model for Bangs from disk
Loaded model for Oval_Face from disk
Loaded model for Narrow_Eyes from disk
Loaded model for Bags_Under_Eyes from disk
Loaded model for Pointy_Nose from disk
# of attr models is 12
iter:0 const:[10.]
Loss_Overall:7385.9272, Loss_Attack:0.0000, Loss_attr:1.2358
Loss_Latent_L2Dist:20.1870, Loss_Img_L2Dist:7021.6318
target_lab_score:-2.7435, max_nontarget_lab_score:5.0185

iter:10 const:[10.]
Loss_Overall:5924.4990, Loss_Attack:0.0000, Loss_attr:0.8909
Loss_Latent_L2Dist:1159.0074, Loss_Img_L2Dist:4563.6479
target_lab_score:0.6903, max_nontarget_lab_score:6.1033

iter:20 const:[10.]
Loss_Overall:3637.8708, Loss_Attack:0.0000, Loss_attr:0.5807
Loss_Latent_L2Dist:789.6542, Loss_Img_L2Dist:2642.2075
target_lab_score:-1.1012, max_nontarget_lab_score:6.5360
```


AI EXPLAINABILITY 360 (V0.1.0): CONTRASTIVE EXPLANATIONS VIA CEM-MAF

```
iter:240 const:[10.]
Loss_Overall:2083.2930, Loss_Attack:0.0000, Loss_attr:0.4539
Loss_Latent_L2Dist:265.6353, Loss_Img_L2Dist:1643.6389
target_lab_score:-1.6940, max_nontarget_lab_score:7.6957

[INFO] Orig class:4, Adv class:6, Orig prob:[[-5.7961226 -6.4976497 -5.1008477 -6.8349266  4.7267528 -3.
3094192  -3.0575497 -4.120827 ]], Adv prob:[[-6.0269275  -4.697468   -3.5946152  -4.4234085  -0.62827617
-5.6178136   7.752234    0.4052744 ]]
```

```python
plt.axis("off")
plt.imshow(adv_pn[0,:,:,:]+0.5)
plt.show()

# Compute new classes
adv_prob, adv_class, adv_prob_str = mymodel.predict_long(adv_pn)
young_flag = adv_class % 2
smile_flag = (adv_class // 2) % 2
sex_flag = (adv_class // 4) % 2
print("Pertinent Negative pred:{}".format(adv_class))
print("Male:{}, Smile:{}, Young:{}".format(sex_flag, smile_flag, young_flag))
print(attr_pn)
```

```
Pertinent Negative pred:6
Male:1, Smile:1, Young:0
Added High_Cheekbones
```

Credit Approval Tutorial

This tutorial illustrates the use of several methods in the AI Explainability 360 Toolkit to provide different kinds of explanations suited to different users in the context of a credit approval process enabled by machine learning. We use data from the FICO Explainable Machine Learning Challenge as described below. The three types of users (a.k.a. consumers) that we consider are a data scientist, who evaluates the machine learning model before deployment, a loan officer, who makes the final decision based on the model's output, and a bank customer, who wants to understand the reasons for their application result.

For the data scientist, we present two directly interpretable rule-based models that provide global understanding of their behavior. These models are produced by the Boolean Rule Column Generation (BRCG, class `BooleanRuleCG`) and Logistic Rule Regression (LogRR, class `LogisticRuleRegression`) algorithms in AIX360. The former yields very simple OR-of-ANDs classification rules while the latter gives weighted combinations of rules that are more accurate and still interpretable.

For the loan officer, we demonstrate a different way of explaining machine learning predictions by showing examples, specifically *prototypes* or representatives in the training data that are similar to a given loan applicant and receive the same class label. We use the ProtoDash method (class `ProtodashExplainer`) to find these prototypes.

For the bank customer, we consider the Contrastive Explanations Method (CEM, class `CEMExplainer`) for explaining the predictions of black box models to end users. CEM builds upon the popular approach of highlighting features present in the input instance that are responsible for the model's classification. In addition to these, CEM also identifies features that are (minimally) absent in the input instance, but whose presence would have altered the classification.

Data scientist: Boolean Rule and Logistic Rule Regression models

```
# Load FICO HELOC data with special values converted to np.nan
from aix360.datasets.heloc_dataset import HELOCDataset, nan_preprocessing
data = HELOCDataset(custom_preprocessing=nan_preprocessing).data()
# Separate target variable
y = data.pop('RiskPerformance')

