Traditional methods for interpreting predictive models are not enough
Image Source: https://xkcd.com/1838/

What I cannot understand, I cannot build with confidence.

Is there a way to score our models on fairness, accountability, and transparency?

Strata

DATA CONFERENCE

DATASCIENCE.COM

Human in the Loop: Bayesian Rules Enabling Explainable AI

March 8, 2018

Head to Booth 1215 for a live demo of the DataScience.com Platform

About Me

Pramit Choudhary

@MaverickPramit

https://www.linkedin.com/in/pramitc/

https://github.com/pramitchoudhary

DATASCIENCE.COM
SKATER

PyData

I am a lead data scientist at DataScience.com. I enjoy applying and optimizing classical machine learning algorithms, NLP, and Bayesian design strategy to solve real-world problems. Currently, I am exploring on better ways to extract, evaluate, and explain the learned decision policies of models. Before joining DataScience.com, I used machine learning algorithms to find love for eHarmony customers. I am one of the principal authors of Skater, a model interpretation package for Python. I also organize the PyData Socal meet-up.

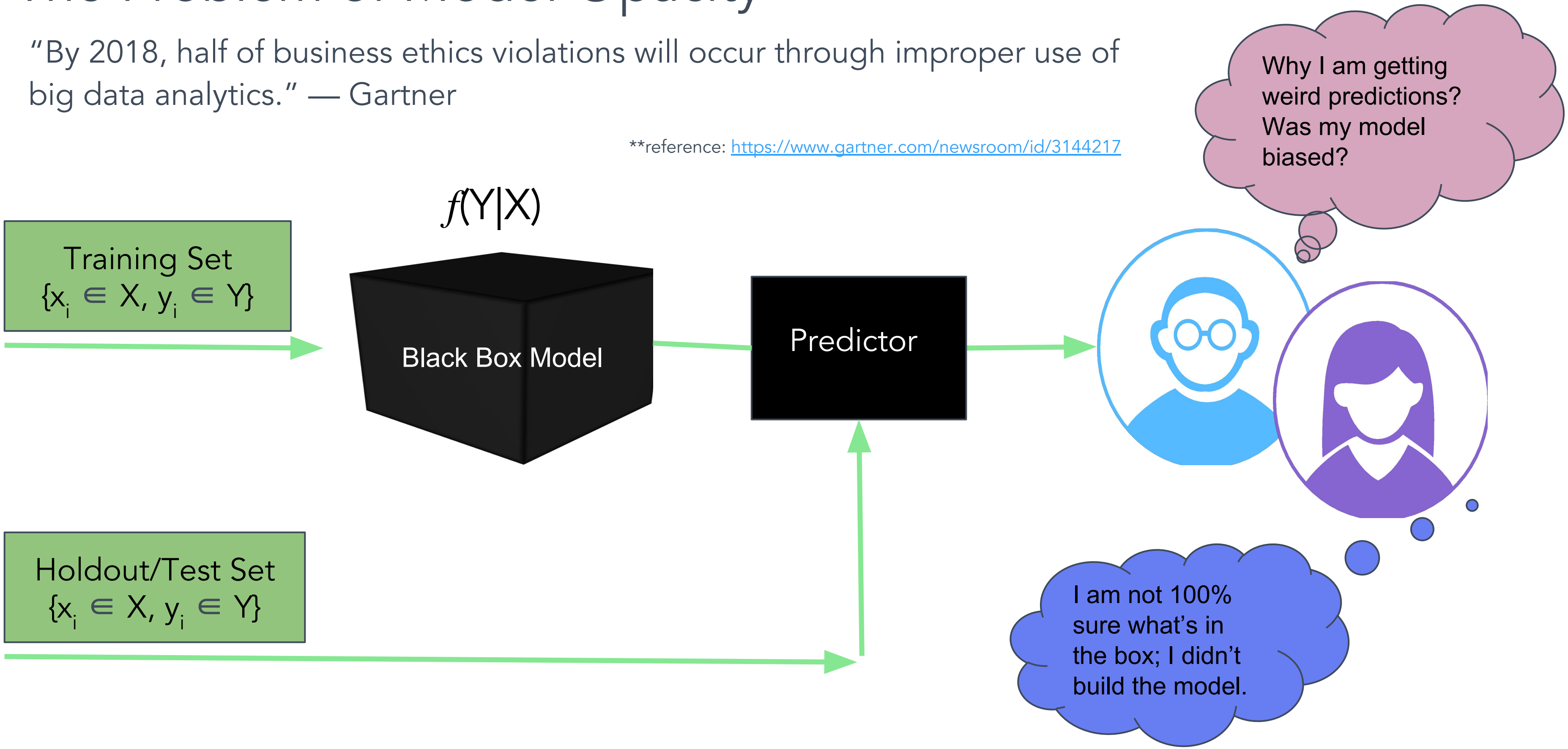
Agenda

- Understand the problem of model opacity
- Define the “what” and “why” of model interpretation
- Define the scope of model interpretation
- How do we enable interpretability?
- What is the Bayesian rule list?
- Understand the tension between interpretability and performance
- Benchmark numbers
- What is Skater and how does it help you build models the right way?
- References

The Problem of Model Opacity

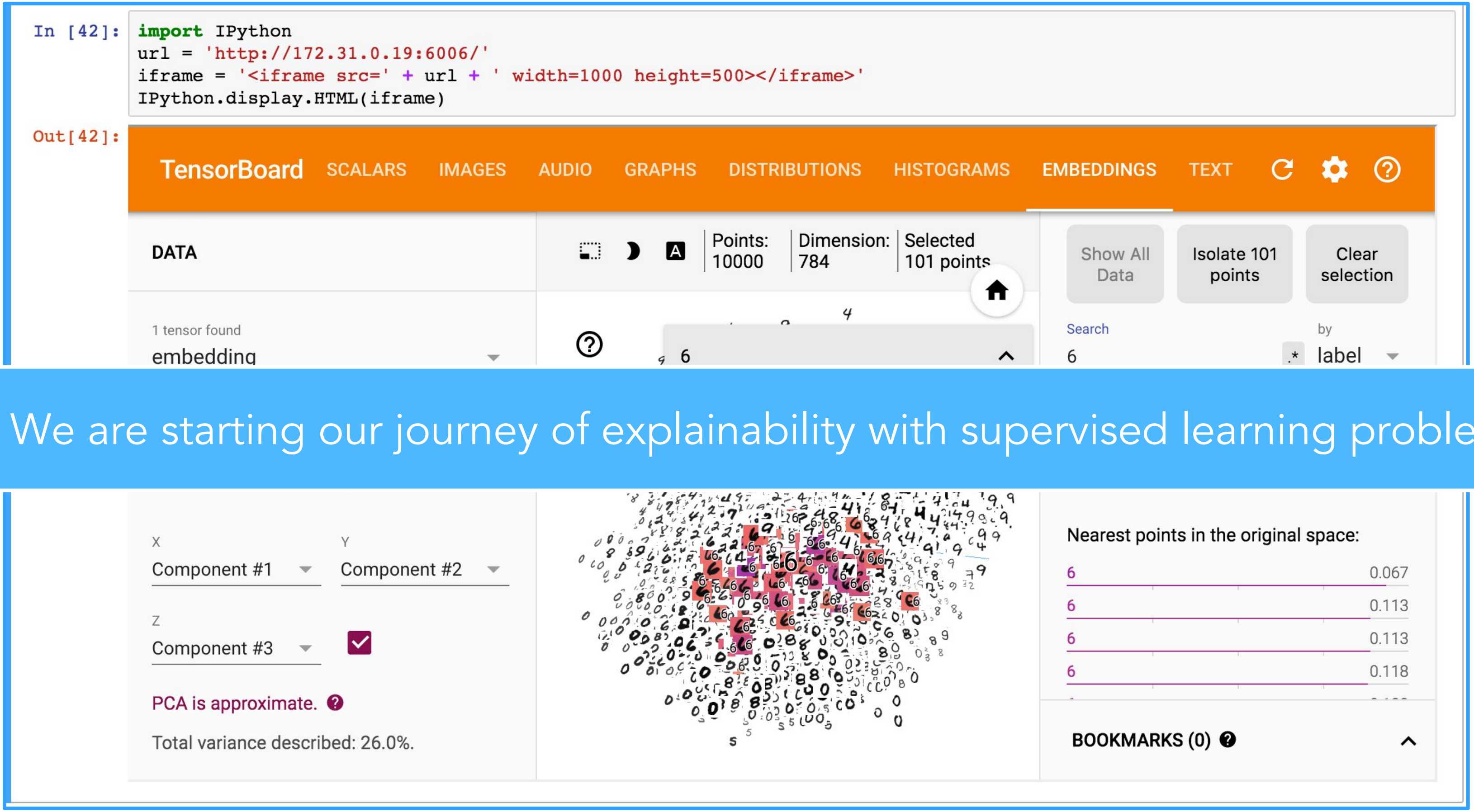
"By 2018, half of business ethics violations will occur through improper use of big data analytics." — Gartner

