Winning Queen City Hackathon

Case Study

By Ibrahim Sadikov

Powered By: ABDSOCIETY www.abdsociety.org QUEEN CITY HACKATHON RED VENTURES

About Me

- I am Ibrahim go by Abe
- Data Scientist at Genpact
- Graduate student at UNCC School of Data Science
- Undergraduate degree in Economics with Finance
- Hometown **Samarkand**
- Love Kaggling and being engaged in Charlotte's Data Science community

github.com/Ibrokhimsadikov/ABD_Society

Agenda

01 ABOUT COMPETITION

02 OBJECTIVES & WORKFLOW

03 OUTCOMES AND REMARKS

04 PLATFORM MAYA

05 H2O, LIME, SHINY

06 DEMO

About Competition

Economic and Social Mobility

The Defining Issue of Our Time

- 25% of Charlotte's children experience poverty.
- 44% of Mecklenburg County children ages 0-5 live in households earning below 200% of the Federal poverty level.
- The poverty rate of working age African Americans and Hispanics is twice that for whites in Charlotte.
- 14.5% of young people ages 16-24 are neither working nor going to school or training.

Hackathon: Open-Ended Track-Problem statement

- A 2014 study ranked **Charlotte 50th out of 50** in the chance that a child born into a low-income family could move into the top 20 percent of income earners by adulthood

- **Leading on Opportunity** published *21 Community Strategies for Economic Mobility* addressing ways that Charlotte can help improve opportunities for its residents

- **Create a tool,** policy, **predictive model**, or other software to help address Charlotte's low levels of upward mobility.

Competition workflow

9 pm
- Kick off
- Choosing Track
- Research

12 am
- Literature Review
- How define EM

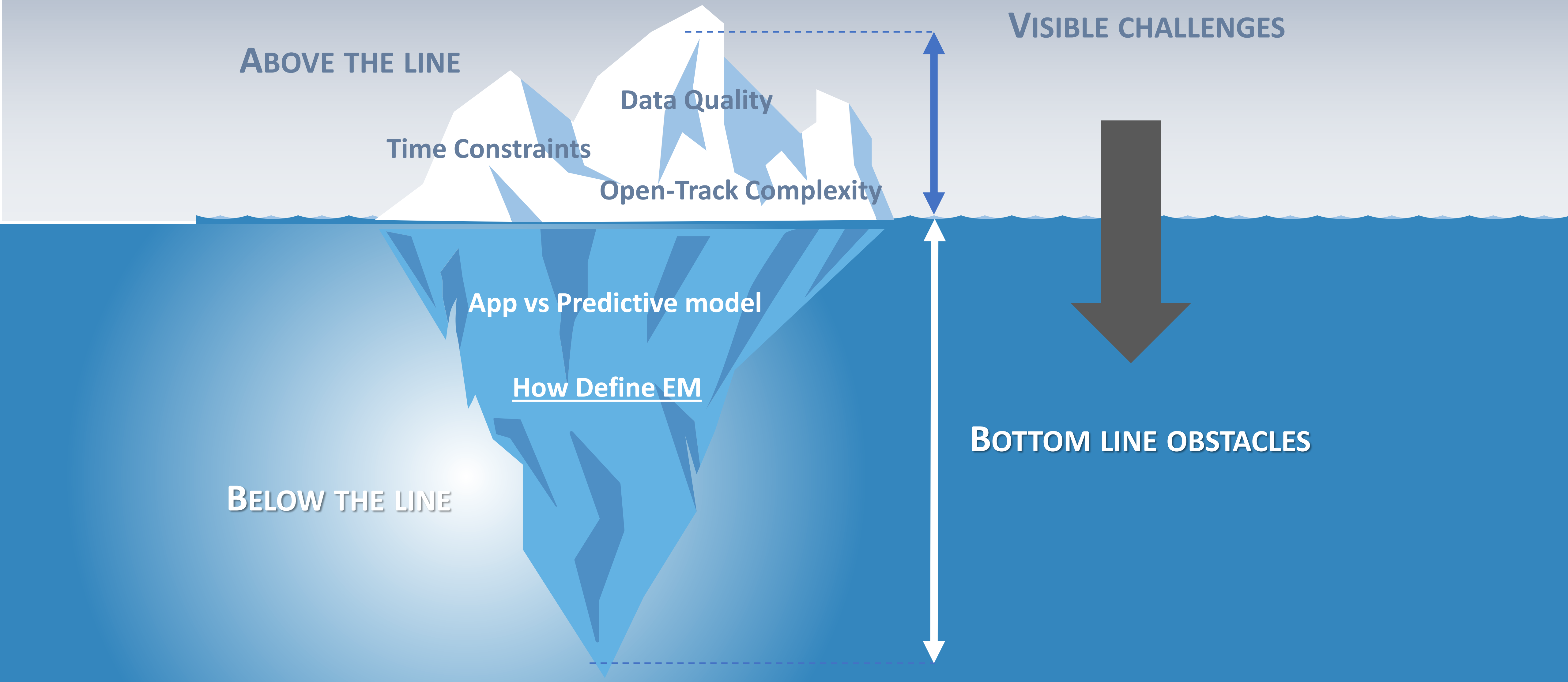
2 am
- Data Exploration
- External Sources
- Building App

4 am
- Debugging
- Model build H2O

8 am
- Debugging
- Lime part & EDA

9:30 am
- Deployment
- Presentation

The Iceberg Analogy Of Competition Challenges



Objectives

Long-term & Tangible

My purpose was to come up with some solution that could have more long term affect and can be implementable. Create End-End to data science platform to support future decision making

