Machine Learning Explainability

Allen Li

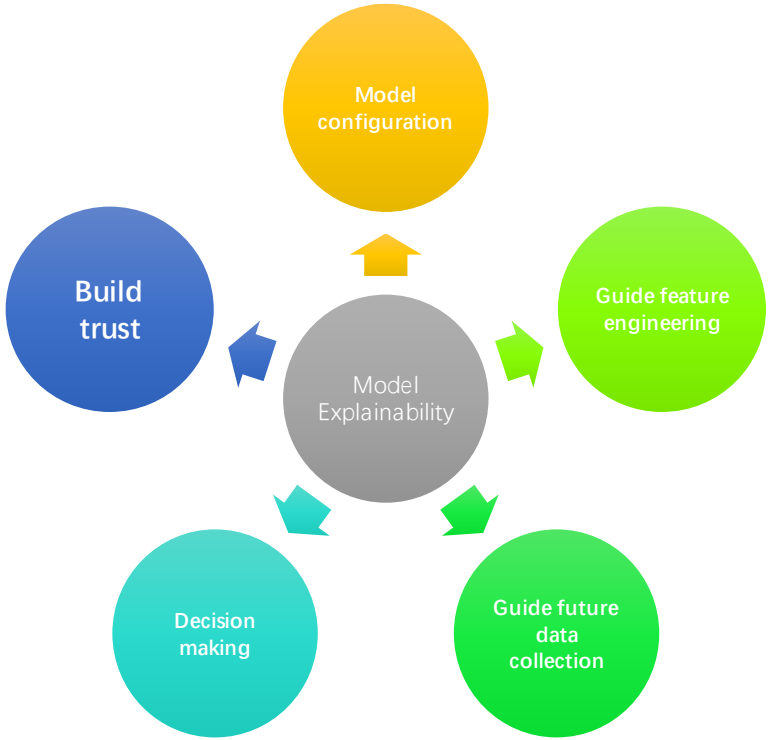
Black Box Models

The Great AI Debate - NIPS2017 - Yann LeCun

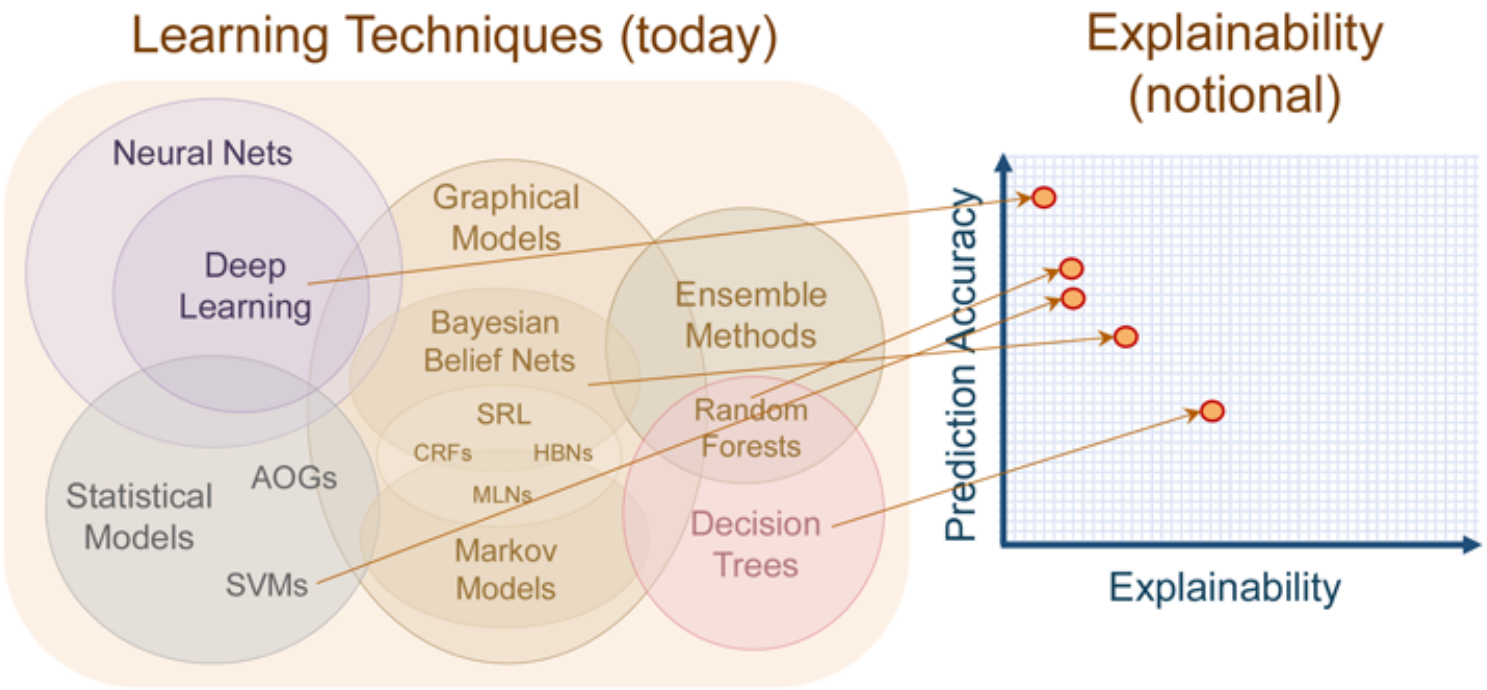
The Artificial Intelligence Channel