**reference: https://www.gartner.com/newsroom/id/3144217



What is Model Interpretation?

- An extension of model evaluation that helps to foster a better understanding of a model’s learned decision policies.
- Ability to explain and present a model in a way that is human understandable.
- Human understandable: The model’s result is self descriptive & needs no further explanation.



We are starting our journey of explainability with supervised learning problems.

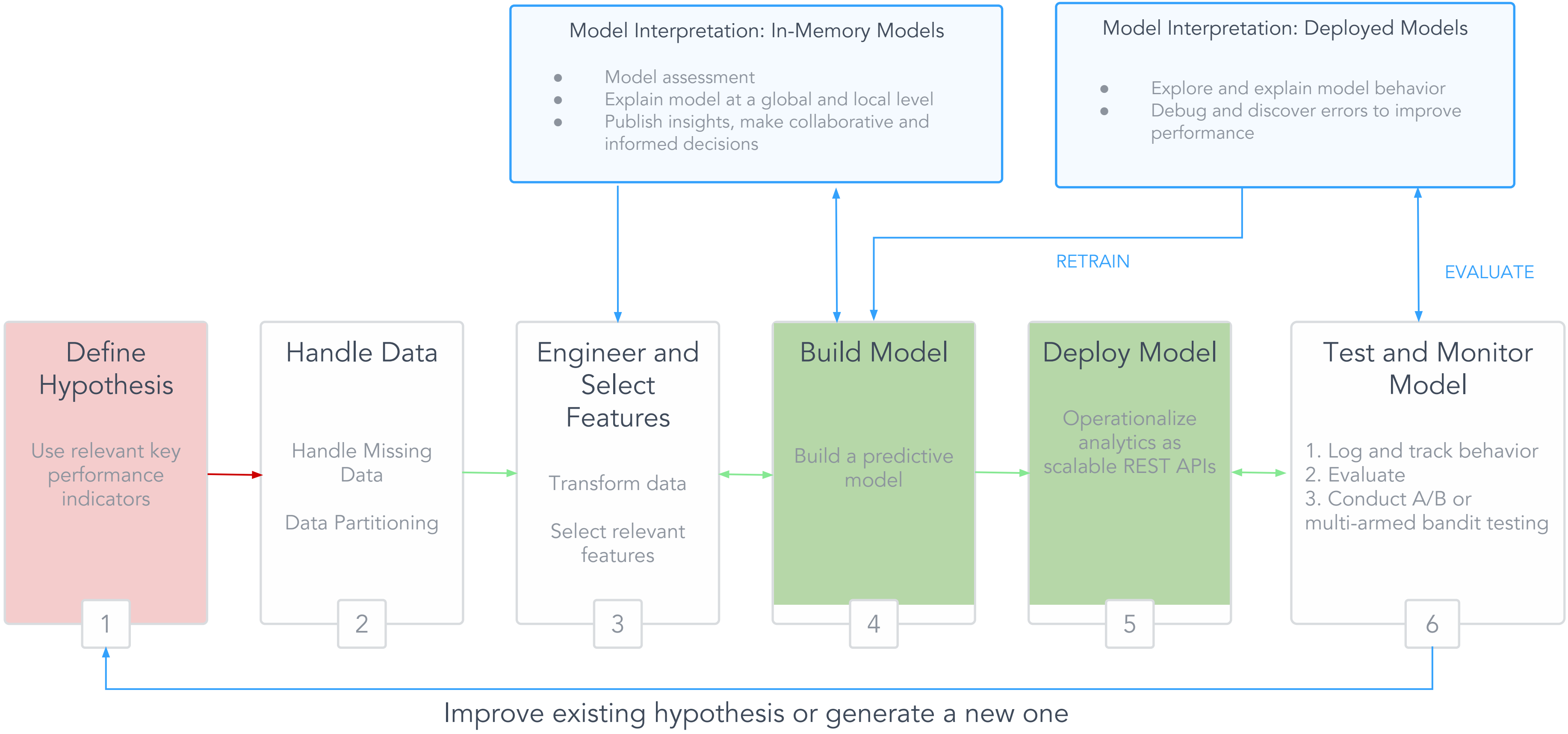
Image source: constructed using tensorboard

What Do We Want to Achieve?