Prioritize strategies

To sort significant determinants of EM from proposed 21 strategic factors so that county can focus its constraint budget for the most significant influential factors based on ML predicted estimates

Define EM

How can we elucidate EM as a numeric value so that we can input it as label to predictive models. To define mathematically how Economic mobility can be measured

What did work, what I missed

Despite my 1st place finish, many of my experiments did not work, and I am eager to learn and co-operate with anyone interested in this topic

What I was able to accomplish	What did not work
• **Empirical approach to define EM** • **Scalable and easy to navigate PaaS** • **End to end Data analytics Pipeline** • **Integration of $H2O$ robust ML**	• **Garbage in, Garbage out** • **Data provided from plenty sources, not ready-to-model** • **Uneven data quality** • **Fail to derive EM as depended variable from data sources** • **Lack of Observations**

MaYa 1.0

#Tackle 50

Define Economic Mobility

Through several literature review we came up of estimating the EM by regressing **log child income** (log Y_i) **on log parent income** Proposed by Solon, 1999 (a.k.a Intergenerational Income Elasticity) (log X_i), which yields a coefficient of :

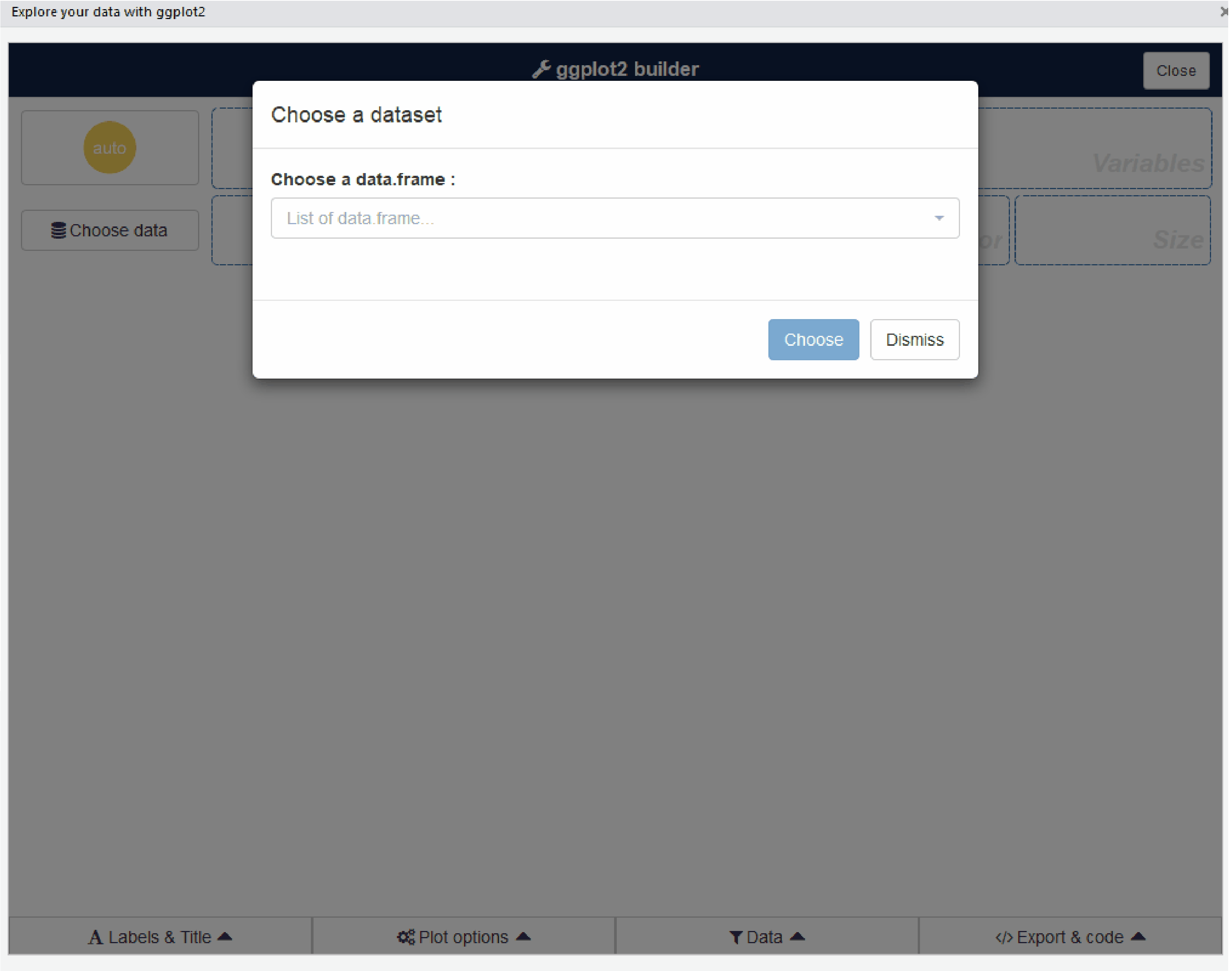
$$\rho XY \frac{SD(\log Y_i)}{SD(\log X_i)},$$

We proposing to use this coefficient estimates as a label e.i dependent variable. This totally make sense from our point of view as Economic mobility is proxy to that.

Original Paper: https://scholar.harvard.edu/files/hendren/files/mobility_geo.pdf

High-level analytics infrastructure

Tableau-like Drag and Drop GUI Visualization with Esquisse:



https://github.com/dreamRs/esquisse

Automatic and Scalable Machine Learning with H2O in R

What is H2O?

Java-Based Software for In-Memory Data Modeling

AutoML tends to automate the maximum number of steps in an ML pipeline — with a minimum amount of human effort — without compromising the model's performance by making ML accessible to everyone.