https://www.youtube.com/watch?v=93Xv8vJ2acI

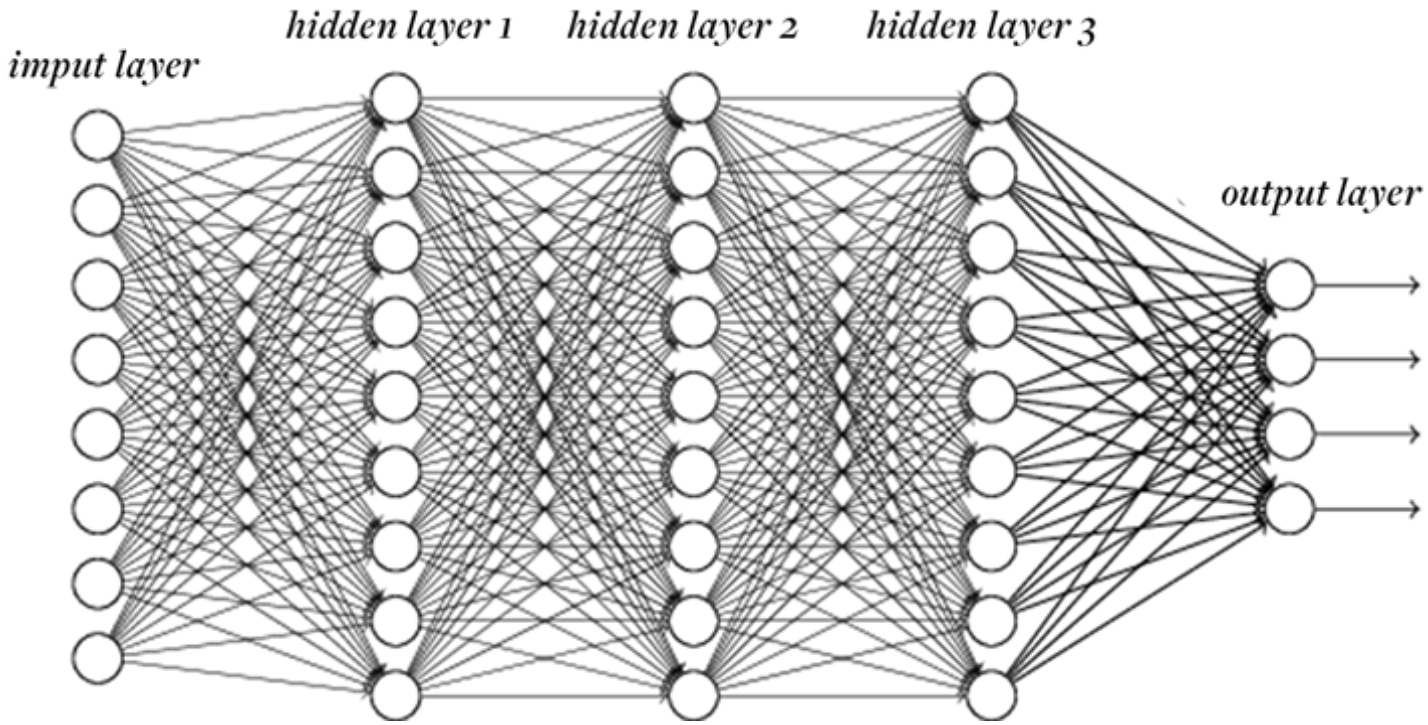
Model Explainability Implications



Model Comparisons



Challenges


imput layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

Some techniques

Before constructing model:

- Feature selection, engineering, data visualization
- Dimensionality reduction: eg. PCA, SOM
- Clustering

Selection of model:

- Rule-based / per-feature-based models
- Visualizations
- Model evaluation metrics/reports

After model being constructed:

Feature importance at global & individual level

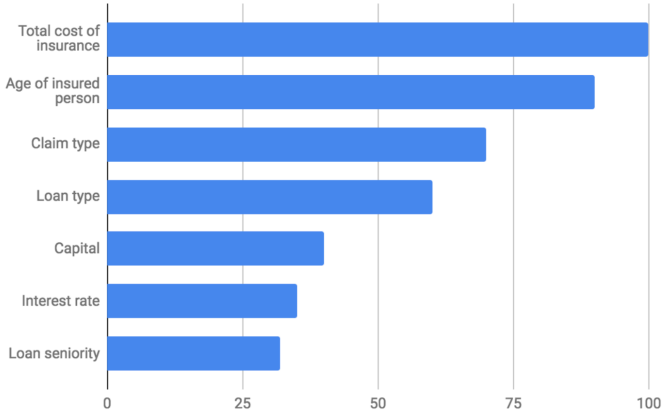
Random forest interpretation with scikit-learn

In one of my previous posts I discussed how random forests can be turned into a "white box", such that each prediction is decomposed into a sum of contributions from each feature i.e. $prediction = bias + feature_1\ contribution + \ldots + feature_n\ contribution$.

I've a had quite a few requests for code to do this. Unfortunately, most random forest libraries (including scikit-learn) don't expose tree paths of predictions. The implementation for sklearn required a hacky patch for exposing the paths. Fortunately, since 0.17.dev, scikit-learn has two additions in the API that make this relatively straightforward: obtaining leaf node_ids for predictions, and storing all intermediate values in all nodes in decision trees, not only leaf nodes. Combining these, it is possible to extract the prediction paths for each individual p
decompose the predictions via inspecting the paths.

Without further ado, the code is available at github, and also via `pip install tre`

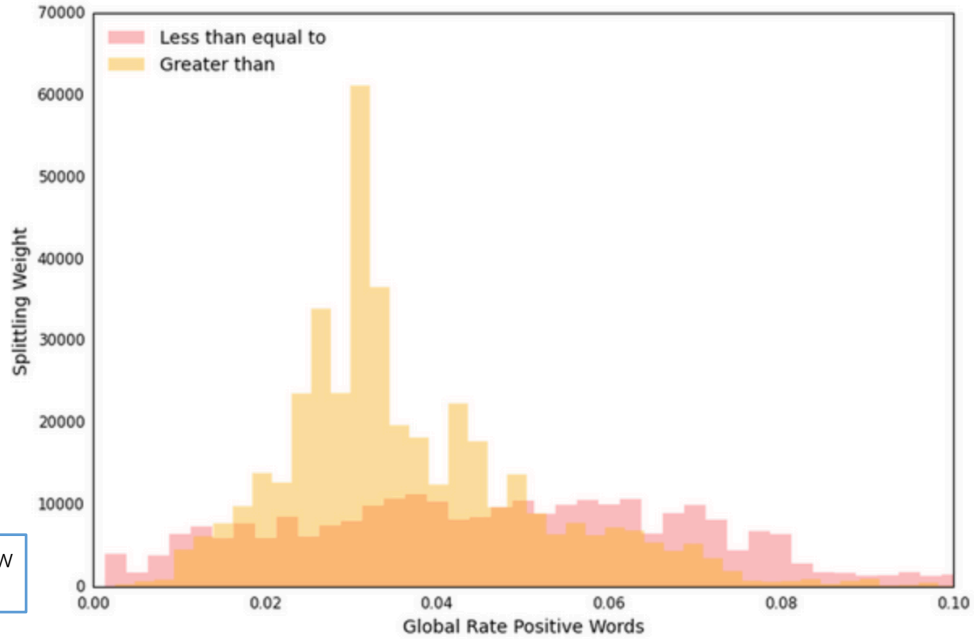
Note: this requires scikit-learn 0.17, which is still in development. You can check at http://scikit-learn.org/stable/install.html#install-bleeding-edge