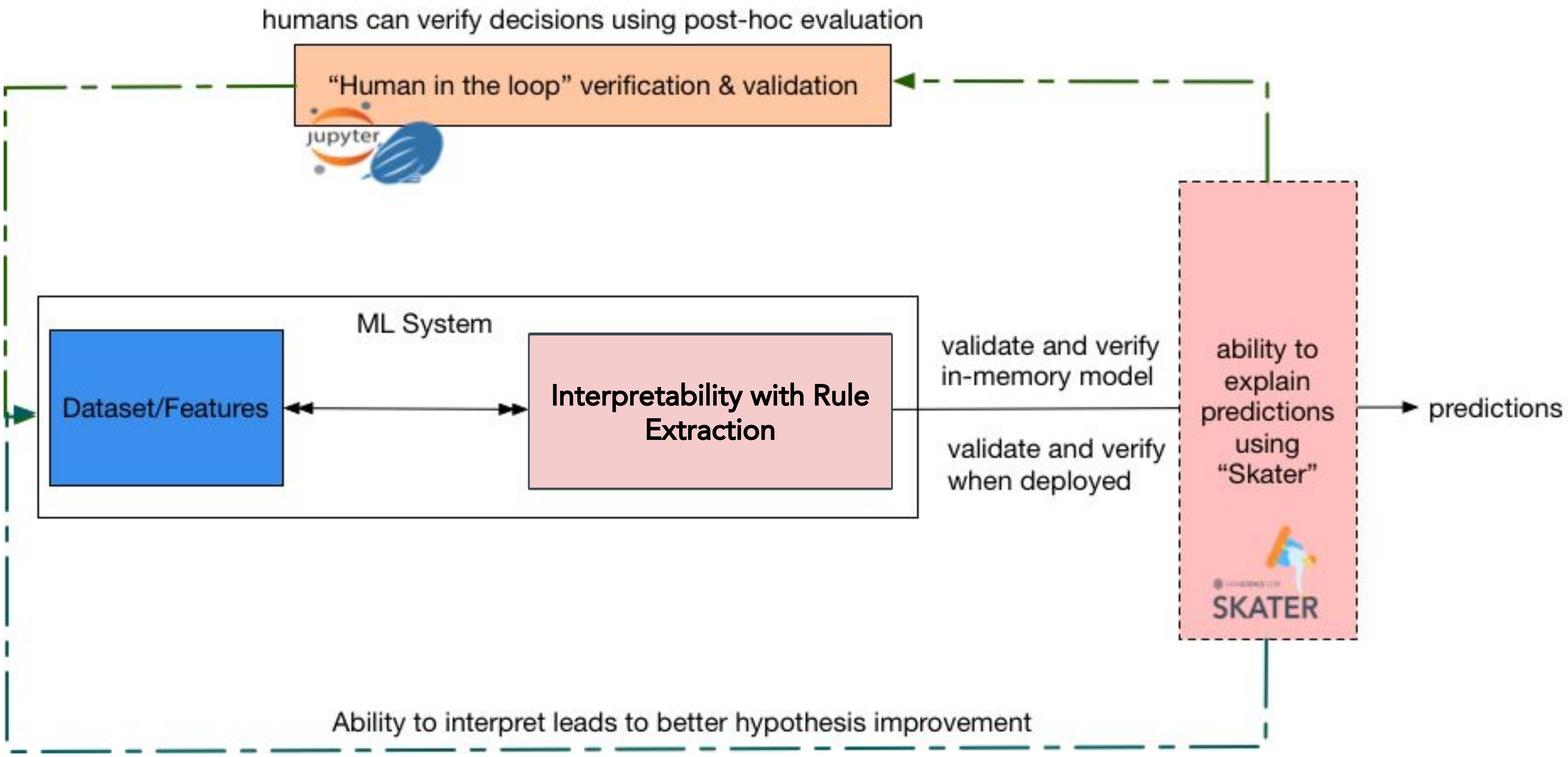
With model interpretation, we want to answer the following questions:

- **Why** did the model behave in a certain way?
- **What** was the reason for false positives? What are the relevant variables driving a model's outcome, e.g., customer lifetime value, fraud detection, image classification, spam detection?
- **How** can we trust the predictions of a "black box" model? Is the predictive model biased?

Machine Learning Workflow



An Interpretable Machine Learning System



Why is Model Interpretation Important?

"Explain the model."

Producer:

- Data scientist/analyst building a model
- Consultants helping clients

Consumer/Decision Maker:

- Business owners or data engineers
- Risk/security assessment managers
- Humans being affected by the model

Ideas collapse.

Image source: Edu Lauton on Unsplash

Motives for Model Interpretation

Producer

- Data Scientist /
- Machine Learning Engineer
- Data Analyst
- Statistician

1. **Debugging and improving** an ML system
2. **Exploring and discovering latent or hidden feature interactions** (useful for feature engineering/selection and resolving preconceptions)
3. Understanding **model variability**
4. Helps in **model comparison**
5. Building **domain knowledge** about a particular use case
6. Brings **transparency** to decision making to enable **trust**

Consumer

- Data Science Manager
- Business owner
- Data Engineer
- Auditors / Risk Managers

1. Explain **the model/algorithm**
2. Explain **the key features driving the KPI**
3. **Verify and validate the accountability** of ML learning systems, e.g. causes for False positives in credit scoring, insurance claim frauds
4. Identify **blind spots to** prevent adversarial attacks or fixing dataset errors
5. **Ability to share** the explanations to consumers of the predictive model?
6. Comply with **Data Protection Regulations**, e.g. EU's GDPR

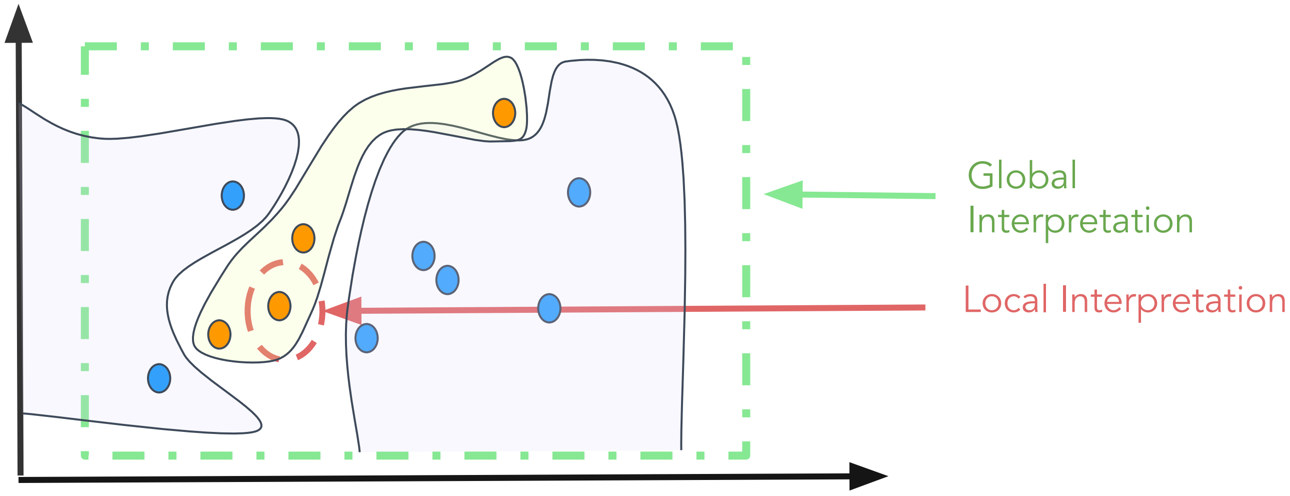
Scope Of Interpretation

Global Interpretation

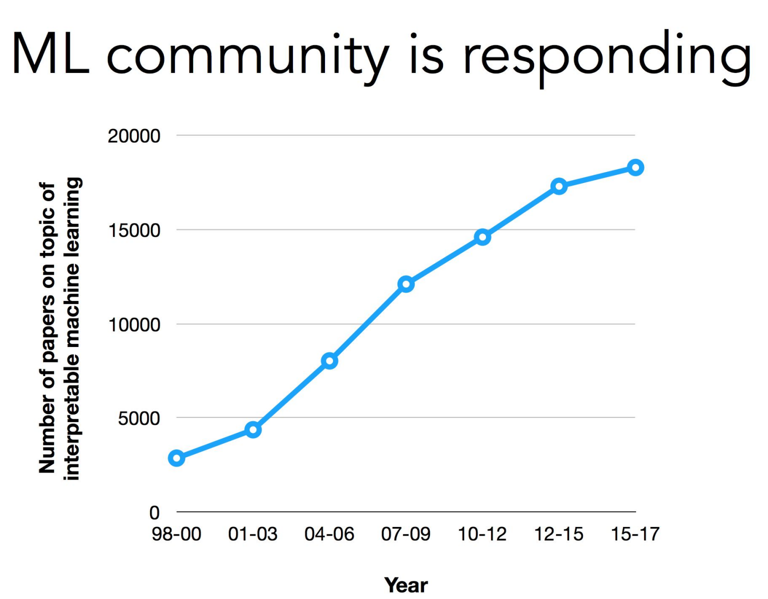
Being able to explain the conditional interaction between dependent(*response*) variables and independent(*predictor, or explanatory*) variables based on the complete dataset

Local Interpretation

Being able to explain the conditional interaction between dependent(*response*) variables and independent(*predictor, or explanatory*) variables with respect to a single prediction



How Do We Enable Model Interpretation?