Open Source

Big Data Ecosystem

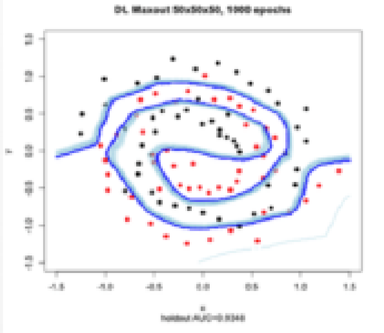
hadoop Spark

Flexible Interface

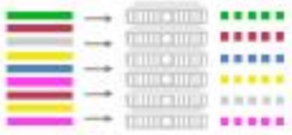
R Java python

Scala H_2O Flow

Smart and Fast Algorithms



Scalability and Performance



- Distributed In-Memory Computing Platform
- Distributed Algorithms
- Fine-Grain MapReduce

Rapid Model Deployment

- Highly portable models deployed in Java (POJO)
- Automated and streamlined scoring service deployment with Rest API*

GPU Enablement*

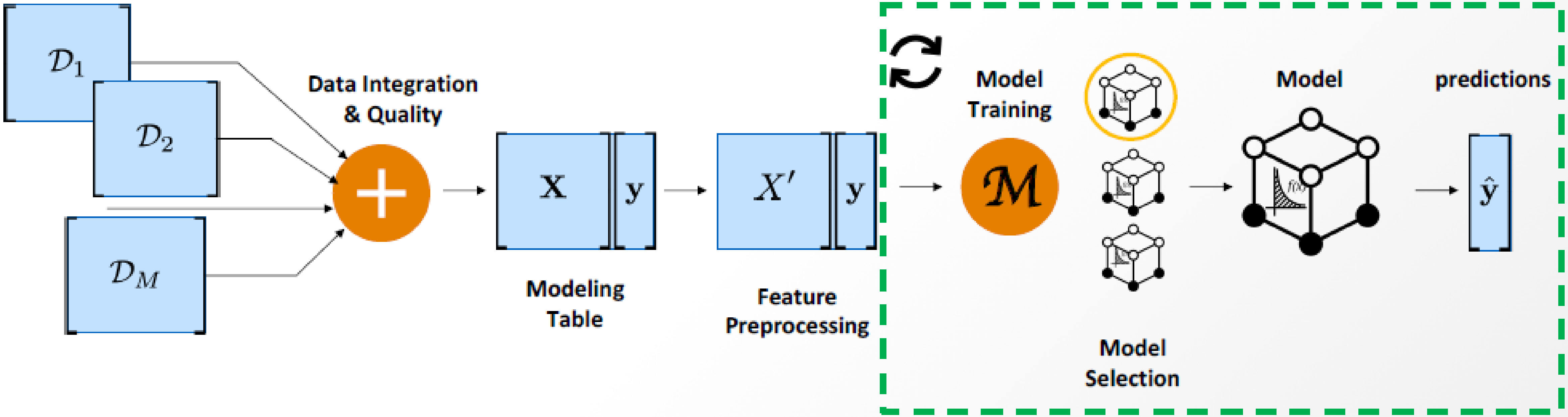
Cloud Integration

amazon web services

Microsoft Azure

Google

Where Automatic Machine Learning Fits



H2O's AutoML

H2O AutoML in R

Example

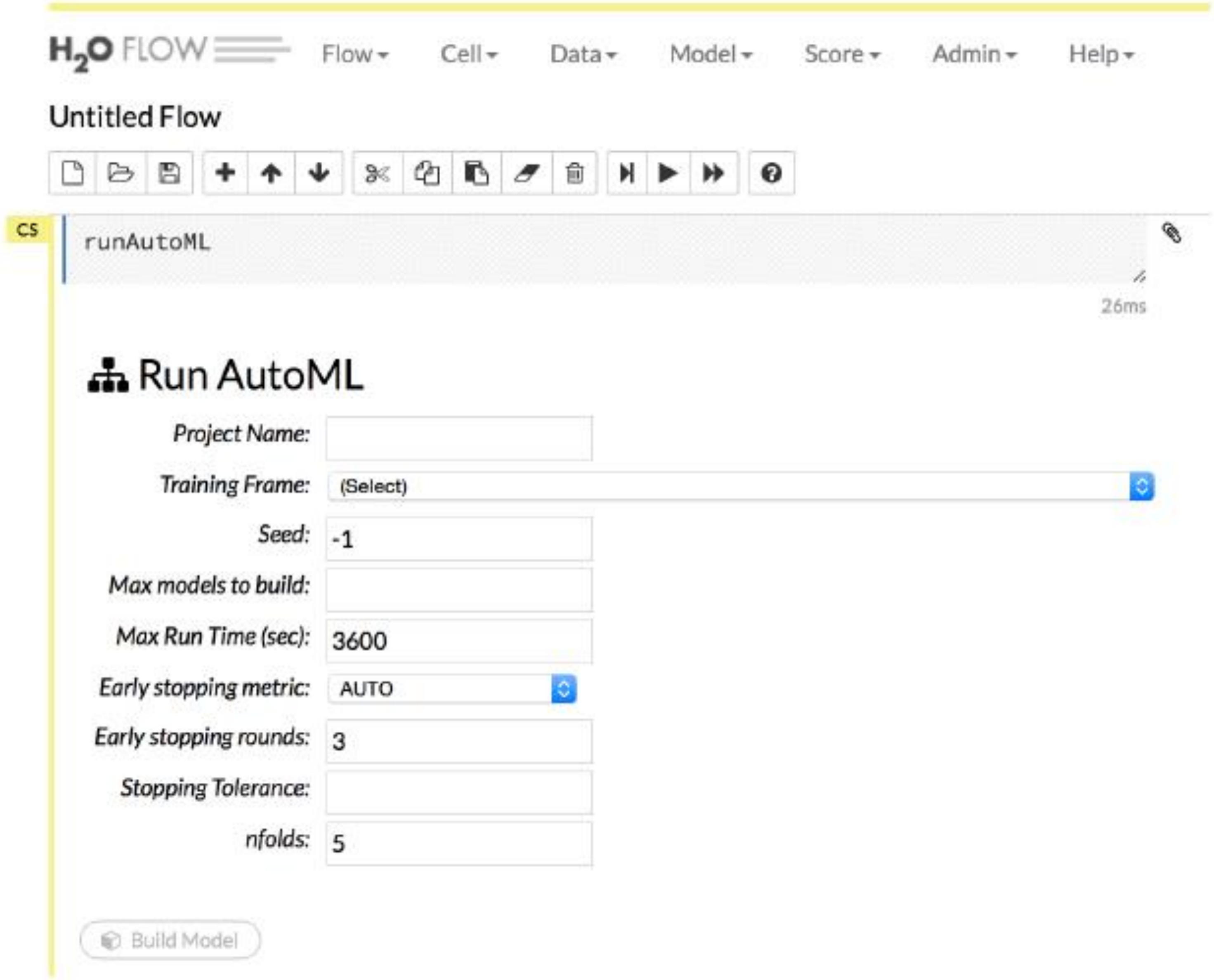
```r
library(h2o)
h2o.init()

train <- h2o.importFile("train.csv")


aml <- h2o.automl(y = "response_colname",
                  training_frame = train,
                  max_runtime_secs = 600)

lb <- aml@leaderboard
```