# Split data into training and test sets using fixed random seed
from sklearn.model_selection import train_test_split
dfTrain, dfTest, yTrain, yTest = train_test_split(data, y, random_state=0, stratify=y)
dfTrain.head().transpose()
```

	8960	8403	1949	4886	4998
ExternalRiskEstimate	64.0	57.0	59.0	65.0	65.0
MSinceOldestTradeOpen	175.0	47.0	168.0	228.0	117.0
MSinceMostRecentTradeOpen	6.0	9.0	3.0	5.0	7.0
AverageMInFile	97.0	35.0	38.0	69.0	48.0
NumSatisfactoryTrades	29.0	5.0	21.0	24.0	7.0
NumTrades60Ever2DerogPubRec	9.0	1.0	0.0	3.0	1.0
NumTrades90Ever2DerogPubRec	9.0	0.0	0.0	2.0	1.0

Sample of
FICO HELOC data

BRCG requires data to be binarized

```
# Binarize data and also return standardized ordinal features
from aix360.algorithms.rbm import FeatureBinarizer
fb = FeatureBinarizer(negations=True, returnOrd=True)
dfTrain, dfTrainStd = fb.fit_transform(dfTrain)
dfTest, dfTestStd = fb.transform(dfTest)
dfTrain['ExternalRiskEstimate'].head()
```

operation	<=									>									==	!=
value	59.0	63.0	66.0	69.0	72.0	75.0	78.0	82.0	86.0	59.0	63.0	66.0	69.0	72.0	75.0	78.0	82.0	86.0	NaN	NaN
8960	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1
8403	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
1949	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
4886	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1
4998	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1

Run Boolean Rule Column Generation

```
# Instantiate BRCG with small complexity penalty and large beam search width
from aix360.algorithms.rbm import BooleanRuleCG
br = BooleanRuleCG(lambda0=1e-3, lambda1=1e-3, CNF=True)

# Train, print, and evaluate model
br.fit(dfTrain, yTrain)
from sklearn.metrics import accuracy_score
print('Training accuracy:', accuracy_score(yTrain, br.predict(dfTrain)))
print('Test accuracy:', accuracy_score(yTest, br.predict(dfTest)))
print('Predict Y=0 if ANY of the following rules are satisfied, otherwise Y=1:')
print(br.explain()['rules'])
```

```
Learning CNF rule with complexity parameters lambda0=0.001, lambda1=0.001
Initial LP solved
Iteration: 1, Objective: 0.2895
Iteration: 2, Objective: 0.2895
Iteration: 3, Objective: 0.2895
Iteration: 4, Objective: 0.2895
Iteration: 5, Objective: 0.2864
Iteration: 6, Objective: 0.2864
Iteration: 7, Objective: 0.2864
Training accuracy: 0.719573146021883
Test accuracy: 0.696515397082658
Predict Y=0 if ANY of the following rules are satisfied, otherwise Y=1:
['ExternalRiskEstimate <= 75.00 AND NumSatisfactoryTrades <= 17.00', 'ExternalRiskEstimate <= 72.00 AND NumSatisfactoryTrades > 17.00']
```


Loan Officer: Prototypical explanations for HELOC use case

Import statements

```
import pandas as pd
import numpy as np
import tensorflow as tf
from keras.models import Sequential, Model, load_model, model_from_json
from keras.layers import Dense
import matplotlib.pyplot as plt
from IPython.core.display import display, HTML

from aix360.algorithms.contrastive import CEMExplainer, KerasClassifier
from aix360.algorithms.protodash import ProtodashExplainer
from aix360.datasets.heloc_dataset import HELOCDataset
```

Load HELOC dataset and show sample applicants ¶

```
heloc = HELOCDataset()
df = heloc.dataframe()
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 24)
pd.set_option('display.width', 1000)
print("Size of HELOC dataset:", df.shape)
print("Number of \"Good\" applicants:", np.sum(df['RiskPerformance']=='Good'))
print("Number of \"Bad\" applicants:", np.sum(df['RiskPerformance']=='Bad'))
print("Sample Applicants:")
df.head(10).transpose()
```


1. **Process and Normalize HELOC dataset for training**

2. **Define and train a Neural Network classifier (loan approval model to be explained)**

3. **Obtain similar samples as explanations for a HELOC applicant predicted as "Good"**

```
idx = 8

X = xn_test[idx].reshape((1,) + xn_test[idx].shape)
print("Chosen Sample:", idx)
print("Prediction made by the model:", class_names[np.argmax(nn.predict_proba(X))])
print("Prediction probabilities:", nn.predict_proba(X))
print("")

# attach the prediction made by the model to X
X = np.hstack((X, nn.predict_classes(X).reshape((1,1))))

Xun = x_test[idx].reshape((1,) + x_test[idx].shape)
dfx = pd.DataFrame.from_records(Xun.astype('double')) # Create dataframe with original feature values
dfx[23] = class_names[X[0, -1]]
dfx.columns = df.columns
dfx.transpose()
```

```
Chosen Sample: 8
Prediction made by the model: Good
Prediction probabilities: [[-0.1889221   0.29527372]]
```


Find similar applicants predicted as "good" using the protodash explainer.