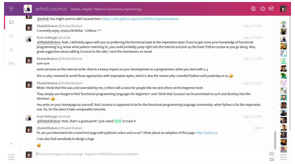


Reference: Been Kim(ICML'17) Google Brain
(http://people.csail.mit.edu/beenkim/papers/BeenK_FinaleDV_ICML2017_tutorial.pdf)

Introducing Skater

GitHub https://github.com/datascienceinc/Skater

If you like the idea, give us a star!



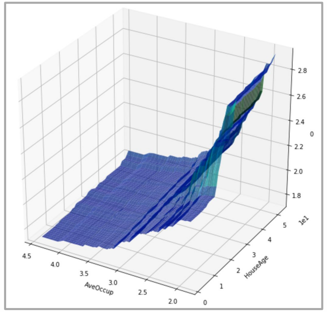
Gitter Channel (join us here):
https://gitter.im/datascienceinc-skater/Lobby

DATASCIENCE.COM

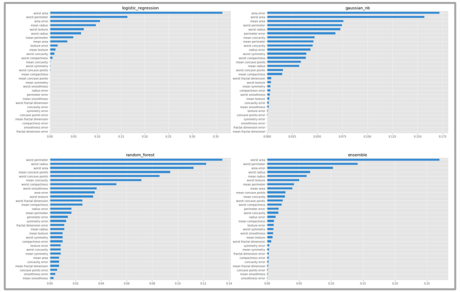
SKATER

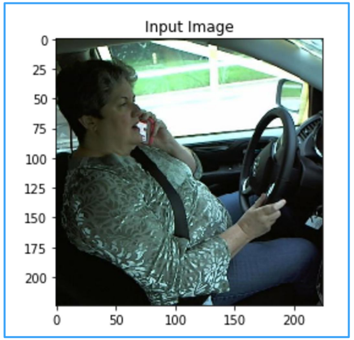
1. Post-Hoc Evaluation of Models

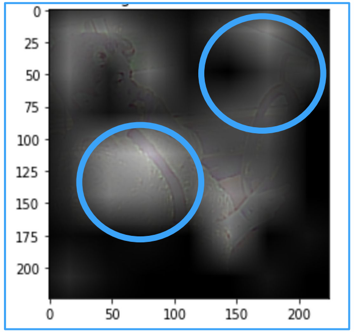
How Do We Enable Interpretation?



- **Post-hoc evaluation**: A black-box model is built, and we need a way to interpret it.
  - Model agnostic partial dependence plot
    G. Hooker(KDD'04). Discovering additive structure in black box functions
  - Model agnostic feature importance
  - Local interpretable model agnostic explanation (LIME)
    Marco Tulio Ribeiro et. al(2016). Nothing Else Matters
  - Saliency mask for DNN (image/text): Not supported yet; coming soon…
    Ning Xie et. al(NIPS' 2017). Relating Input Concepts to Convolutional Neural Network Decisions



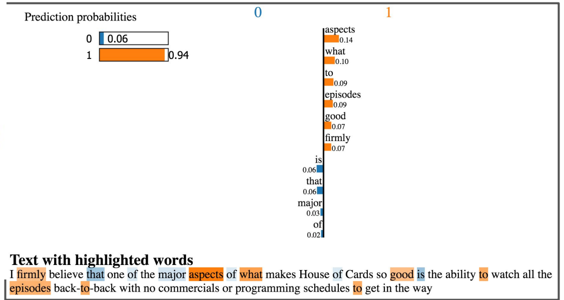




GM, at least, is heading in that direction. One of the post-sale questions they asked me was if I'd like the choice of a cigarette liter or an accessory plug, and another whether I'd like the choice of an ashtray or a cup holder.

The '93 Geo Storms have the cigarette lighter vs accessory plug option (which did not exist in the '92 I bought) -- I'm not sure about the ash tray vs cup holder. It's a step in the right direction.

The ashtray does make a convenient change-holder so it's not completely useless.



2. Bayesian Rule List: Building Naturally Interpretable Models Via Rule Extraction

Demo

Building a Model Using a Bayesian Rule List and Skater

1. https://github.com/datascienceinc/Skater/blob/master/examples/rule_list_notebooks/rule_lists_continuous_features.ipynb

2. https://github.com/datascienceinc/Skater/blob/master/examples/rule_list_notebooks/rule_lists_titanic_dataset.ipynb

3. https://github.com/datascienceinc/Skater/blob/master/examples/credit_analysis/credit_analysis_rule_lists.ipynb

How Do We Enable Interpretation?

- Using a probabilistic interpretable estimator (bayesian rule list):
  a. Generative probabilistic classifier P(y = 1| x) for each x
  b. Initially designed by Letham, Rudin, McCormick, Madigan (2015)
  c. Improved by Hongyu Yang. et. al. as Scalable Bayesian Rule List (2017)
  d. Works great for Tabular datasets with discrete and independent meaninful features
  e. Competitor to decision trees; greedy splitting and pruning
  f. Built using pre-mined association rules (frequent pattern-matching algorithms)
     - ECLAT (*Equivalence Class Clustering and Bottom up Lattice Traversal*)
     - Non-frequent patterns are not considered
  g. Build a bayesian hierarchical model over frequently occuring pre-mined rule lists
  h. Applies MCMC (Metropolis–Hastings algorithm) to sample from posterior distribution over permutation of "IF-THEN-ELSE" conditional statement
  i. **Output:** Generates a logical structure of human-interpretable IF then ELSE decision stumps
  j. **Scope of interpretation:** global and local

Bayesian Rule List

- Consider independent and identically distributed(i.i.d) training examples of the form {X, Y} -> $\{(x_i, y_i)\}^{n}_{i=1}$ where $x_i \in X$ as encoded features and $y_i \in Y$ as binary labels [0s or 1s].
- A typical bayesian rule list estimator would look like this:

If 'x' obeys 'rule:[1]' then p(y=1 | x) = …
else if 'x' obeys 'rule:[2]' then p(y=1 | x) = …
…
else if 'x' obeys 'rule:[n+1]' then p(y=1 | x) = …

…

else if 'x' obeys 'rule[m]' then p(y=1 | x) = …
else default rule p(y=1 | x)

Each rule is independent and selected from a **set of pre-mined rules** using frequent matching algorithms, e.g., ECLAT.

Goal: Optimize over the **possible set of pre-mined rules** and their **order** to create the final set of interpretable decision stumps.

Example: Rule List Representation

Optimize cardinality of rules horizontally and vertically

If { 2-Hour_serum_insulin_(mu_U/ml)=(192.75, 384.5] and Diabetes_pedigree_function=(576.25, 768.0] } then positive probability = 0.23%

else if { Glucose_concentration_test=(0.999, 192.75] } then positive probability = 0.94%

else if { Body_mass_index=(0.999, 192.75] } then positive probability = 0.83%

else if { Glucose_concentration_test=(576.25, 768.0] } then positive probability = 0.28%

else (default rule) then positive probability = 0.62%

Goal: Optimize on **finite** number of rules maintaining **accuracy.**

Sampling: Rules are sampled from posterior distribution over a permutation of pre-mined rules.

Scope of Interpretation: Global and local.