H2O AutoML in Flow GUI



H2O AutoML Leaderboard

model_id	auc	logloss
StackedEnsemble_AllModels_0_AutoML_20171121_012135	0.788321	0.554019
StackedEnsemble_BestOfFamily_0_AutoML_20171121_012135	0.783099	0.559286
GBM_grid_0_AutoML_20171121_012135_model_1	0.780554	0.560248
GBM_grid_0_AutoML_20171121_012135_model_0	0.779713	0.562142
GBM_grid_0_AutoML_20171121_012135_model_2	0.776206	0.564970
GBM_grid_0_AutoML_20171121_012135_model_3	0.771026	0.570270
DRF_0_AutoML_20171121_012135	0.734653	0.601520
XRT_0_AutoML_20171121_012135	0.730457	0.611706
GBM_grid_0_AutoML_20171121_012135_model_4	0.727098	0.666513
GLM_grid_0_AutoML_20171121_012135_model_0	0.685211	0.635138

Example Leaderboard for binary classification

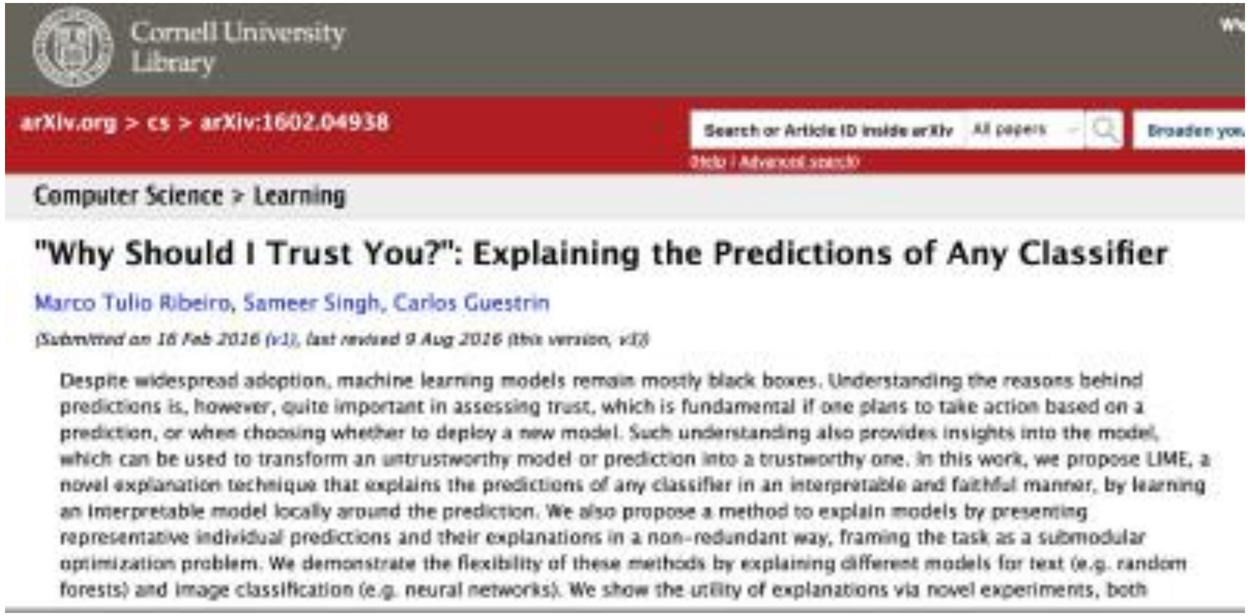
Why H2O?