Features importance of Zelros predictive insurance claim management system (illustration)

Pros: provide insights & explainability on features, easy to use
Cons: biased results

Decision threshold distribution



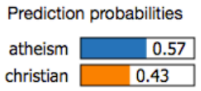
Airbnb approach on Random Forest explainability

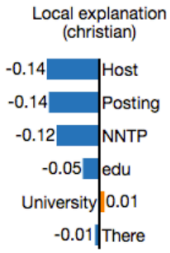
Pros: provide insights & explainability on decision thresholds, easy to follow
Cons: limited scope

Local Interpretable model-agnostic explanations (LIME)

Goal: to identify an **interpretable** model over the interpretable representation that is locally faithful to the classifier

pip install lime





Text with highlighted words

From: johnchad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11
NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the DAR
This is the same question I have and I have not seen an answer on the
net. If anyone has a contact please post on the net or email me.

Thanks,

Pros: model-agnostic
Cons: hard to fit on complex models with tricky feature engineering, special treatment for image & NLP, complicated to use

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

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

ABSTRACT

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 simulated and with human subjects, on various scenarios that require trust: deciding if one should trust a prediction, choosing between models, improving an untrustworthy classifier, and identifying why a classifier should not be trusted.

how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident that the model will perform well on real-world data, according to the metrics of interest. Currently, models are evaluated using accuracy metrics on an available validation dataset. However, real-world data is often significantly different, and further, the evaluation metric may not be indicative of the product's goal. Inspecting individual predictions and their explanations is a worthwhile solution, in addition to such metrics. In this case, it is important to aid users by suggesting which instances to inspect, especially for large datasets.

In this paper, we propose providing explanations for individual predictions as a solution to the "trusting a prediction" problem, and selecting multiple such predictions (and explanations) as a solution to the "trusting the model" problem. Our main contributions are summarized as follows.

- LIME, an algorithm that can explain the predictions of *any*

SHAP

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg
Paul G. Allen School of Computer Science
University of Washington
Seattle, WA 98105
slund1@cs.washington.edu

Su-In Lee
Paul G. Allen School of Computer Science
Department of Genome Sciences
University of Washington
Seattle, WA 98105
suinlee@cs.washington.edu