Figure: BRL output on common diabetes dataset (http://scikit-learn.org/stable/datasets/index.html#diabetes-dataset)

Generative vs. Discriminative Models

Input Examples
$\{x_i \in X, y_i \in Y\}$

Learns p(Y | X) directly

Discriminative Model:
Models the posterior probability directly; maps input X to output and labels Y directly; e.g., SVM, NN.

Learns p(Y | X) in-directly

$p(Y \mid X) \propto p(X \mid Y) * p(Y)$

Generative Model:
Models a joint probability of input X and output Y p(X, Y); computes prediction P(Y | X) using Bayes' rule; e.g., Naive Bayes, GAN (Generative Adversarial Network), BRL.

** Reference: Ng and Jordan(2001) [On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes]

Optimization Goals for Bayesian Rule List

Sample from a posterior distribution over a permutation of pre-mined "IF-THEN-ELSE" conditional statement:

$$p(d|X, Y, \mathcal{A}, \alpha, \lambda, \eta) \propto p(Y|X, d, \alpha) * p(d|\mathcal{A}, \lambda, \eta)$$

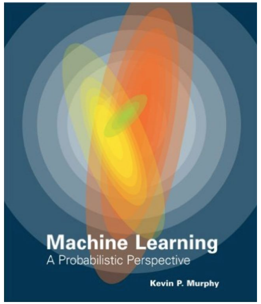
Posterior: Conditional probability of an event based on relevant evidence

$\propto$ **Likelihood:** Probability of an event that has already occurred (binomial distribution).

✖ **Prior Probability:** Probability of one's belief before evidence (beta distribution).

where,

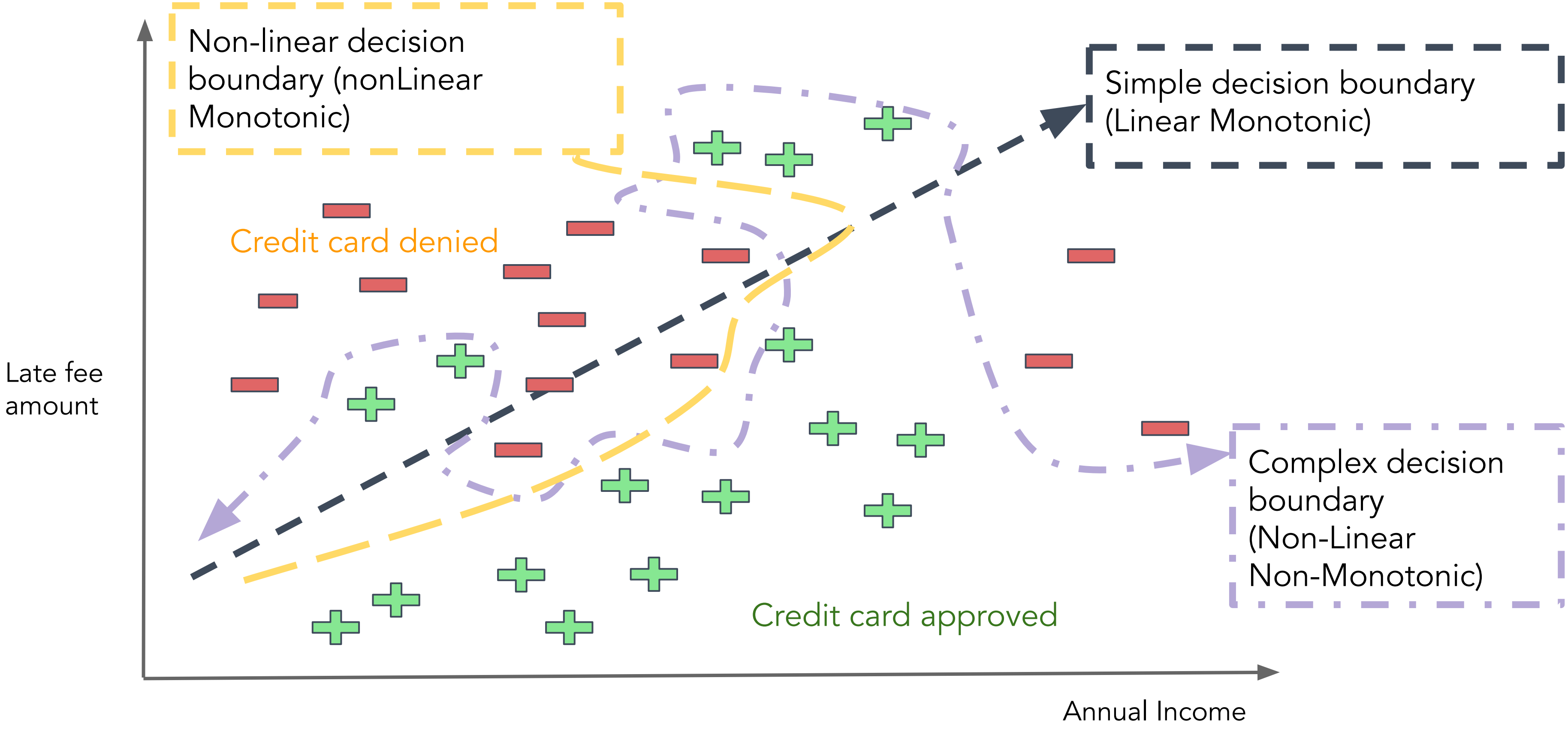
- d = Ordered subset of rules
- $\mathcal{A}$: Pre-mined collection of all rules using the frequent pattern matching algorithm
- Prior hyper-parameters: α, λ, η
  - $\alpha = [\alpha 0, \alpha 1]$: Prior parameter for each label in a binary classification problem
  - λ: Hyper-parameter for the expected length of the rule list
  - η: Hyper-parameter for the expected cardinality of each rule in the optimal rule list