- Being able to generate various models automatically and simultaneously
- Fully open-source
- Automatic training and tuning of many models.
- Scalable on Local Compute: Distributed, In-Memory Processing speeds up computation
- Easily deployable models to production with Docker + H2O AutoML + Shiny Web Applications
- Supports widely used statistical & machine learning algorithms, including GBMs, GLMs, Deep Learning, and etc
- Flexibility of choosing desired run time (default one hour)
- Specify the maximum number of models to build in an AutoML run
- Easily handles imbalanced data
- Choose sort metric like from RMSE, AUC, MSE, logloss etc

Documentation: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html

After you press the “red button”

LOCAL
INTERPRETABLE
MODEL-AGNOSTIC
EXPLANATIONS


Cornell University Library

arXiv.org > cs > arXiv:1602.04938

Search or Article ID inside arXiv | All papers | Broaden you

Help | Advanced search

Computer Science > Learning

"Why Should I Trust You?": Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin

(Submitted on 16 Feb 2016 (v1), last revised 9 Aug 2016 (this version, v3))

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing trust, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one. In this work, we propose LIME, a novel explanation technique that explains the predictions of any classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction. We also propose a method to explain models by presenting representative individual predictions and their explanations in a non-redundant way, framing the task as a submodular optimization problem. We demonstrate the flexibility of these methods by explaining different models for text (e.g. random forests) and image classification (e.g. neural networks). We show the utility of explanations via novel experiments, both

Job Done … or not …

"Just one more thing"

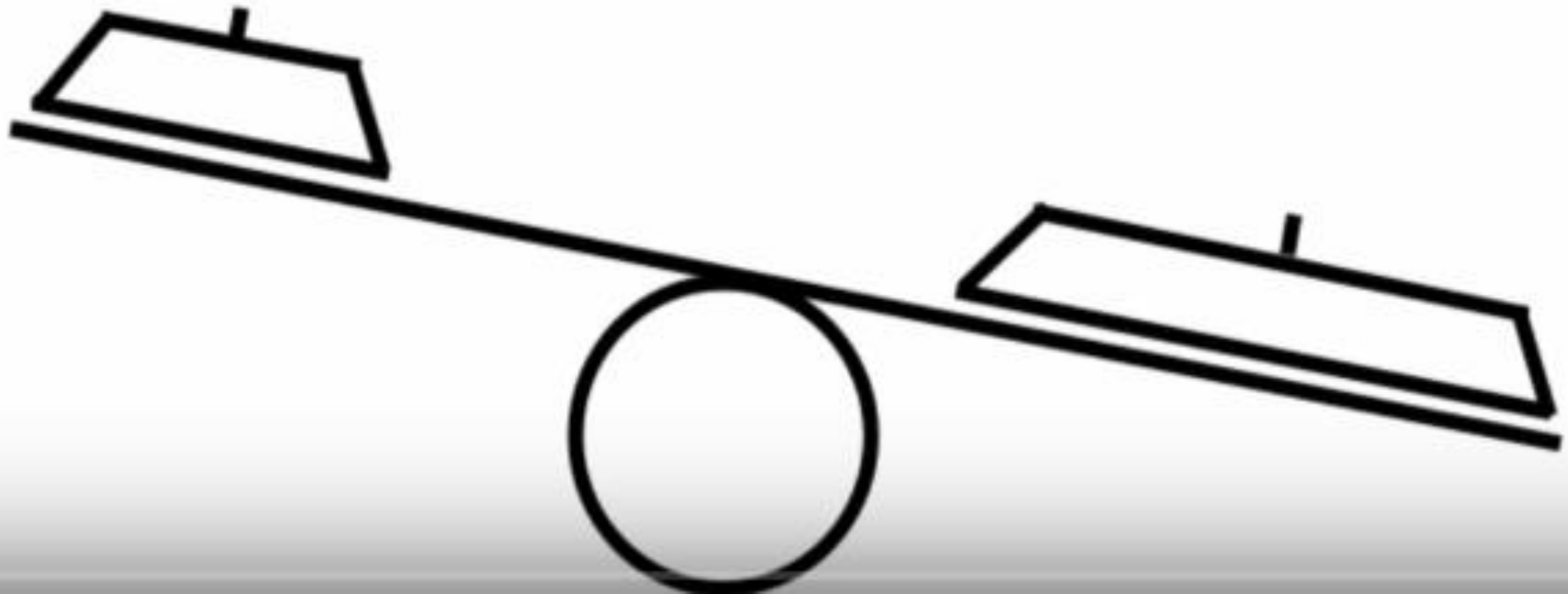
Job Done … or not …

Can you explain how your model works?