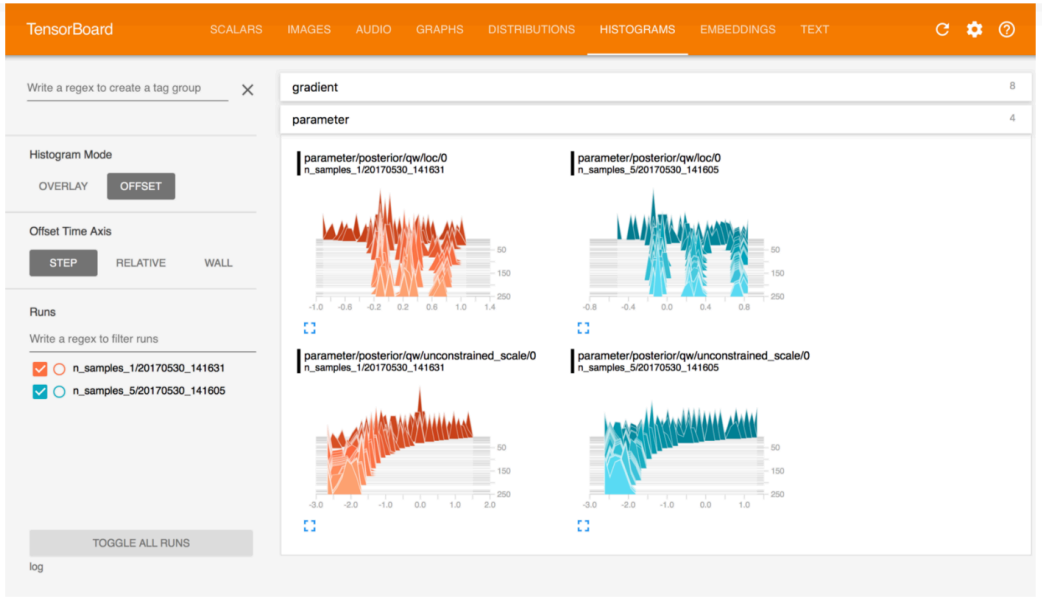
Abstract

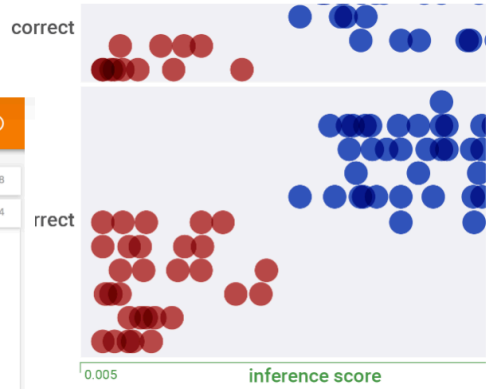
Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to

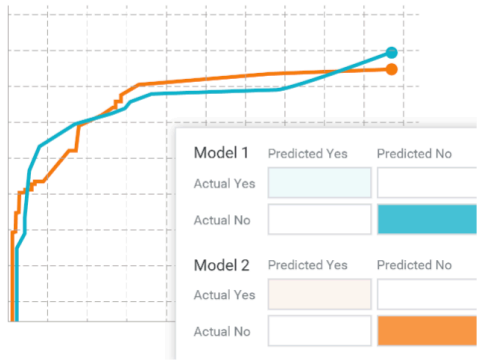
pip install shap

https://github.com/slundberg/shap

Tensorboard & what-if tool

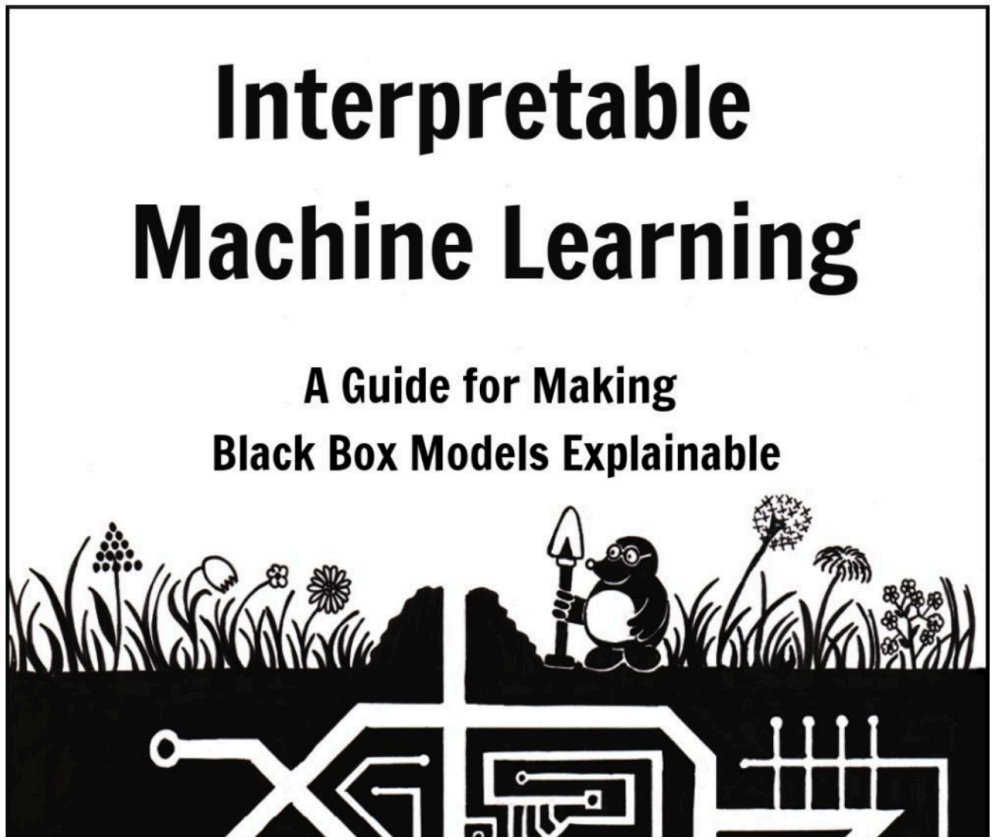






Pros: helps visually understand models and results
Cons: restricted to tensorflow, narrow audience

End with a book



https://christophm.github.io/interpretable-ml-book/index.html