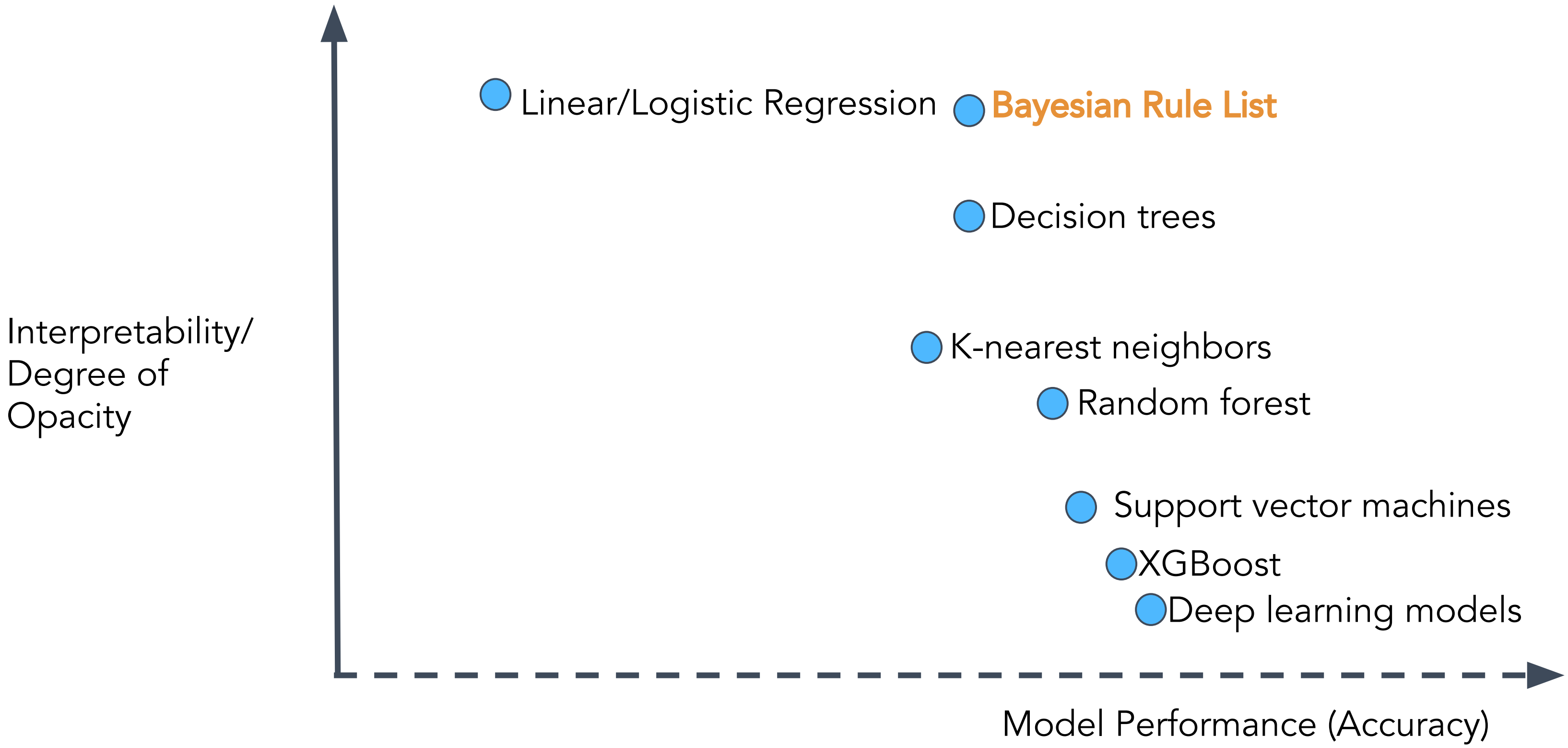
See Chapter Three of Machine Learning: A Probabilistic Perspective

Tension Between Interpretability and Model Performance

Performance vs. Interpretability



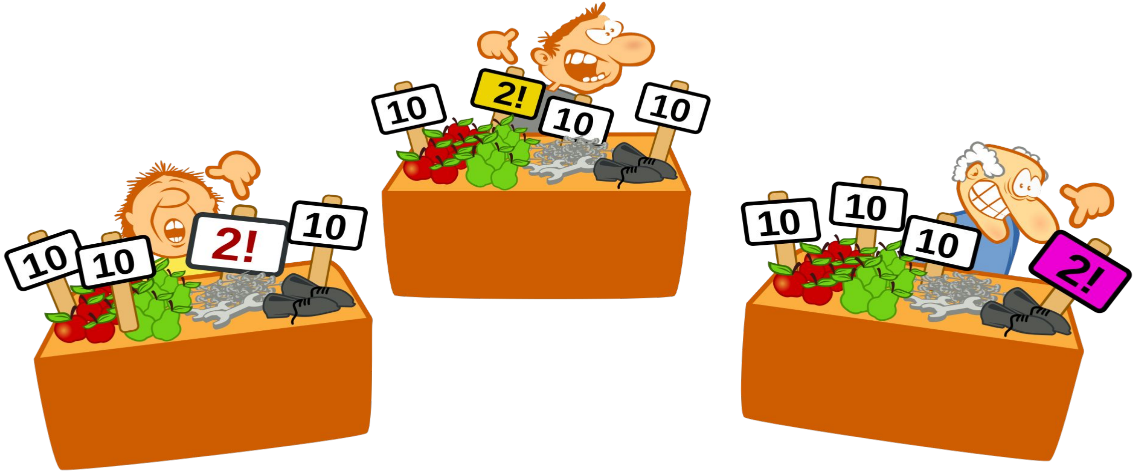
Tension Between Interpretability and Model Performance



** Remember: The purpose of the chart is not to mirror any benchmark on model performance, but to articulate the opacity of predictive models

No Free Lunch Theorem

"Any elevated performance over one class of problems is offset by performance over another class." — David H. Wolpert and William G. Macready, (1997), https://ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf



Simplicity: 10, Robustness: 10, Computation Speed: scope for improvement, Interpretability: 10

Simplicity:10, Robustness:10, Scalability: with smart optimization, Interpretability: 10

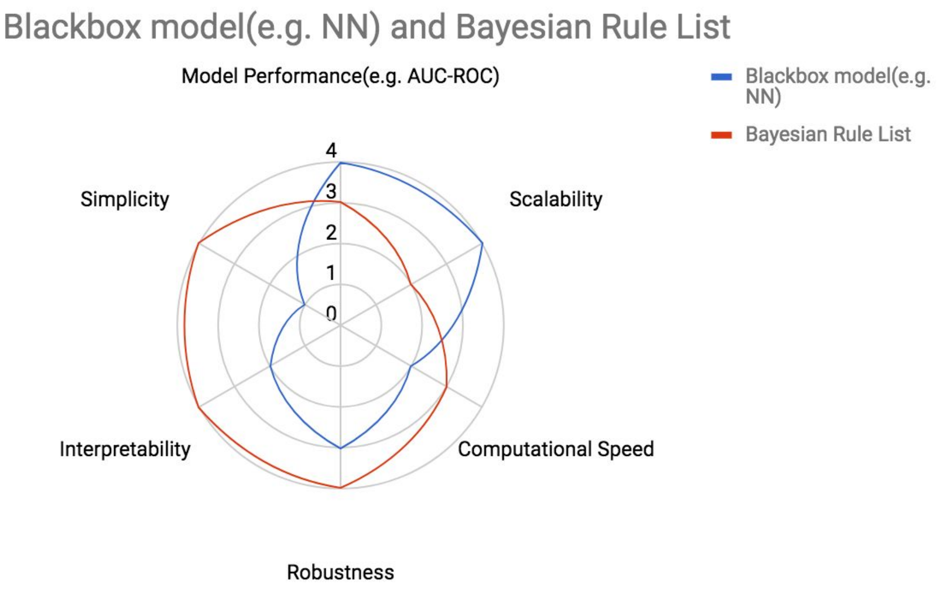
Image source:wiki(Mimooh, https://commons.wikimedia.org/wiki/File:No_free_lunch_theorem.svg

Simplicity Is Key

- **Occam's Razor Principle:** "When presented with competing hypothetical answers to a problem, one should select the one that makes the fewest assumptions."
- **In computational learning, b**uild models with the objective of producing a succinct representation of the training set.

Model Selection Policies:

- Model Performance (e.g., AUC-ROC): How accurate is the model?
- Scalability: Can the model handle huge volume of data?
- Computational Speed: Does the model take a long time to build?
- Robustness: Are the predicted result stable over a period of time?
- Interpretability: Can one interpret the output in a human understandable way?
- Simplicity: Can one explain the model easily?



What If We Achieve Accuracy?