System whose internal workings are not readily understood

INTERPRETABILITY

ACCURACY



INTERPRETABILITY

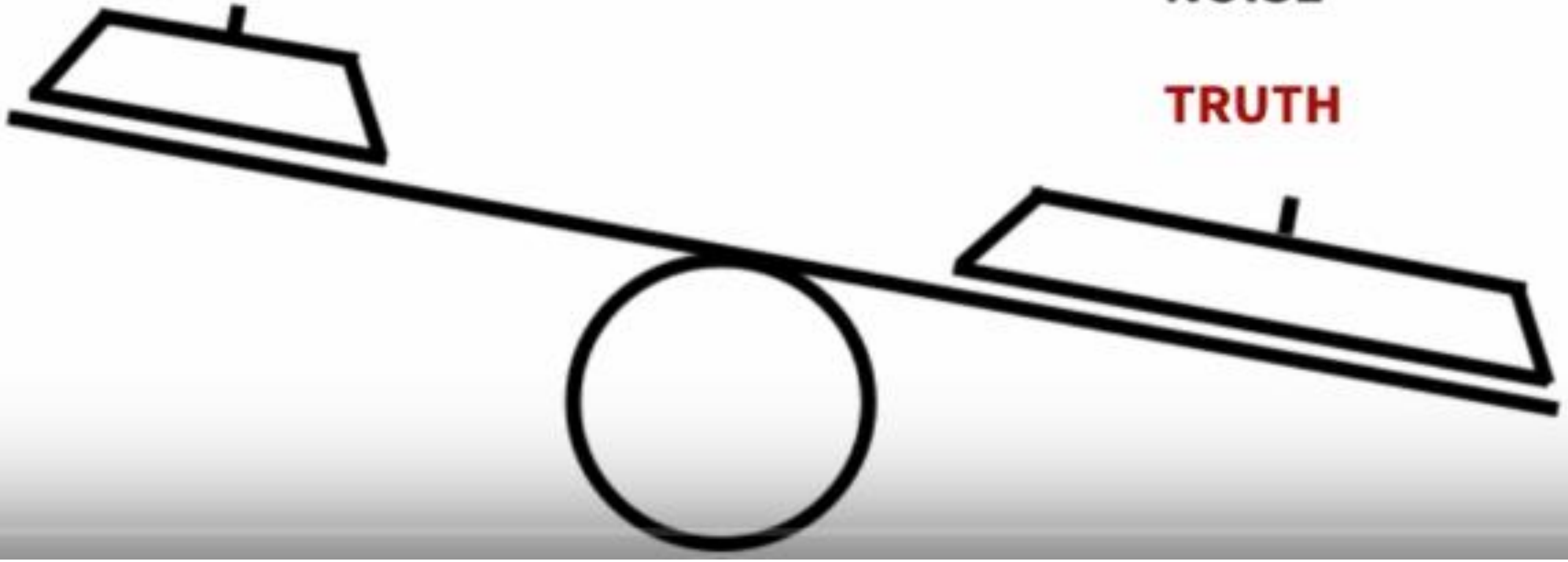
ACCURACY

OVERFITTING

CORRELATION

NOISE

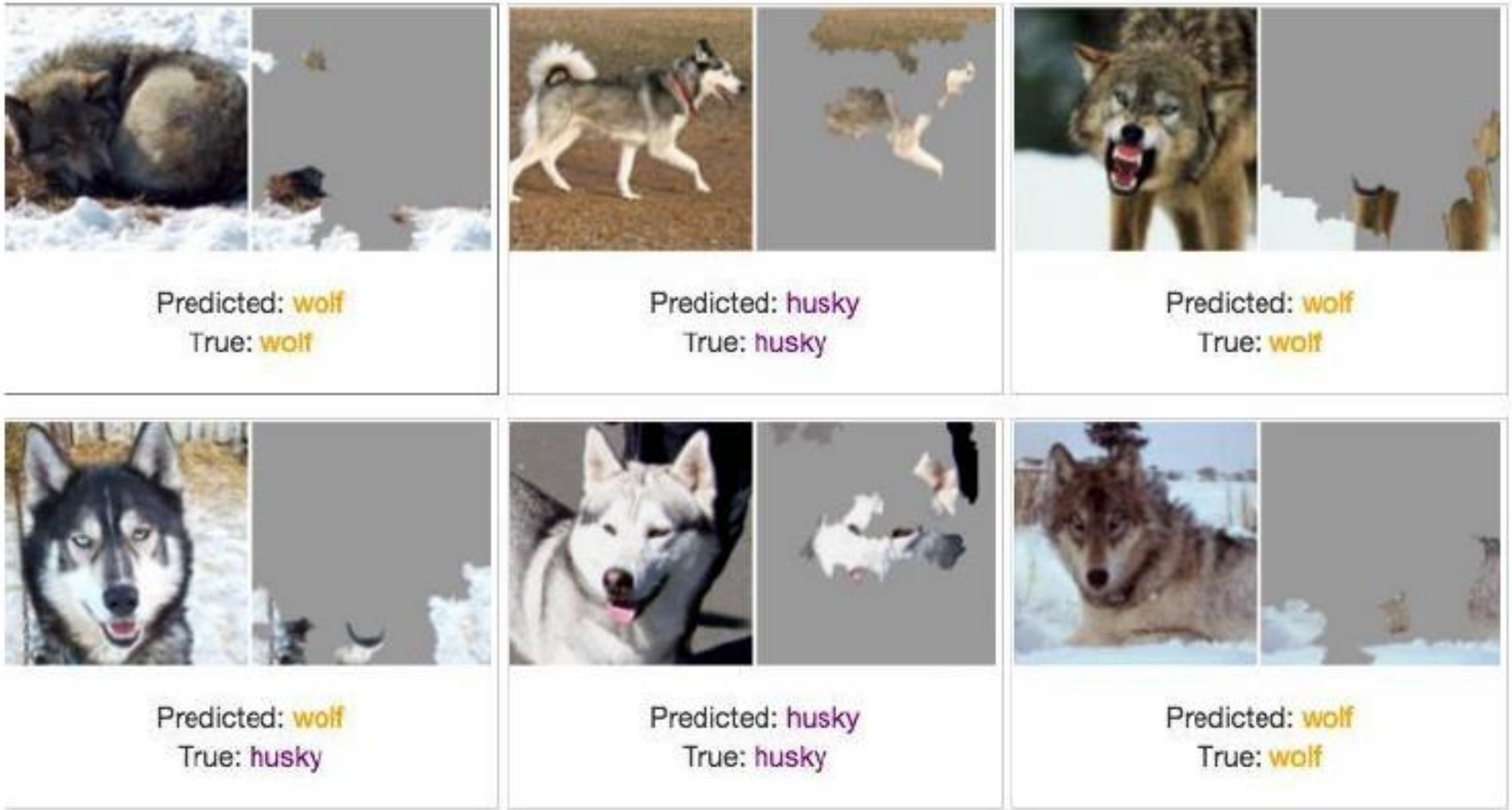
TRUTH



CAN YOU BUILD YOUR TRUST BASED ON ACCURACY?