```
explainer = ProtodashExplainer()
(W, S, setValues) = explainer.explain(X, z_train_good, m=5) # Return weights W, Prototypes S and object
ive function values
```

```
     pcost       dcost       gap    pres   dres
 0:  0.0000e+00 -2.0000e+04  4e+00  1e+00  1e+00
 1:  1.8207e+01 -2.2985e+05  5e+01  1e+00  1e+00
 2: -1.6771e+00 -1.4132e+06  3e+02  1e+00  1e+00
 3:  6.4653e-01 -7.7669e+06  2e+03  1e+00  1e+00
 4:  9.0963e-01 -1.6930e+08  3e+04  1e+00  1e+00
 5:  6.8400e-01 -8.7461e+10  2e+07  1e+00  1e+00
 6:  2.1065e+08 -1.7700e+18  2e+18  6e-13  9e-03
 7:  2.1065e+08 -1.7700e+16  2e+16  6e-15  1e-03
 8:  2.1065e+08 -1.7700e+14  2e+14  4e-16  3e-05
 9:  2.1065e+08 -1.7706e+12  2e+12  2e-16  5e-07
10:  2.1059e+08 -1.8270e+10  2e+10  2e-16  6e-09

20:  7.2389e+00 -2.2354e+01  3e+01  2e-16  7e-16
21: -1.5947e+00 -5.4973e+00  4e+00  2e-16  6e-16
22: -2.2383e+00 -2.5578e+00  3e-01  2e-16  1e-16
23: -2.2526e+00 -2.2903e+00  4e-02  2e-16  7e-17
24: -2.2616e+00 -2.2685e+00  7e-03  3e-16  8e-17
25: -2.2622e+00 -2.2630e+00  8e-04  2e-16  1e-16
26: -2.2622e+00 -2.2622e+00  2e-05  2e-16  2e-16
27: -2.2622e+00 -2.2622e+00  2e-07  2e-16  2e-16
Optimal solution found.
```


Display similar users and give explanation as to why they are similar

```python
dfs = pd.DataFrame.from_records(zun_train_good[S, 0:-1].astype('double'))
RP=[]
for i in range(S.shape[0]):
    RP.append(class_names[z_train_good[S[i], -1]]) # Append class names
dfs[23] = RP
dfs.columns = df.columns
dfs["Weight"] = np.around(W, 5)/np.sum(np.around(W, 5)) # Calculate normalized importance weights
dfs.transpose()
```

	0	1	2	3	4
ExternalRiskEstimate	85	89	77	83	73
MSinceOldestTradeOpen	223	379	338	789	230
MSinceMostRecentTradeOpen	13	156	2	6	5
AverageMInFile	87	257	109	102	89
NumSatisfactoryTrades	23	3	16	41	61
NumInqLast6M	1	0	1	1	2
NumInqLast6Mexcl7days	1	0	1	0	2
NetFractionRevolvingBurden	4	0	2	1	59
NetFractionInstallBurden	0	0	0	0	72
NumRevolvingTradesWBalance	4	0	1	3	9
NumInstallTradesWBalance	1	0	1	0	1
NumBank2NatlTradesWHighUtilization	0	0	0	1	7
PercentTradesWBalance	50	0	22	23	53
RiskPerformance	Good	Good	Good	Good	Good
Weight	0.730222	0.0690562	0.0978593	0.0498047	0.0530578

Most prototypes have no debt

AIX360: IBM RESEARCH AI EXPLAINABILITY 360 TOOLKIT

Goals

- Support a community of users and contributors who will together help make models and their predictions more transparent.
- Support and advance research efforts in explainability.
- Contribute efforts to engender trust in AI.

IBM Research AIX360	
Explainability Algorithms	8 innovations to explain data and AI models
Repositories	github.ibm.com/AIX360 github.com/IBM/AIX360
Interactive Experience	aix360.mybluemix.net
API	aix360.readthedocs.io
Tutorials	13 notebooks (finance, healthcare, lifestyle, Attrition, etc.)
Developers	> 15 Researchers + Software engineers across YKT, India, Argentina

Trusted AI Toolkits

Adversarial Robustness 360 ✓	AI Fairness 360 ✓	AI Explainability 360 ✓	Causal Inference 360

Why Explainable AI Will Be the Next Big Disruptive Trend in Business AlleyWatch

Don't Trust Artificial Intelligence? Time To Open The AI 'Black Box'

CIO JOURNAL

Companies Grapple With AI's Opaque Decision-Making Process

THE WALL STREET JOURNAL.

Q&A

Thank you