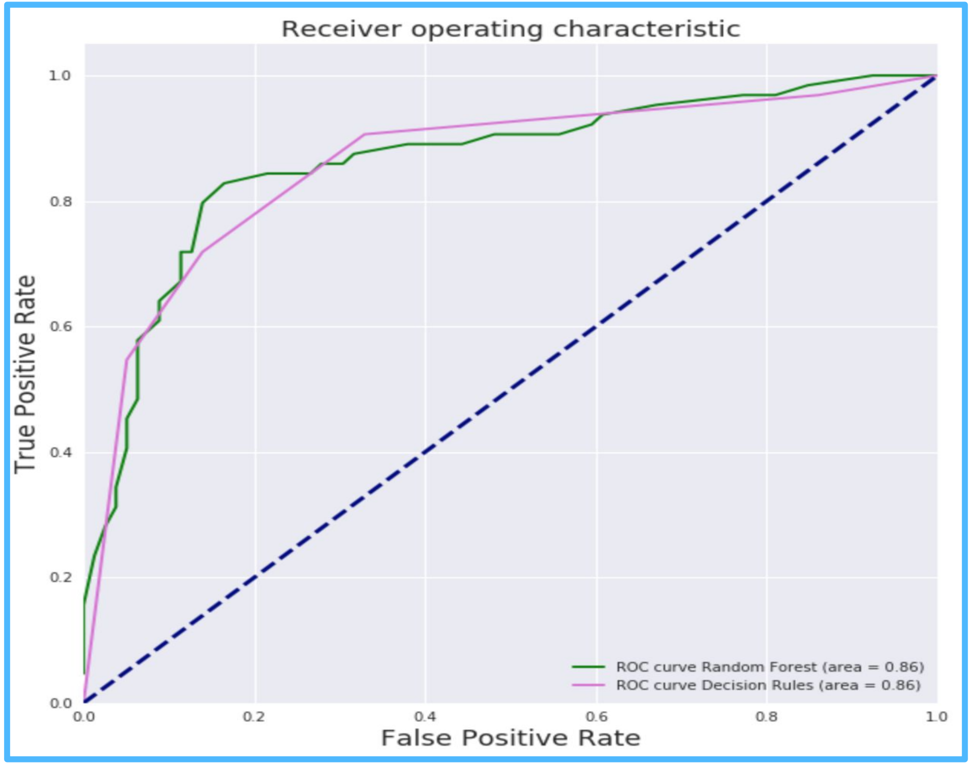


Figure: Comparison of BRL and RF using AUC of ROC on Titanic dataset

Performance Benchmark Using BRL

Dataset	Data Type	Problem Type	Model Type	Train Accuracy	Test Accuracy	Train AUC-ROC	Test AUC-ROC	Computation Time (in sec)
Diabetes dataset (*Train: 576 rows; Test: 192*)	Tabular data: continuous features	Supervised Classification				0.82	0.76	0.74
Diabetes dataset (*Train: 576 rows; Test: 192*)	Tabular data: continuous features	Supervised Classification	RF	1.0	0.75	0.81	0.80	0.14
Titanic dataset (*Train: 571 rows; Test: 143 rows*)	Tabular data: categorical & continuous	Supervised Classification	BRLC	0.80	0.86	0.84	0.86	0.67
Titanic dataset (*Train: 571 rows; Test: 143 rows*)	Tabular data: categorical & continuous	Supervised Classification	RF	1.0	0.81	1.0	0.86	0.07
Credit analysis (Train: 29,839 rows; Test: 9,947 rows)	Tabular data: categorical & continuous	S C			0.86	0.65	0.65	2.81
Credit analysis (Train: 29,839 rows; Test: 9,947 rows)	Tabular data: categorical & continuous	Supervised Classification	Linear SVM	0.85	0.86	0.68	0.70	0.15

Could be improved with more thoughtful feature engineering and selection

0.05 difference in performance on hold out using 10% of the data compared to SVM

Skater: BRL API Overview (BRLC)

```
from skater.core.global_interpretation.interpretable_models.brlc import BRLC
...
from sklearn.datasets.mldata import fetch_mldata
input_df = fetch_mldata("diabetes")
...
Xtrain, Xtest, ytrain, ytest = train_test_split(input_df, y, test_size=0.20, random_state=0)

sbrl_model = BRLC(min_rule_len=1, max_rule_len=10, iterations=10000, n_chains=20, drop_features=True)

# Train a model, by default discretizer is enabled. So, you wish to exclude features then exclude them using
# the undiscretize_feature_list parameter
model = sbrl_model.fit(Xtrain, ytrain, bin_labels="default")

#print the learned model
sbrl_inst.print_model()

# Discretize continuous features
features_to_descritize = Xtrain.columns
Xtrain_filtered = sbrl_model.discretizer(Xtrain, features_to_descritize, labels_for_bin="default")

# Genertate probability scores for the likelihood class
predict_scores = sbrl_model.predict_proba(Xtest)

# Generate final prediction
_, y_hat  = sbrl_model.predict(Xtest)

# Persist and reload the model if needed
sbrl_model.save_model("model.pkl")
sbrl_model.load_model("model.pkl")

# Get access to all the learned rules
sbrl_model.access_learned_rules("all")
```