Only 1 mistake!

… YES, IF YOU WANT TO BUILD A GREAT SNOW DETECTOR!



Some models are easy to interpret

Linear/Logistic regression

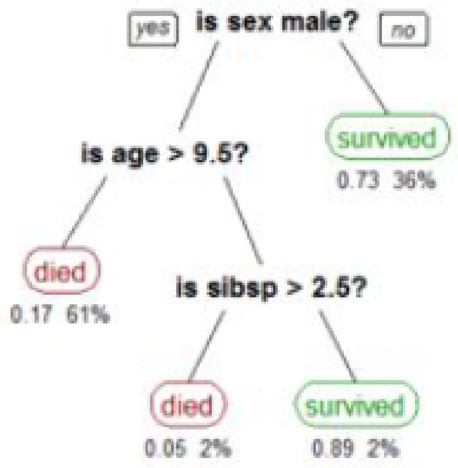
- Weight on each feature
- Know the exact contribution of each feature, negative or positive

$$Y = 3 * X1 - 2 * X2$$

Increasing X1 by 1 unit increases Y by 3 units

Single Decision Tree

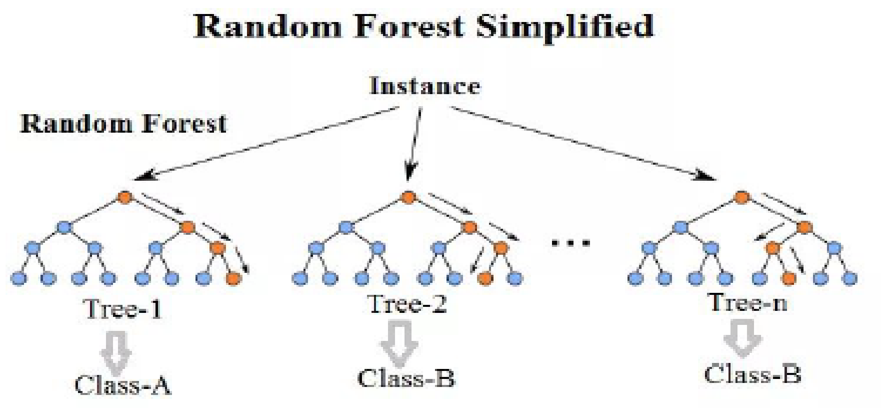
- Easy to understand how a decision was made by reading from top to bottom



Some models are harder to interpret

Ensemble models (random forest, boosting, etc…)

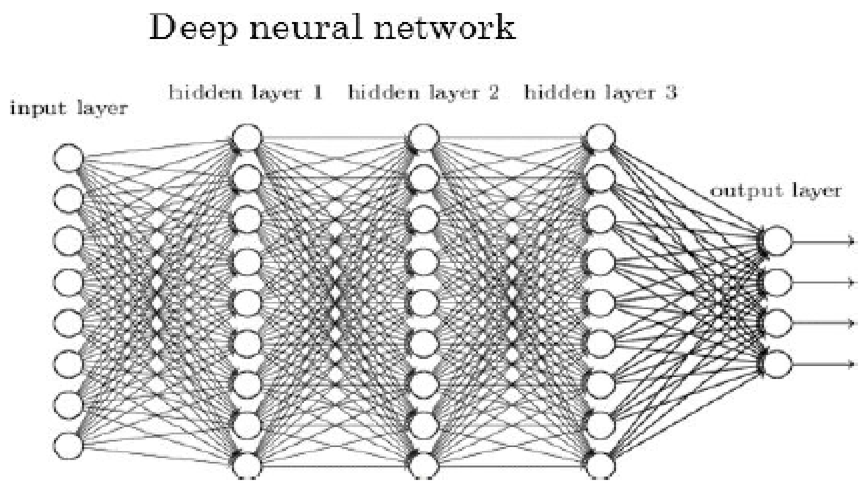
- Hard to understand the role of each feature
- Usually comes with **feature importance**
- Doesn't tell us if feature affects decision positively or negatively



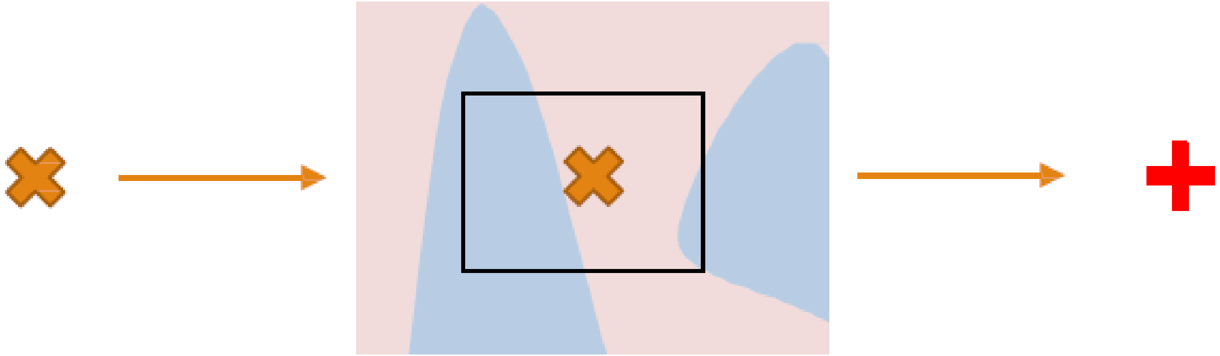
Some are really hard to interpret

Deep Neural Networks

- No straightforward way to relate output to input layer
- "Black-box"