Import the BRLC class

Instantiate BRLC instance

Train a model using fit

Display learned "if-else" conditions

Use discretizer for continuous features

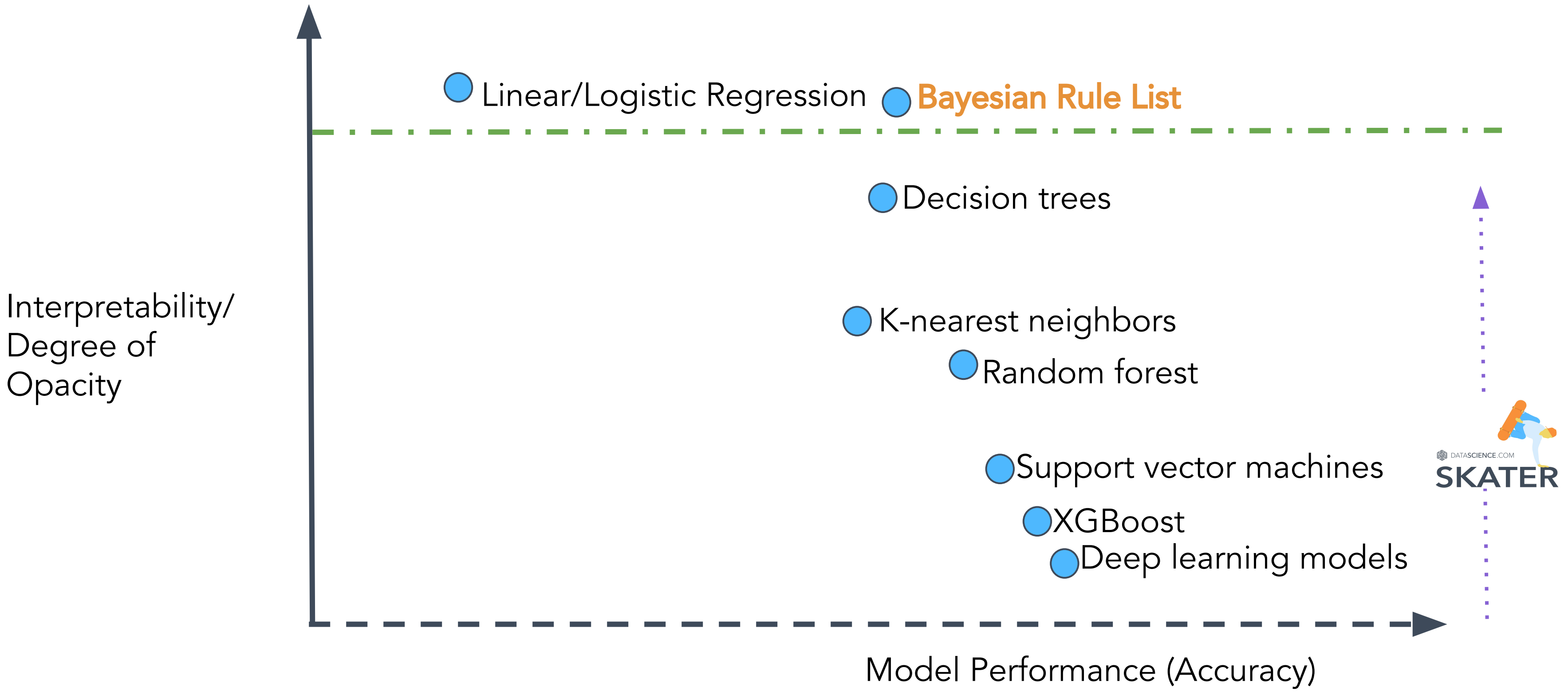
Generate class probabilities

Predict class labels

Persist model

Access other rules

Mission Statement: Enable Interpretability for All Models

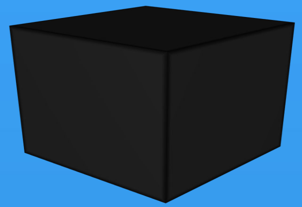


** Remember: The purpose of the chart is not to mirror any benchmark on model performance, but to articulate the opacity of predictive models

WITHOUT INTERPRETATION ...

$f(Y|X)$

Data



Evaluate

How do I understand my models?

Black box model

WITH SKATER ...

Data

Evaluate

- Bayesian rule list (BRL)
- Partial dependence plot
- Relative variable importance
- Local Interpretable Model Explanation (LIME)
- More coming ...

Unboxed model

R or Python model (linear, nonlinear, ensemble, neural networks)

Scikit-learn, caret and rpart packages for CRAN

H20.ai, Algorithmia, etc.

Future Work and Improvement

- Other rule-based algorithm approaches being considered for implementation:
  - H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction
  - Issue: https://github.com/datascienceinc/Skater/issues/207

If Respiratory-Illness$=$Yes **and** Smoker$=$Yes **and** Age$\geq$ 50 **then** Lung Cancer

If Risk-LungCancer$=$Yes **and** Blood-Pressure$\geq$ 0.3 **then** Lung Cancer

If Risk-Depression$=$Yes **and** Past-Depression$=$Yes **then** Depression

If BMI$\geq$ 0.3 **and** Insurance$=$None **and** Blood-Pressure$\geq$ 0.2 **then** Depression

If Smoker$=$Yes **and** BMI$\geq$ 0.2 **and** Age$\geq$ 60 **then** Diabetes

If Risk-Diabetes$=$Yes **and** BMI$\geq$ 0.4 **and** Prob-Infections$\geq$ 0.2 **then** Diabetes

If Doctor-Visits $\geq$ 0.4 **and** Childhood-Obesity$=$Yes **then** Diabetes

Future Work and Improvement (continued)

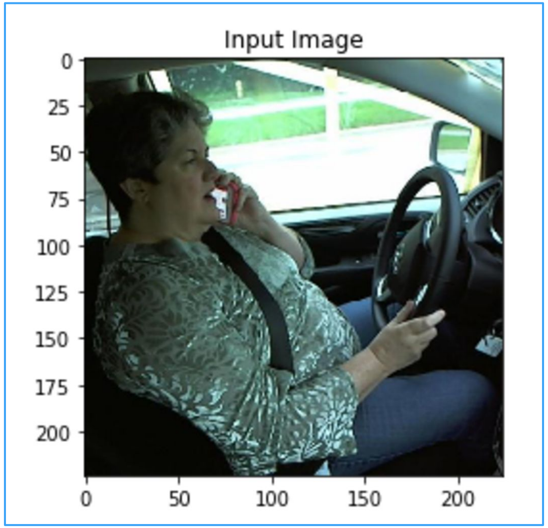
- Improve handling of continuous feature
  - Discretize using entropy criterion with the Minimum Description Length Principle (MDLP) (Reference: Irani, Keki B'93. "Multi-interval discretization of continuous-valued attributes for classification learning.")
  - Issue: https://github.com/datascienceinc/Skater/issues/206

- Improve scalability and computational efficiency for BRL
  - Parallelizing MCMC sampling using Weierstrass Sampler
  - Reference: Parallelizing MCMC via Weierstrass Sampler, https://arxiv.org/abs/1312.4605

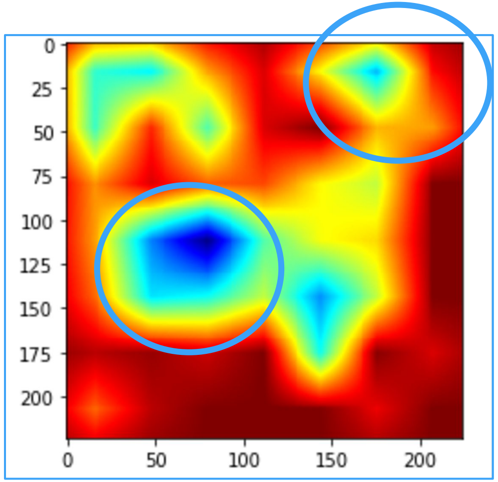
- Add more example notebooks, applied to different use-cases
  - Handling text based models - Kaggle sms-spam-collection dataset
  - More benchmarks

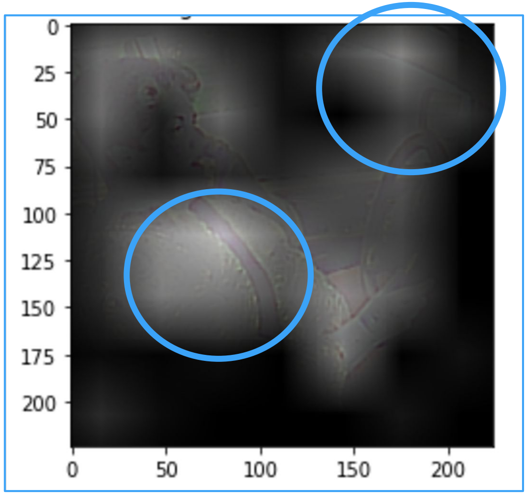
A Quick Glimpse Into The Future

Visual Q&A: Is the person driving the car safely?



Top 5 Predictions:
1. seat belt = 0.75
2. limousine = 0.051
3. golf cart = 0.017
4. minivan = 0.015
5. car mirror = 0.015





Professor. Sameer Singh,

Assistant Professor of Computer Science @ the University of California, Irvine

https://www.datascience.com/resources/webinars/interpreting-machine-learning-models

Paco Nathan,

Director of Learning Group @ O'Reilly Media

Q&A

info@datascience.com

pramit@datascience.com

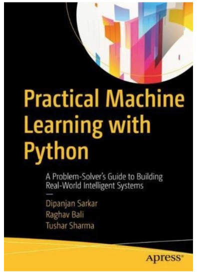
@DataScienceInc

@MaverickPramit

Help wanted: https://github.com/datascienceinc/Skater/labels/help%20wanted

References

- Interpretation references:
  - A. Weller, (ICML 2017).Challenges for Transparency
  - Zachary C. Lipton, (2016). The Mythos of Model Interpretability
- Rule list-related literature:
  - Letham, B., Rudin, C., McCormick, T. H., & Madigan, D. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. Annals of Applied Statistics, 9(3), 1350–1371
  - Yang, H., Rudin, C., Seltzer M. (2016). Scalable Bayesian Rule Lists
- Detailed examples of model interpretation using Skater



- Marco Tulio Ribeiro, et al. (KDD 2016) "Why Should I Trust You?": Explaining the Predictions of Any Classifier