Being Local and Model-Agnostic…



Local: Explains why a single data point was classified as a specific class

Model-agnostic: Treats the model as a black-box. Doesn’t need to know *how* it makes predictions

HOW LIME WORKS

1. Permute data*

2. Calculate distance between permutations and original observations*

3. Make predictions on new data using complex model

4. Pick m features best describing the complex model outcome from the permuted data.*

5. Fit a simple model to the permuted data with m features and similarity scores as weights *

6. Feature weights from the simple model make explanations for the complex models local behaviour

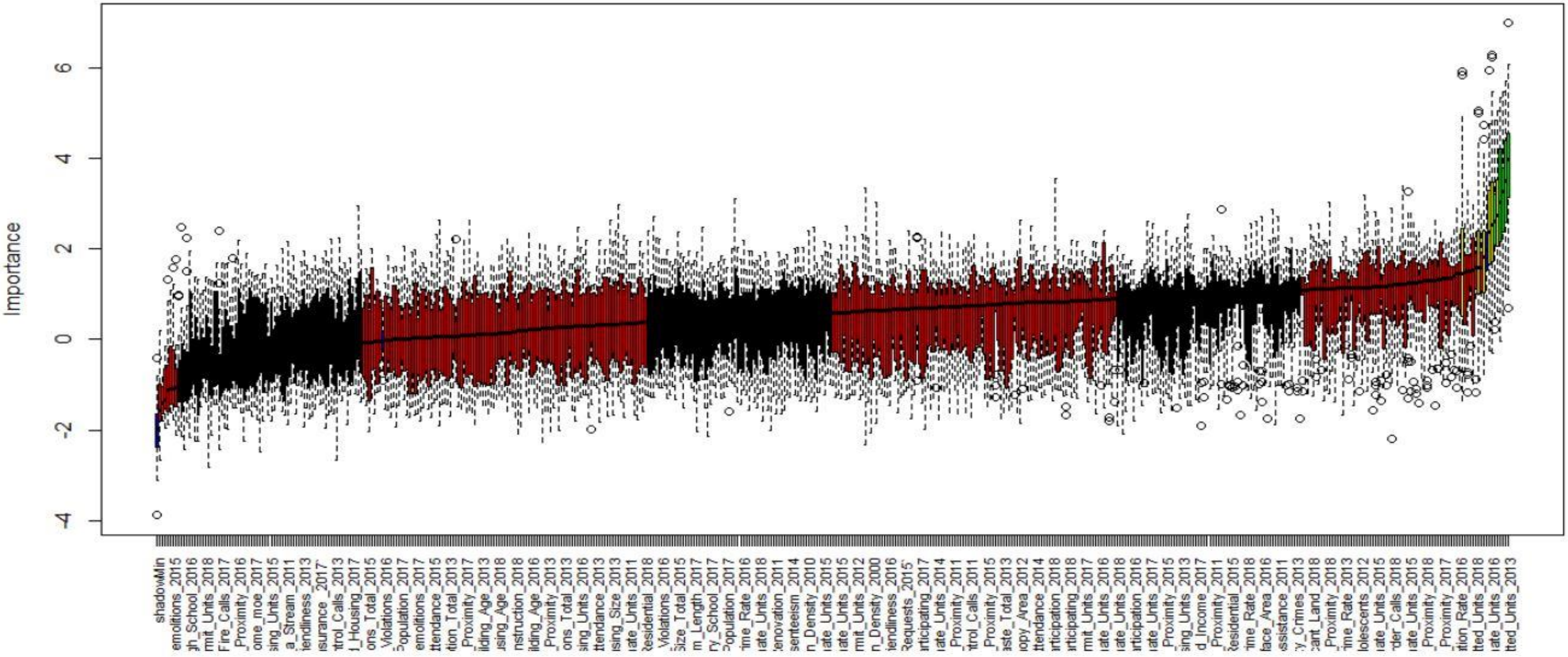
Conclusion

We Limed it

- Gives trust that our complex model makes correct predictions in an ethical way
- Can help debugging our model and spot biases in our data
- Can explain to others why a prediction was made
- Regulations make it mandatory (finance, GDPR, ...)

Alternatives to LIME are **SHAP** (Shapley Additive Explanations) and **ELI5** (Explain Like I am 5)

DEMO



https://ibrokhim-sadikov.shinyapps.io/test/

Acknowledgement

- **Marco Tulio Ribeiro**: Original LIME Framework and Python package

- **Thomas Lin Pedersen**: LIME R package
- **Matt Dancho**: LIME + H2O AutoML example + LIME R package improvement

- **Kasia Kulma**: LIME + H2O AutoML example

Q & A

Appendix

Links for detailed problem understanding on Subject matter:

- https://eml.berkeley.edu/~saez/geo_slides.pdf
- https://www.leadingonopportunity.org/report/executive-summary
- https://scholar.harvard.edu/files/hendren/files/mobility_geo.pdf
- https://opportunityinsights.org/

Reference for H2O autoML and Lime:

- https://uc-r.github.io/lime
- https://github.com/thomasp85/lime
- https://github.com/marcotcr/lime
- https://christophm.github.io/interpretable-ml-book/shapley.html
- https://github.com/slundberg/shap
- https://pydata.org/nyc2018/schedule/presentation/47/

Disclaimer: Most of the content and slides are obtained from original sources, H2O, R and Lime meetups with provided links. I do not claim the content as mine.

High Level Architecture